

# Indicadores

Hunor Moriczi

2022-05-05

## Indicadores en R

En este documento se hará una rápida guía en el proceso de carga de datos desde un archivo .csv, obtención de medias, creación de gráficas de barras, gráficas de densidad etc. Por último se crearán una serie de indicadores que pueden ser útiles para verificar ciertas clases o métodos dentro de un software.

En este caso el software analizado es SonarQube, concretamente las siguientes versiones:

1. 8.9.7.52159
2. 9.2.4.50792
3. 9.3.0.51899

## Ejemplo de función en RMarkdown

En este apartado se mostrará la forma de crear una función sencilla en RMarkdown.

Se crea una variable llamada ejemplo con la cadena "variable de ejemplo" y la muestra.

```
ejemplo <- "variable de ejemplo"
ejemplo
## [1] "variable de ejemplo"
```

## Carga de datos

Para empezar a trabajar se va a relanzar la carga de datos de los archivos .csv, obtenidos en el análisis de SonarQube mediante SourceMeter. Estos datos se encuentran en el directorio de este proyecto, dentro del directorio datos y separados en carpetas identificativas, según la versión analizada del software.

Primero obtengo las rutas de los archivos mediante la función file.choose(), que abrirá un explorador del sistema y me dará las rutas de los archivos seleccionados.

Estas rutas las guardaré en variables para poder hacer uso de ella más adelante.

Obtengo todas las rutas de los archivos obtenidos, con las métricas de clases y métodos, de las 3 versiones analizadas:

```

ruta_ClassV1 <- "datos\\sonarqube-8.9.7.52159\\SonarQube_Hunor-Class-
V1.csv"
ruta_ClassV2 <- "datos\\sonarqube-9.2.4.50792\\SonarQube_Hunor-Class-
V2.csv"
ruta_ClassV3 <- "datos\\sonarqube-9.3.0.51899\\SonarQube_Hunor-Class-
V3.csv"

ruta_MethodV1 <- "datos\\sonarqube-8.9.7.52159\\SonarQube_Hunor-Method-
V1.csv"
ruta_MethodV2 <- "datos\\sonarqube-9.2.4.50792\\SonarQube_Hunor-Method-
V2.csv"
ruta_MethodV3 <- "datos\\sonarqube-9.3.0.51899\\SonarQube_Hunor-Method-
V3.csv"

```

Ahora mediante la librería readr, guardo los datos de los excel en variables.

```

library(readr)
datasetClassV1 <- read_csv(ruta_ClassV1, show_col_types = FALSE)
datasetClassV2 <- read_csv(ruta_ClassV2, show_col_types = FALSE)
datasetClassV3 <- read_csv(ruta_ClassV3, show_col_types = FALSE)

datasetMethodV1 <- read_csv(ruta_MethodV1, show_col_types = FALSE)
datasetMethodV2 <- read_csv(ruta_MethodV2, show_col_types = FALSE)
datasetMethodV3 <- read_csv(ruta_MethodV3, show_col_types = FALSE)

```

Una vez cargados los datos pueden visualizarse a través del comando view(X) o simplemente escribiendo el nombre del valor en la consola.

## Archivos Class

En estos archivos se encuentran las metricas relacionadas con las clases, realizadas mediante SourceMeter. Si se repasan todas las columnas se puede ver un total de más de 40 métricas.

Analizaré las que creo que pueden aportar mayor información o que me pueden ser más útiles a la hora de crear los indicadores.

- Una metrica interesante es la columna CBO. El CBO de una clase es el número de clases a las cuales una clase está ligada. Se da dependencia entre dos clases cuando una clase usa métodos o variables de la otra clase. Las clases relacionadas por herencia no se tienen en cuenta.

Se propone como indicador del esfuerzo necesario para el mantenimiento y en el testeo. Cuanto más acoplamiento se da en una clase, más difícil será reutilizarla.

- Otra que tomaré en cuenta es la columna RFC, que es el cardinal del conjunto de todos los métodos que pueden ser invocados en respuesta a un mensaje a un objeto de la clase o por alguno método en la clase. Esto incluye todos los métodos accesibles dentro de la jerarquía de la clase.

En otras palabras, cuenta las ocurrencias de llamadas a otras clases desde una clase particular.

- Otra medida que se puede tener en cuenta es la columna LCOM, que es una medida de la cohesión de una clase midiendo el número de atributos comunes usados por diferentes métodos, indicando la calidad de la abstracción hecha en la clase. Un valor alto de LCOM implica falta de cohesión, es decir, escasa similitud de los métodos.

Esto puede indicar que la clase está compuesta de elementos no relacionados, incrementando la complejidad y la probabilidad de errores durante el desarrollo.

- Otra medida que se puede emplear es la columna WMC. Que mide la complejidad de una clase. Según he visto en apuntes de clase, se sugiere un umbral de 40 o 20, dependiendo si las clases son o no de interface de usuario respectivamente.

Propondré un indicador para las clases que tengas valores superiores a estos.

#### Archivos Method

- Una métrica para los métodos podría CD o TCLOC, densidad de comentarios o líneas totales de comentarios por método.

El problema a la hora de crear los indicadores con las métricas de métodos es que un gran % no tiene datos relevantes o el valor es 0. Para trabajar con medias u otros datos, no puedo tener en cuenta por ejemplo la media de líneas de código de comentarios ya que me da un valor de 0.48 líneas de código de comentario por método.

Si no tengo en cuenta las clases con 0 comentarios, 80%, y haga los indicadores, quizás sean datos falseados y no tengan relevancia o un sentido real.

#### Variables con las métricas

En el siguiente apartado creo las siguientes variables, que contendrán las columnas con las métricas que me interesan de cada uno de los dataset cargados.

- LOC -> Líneas de código
- CBO -> Acoplamiento entre clases
- WMC -> Complejidad de una clase
- LCOM5 -> Grado de cohesión de una clase

```
LOC_classV1 <- datasetClassV1$LOC
LOC_classV2 <- datasetClassV2$LOC
LOC_classV3 <- datasetClassV3$LOC
```

```
CBO_classV1 <- datasetClassV1$CBO
CBO_classV2 <- datasetClassV2$CBO
CBO_classV3 <- datasetClassV3$CBO
```

```

WMC_classV1 <- datasetClassV1$WMC
WMC_classV2 <- datasetClassV2$WMC
WMC_classV3 <- datasetClassV3$WMC

LCOM5_classV1 <- datasetClassV1$LCOM5
LCOM5_classV2 <- datasetClassV2$LCOM5
LCOM5_classV3 <- datasetClassV3$LCOM5

LOC_methodV1 <- datasetMethodV1$LOC
LOC_methodV2 <- datasetMethodV2$LOC
LOC_methodV3 <- datasetMethodV3$LOC

```

## Máximos y mínimos

En este bloque se obtienen los minimos y maximos de cada vector de datos, para poder realizar unas tablas de frecuencia con unos rangos con sentido

- Cálculo de min y max LOC clases

```

min_LOC_ClassV1 <- min(datasetClassV1$LOC)
min_LOC_ClassV2 <- min(datasetClassV2$LOC)
min_LOC_ClassV3 <- min(datasetClassV3$LOC)
max_LOC_ClassV1 <- max(datasetClassV1$LOC)
max_LOC_ClassV2 <- max(datasetClassV2$LOC)
max_LOC_ClassV3 <- max(datasetClassV3$LOC)

```

- Cálculo de min y max CBO clases

```

min_CBO_ClassV1 <- min(datasetClassV1$CBO)
min_CBO_ClassV2 <- min(datasetClassV2$CBO)
min_CBO_ClassV3 <- min(datasetClassV3$CBO)
max_CBO_ClassV1 <- max(datasetClassV1$CBO)
max_CBO_ClassV2 <- max(datasetClassV2$CBO)
max_CBO_ClassV3 <- max(datasetClassV3$CBO)

```

- Cálculo de min y max LCOM5 clases

```

min_LCOM5_ClassV1 <- min(datasetClassV1$LCOM5)
min_LCOM5_ClassV2 <- min(datasetClassV2$LCOM5)
min_LCOM5_ClassV3 <- min(datasetClassV3$LCOM5)
max_LCOM5_ClassV1 <- max(datasetClassV1$LCOM5)
max_LCOM5_ClassV2 <- max(datasetClassV2$LCOM5)
max_LCOM5_ClassV3 <- max(datasetClassV3$LCOM5)

```

- Cálculo de min y max WMC clases

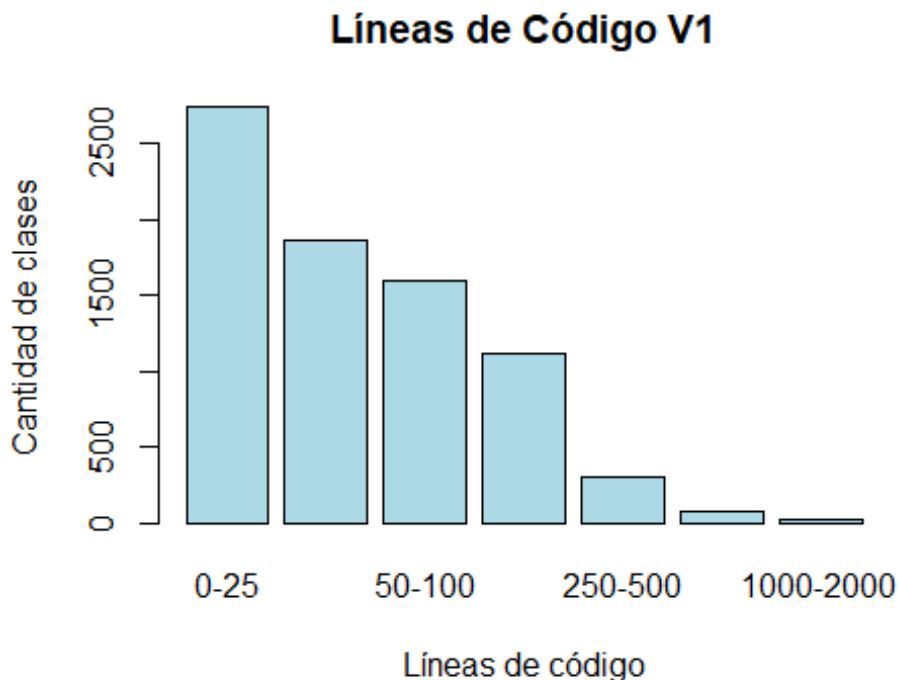
```

min_WMC_ClassV1 <- min(datasetClassV1$WMC)
min_WMC_ClassV2 <- min(datasetClassV2$WMC)
min_WMC_ClassV3 <- min(datasetClassV3$WMC)
max_WMC_ClassV1 <- max(datasetClassV1$WMC)
max_WMC_ClassV2 <- max(datasetClassV2$WMC)
max_WMC_ClassV3 <- max(datasetClassV3$WMC)

```

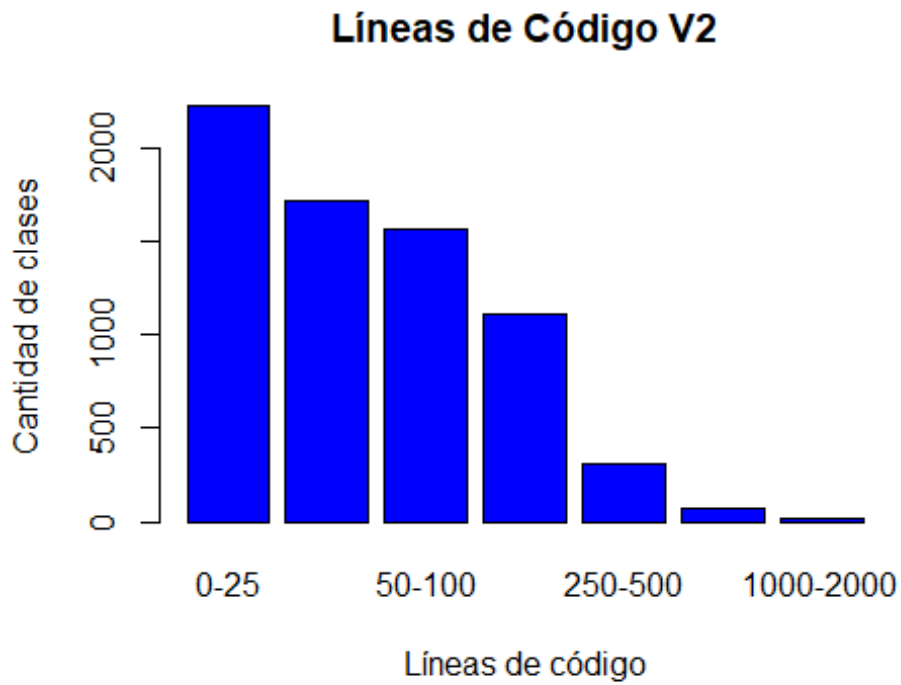
### Gráficas de barras con el LOC de las clases de las 3 versiones

```
freqLOC_classV1 <- table(cut(LOC_classV1, breaks =  
c(0,25,50,100,250,500,1000,2000), right = F, include.lowest = T), exclude  
= NULL)  
freqLOC_classV2 <- table(cut(LOC_classV2, breaks =  
c(0,25,50,100,250,500,1000,2000), right = F, include.lowest = T), exclude  
= NULL)  
freqLOC_classV3 <- table(cut(LOC_classV3, breaks =  
c(0,25,50,100,250,500,1000,2000), right = F, include.lowest = T), exclude  
= NULL)  
  
rownames(freqLOC_classV1) <- c('0-25', '25-50', '50-100', '100-250', '250-  
500', '500-1000', '1000-2000')  
rownames(freqLOC_classV2) <- c('0-25', '25-50', '50-100', '100-250', '250-  
500', '500-1000', '1000-2000')  
rownames(freqLOC_classV3) <- c('0-25', '25-50', '50-100', '100-250', '250-  
500', '500-1000', '1000-2000')  
  
barplot(freqLOC_classV1,  
  main = 'Líneas de Código V1',  
  col = 'lightblue',  
  xlab = 'Líneas de código',  
  ylab = 'Cantidad de clases')
```

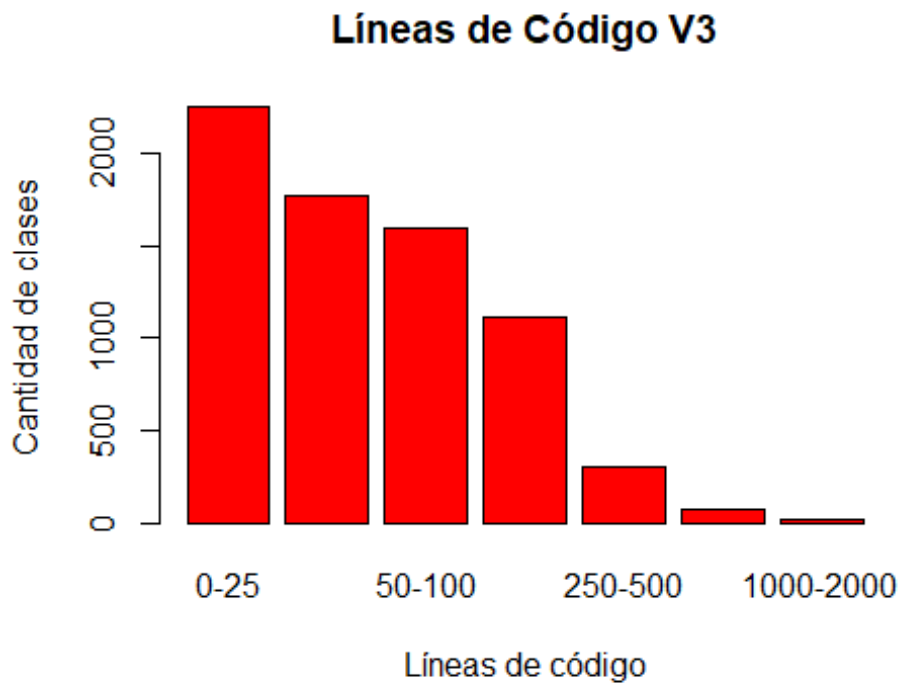


```
barplot(freqLOC_classV2,  
  main = 'Líneas de Código V2',  
  col = 'blue',
```

```
xlab = 'Líneas de código',  
ylab = 'Cantidad de clases')
```



```
barplot(freqLOC_classV3,  
main = 'Líneas de Código V3',  
col = 'red',  
xlab = 'Líneas de código',  
ylab = 'Cantidad de clases')
```

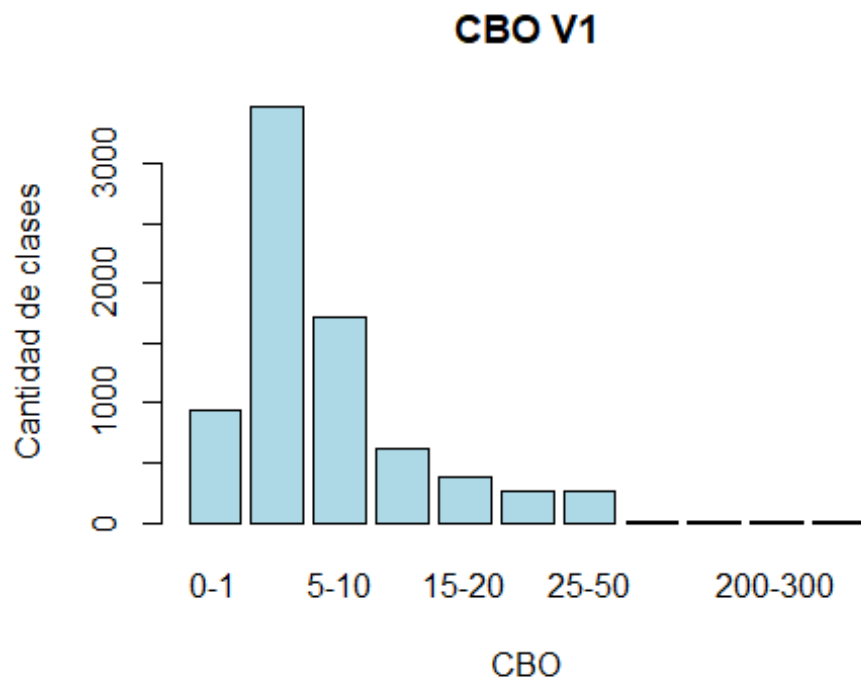


#### Gráficas de barras con el CBO de las clases de las 3 versiones

```
freqCBO_classV1 <- table(cut(CBO_classV1, breaks =
c(0,1,5,10,15,20,25,50,100,200,300,321), right = F, include.lowest = T),
exclude = NULL)
freqCBO_classV2 <- table(cut(CBO_classV2, breaks =
c(0,1,5,10,15,20,25,50,100,200,250), right = F, include.lowest = T),
exclude = NULL)
freqCBO_classV3 <- table(cut(CBO_classV3, breaks =
c(0,1,5,10,15,20,25,50,100,200,250), right = F, include.lowest = T),
exclude = NULL)

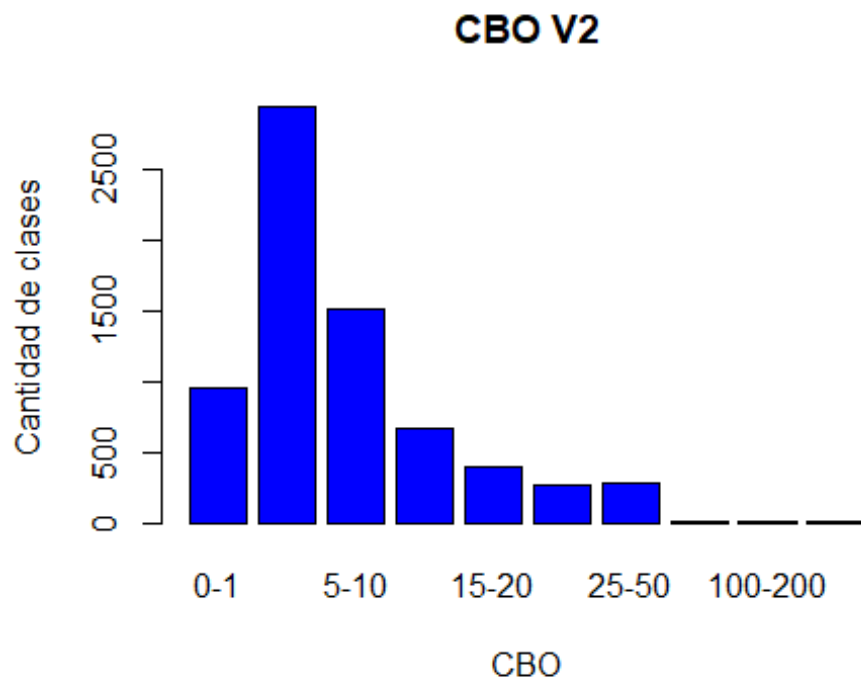
rownames(freqCBO_classV1) <- c('0-1', '1-5', '5-10', '10-15', '15-20', '20-
25', '25-50', '50-100', '100-200', '200-300', '300-321')
rownames(freqCBO_classV2) <- c('0-1', '1-5', '5-10', '10-15', '15-20', '20-
25', '25-50', '50-100', '100-200', '200-250')
rownames(freqCBO_classV3) <- c('0-1', '1-5', '5-10', '10-15', '15-20', '20-
25', '25-50', '50-100', '100-200', '200-250')

barplot(freqCBO_classV1,
        main = 'CBO V1',
        col = 'lightblue',
        xlab = 'CBO',
        ylab = 'Cantidad de clases ')
```

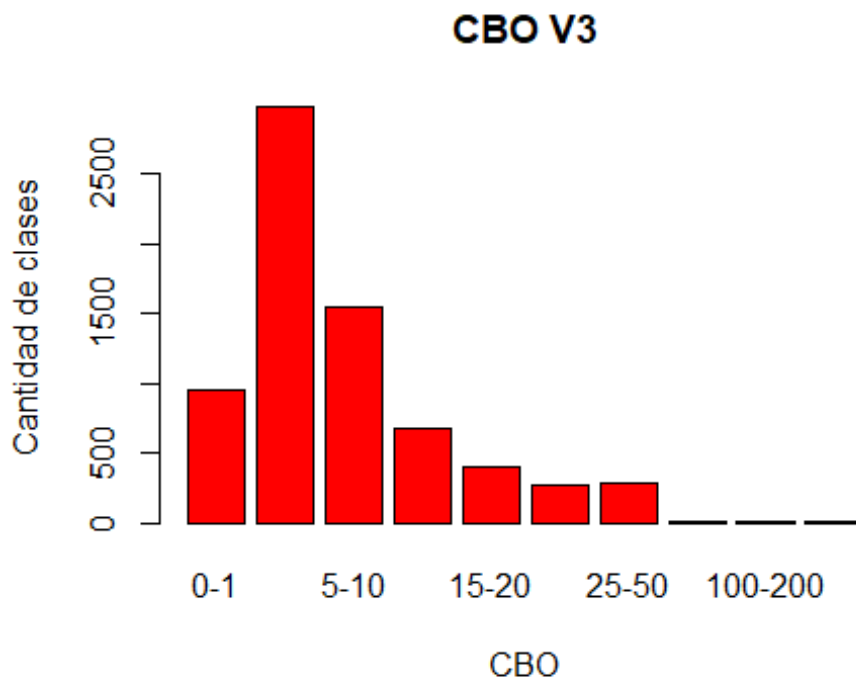


```
barplot(freqCBO_classV2,  
        main = 'CBO V2',  
        col = 'blue',  
        xlab = 'CBO',  
        ylab = 'Cantidad de clases')
```





```
barplot(freqCBO_classV3,  
        main = 'CBO V3',  
        col = 'red',  
        xlab = 'CBO',  
        ylab = 'Cantidad de clases')
```

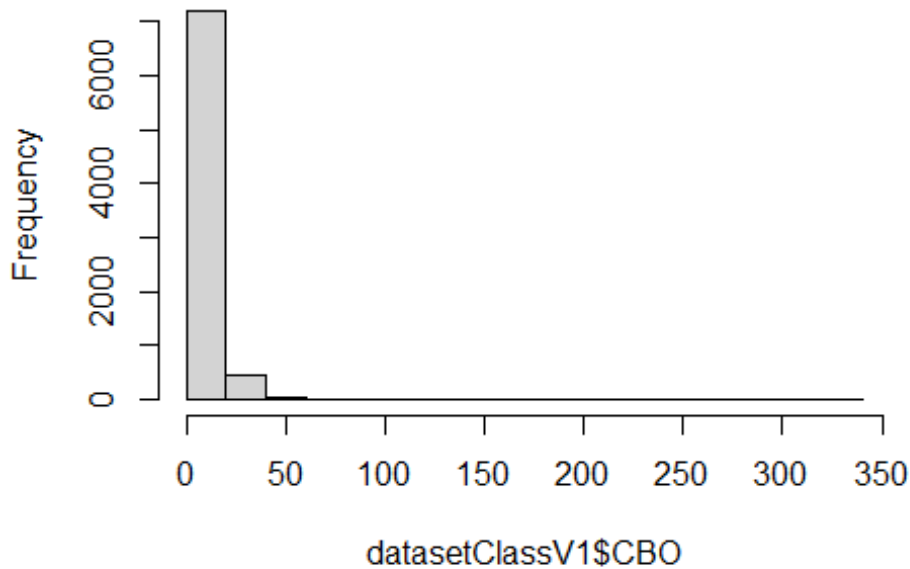


Mediante un histograma se puede sacar la misma información, pero para ver mejor la dispersión de los datos, es mejor emplear barplot y establecer otros rangos personalizados.

Se puede ver en el siguiente ejemplo con el CBO de la versión 1.

```
hist(datasetClassV1$CBO)
```

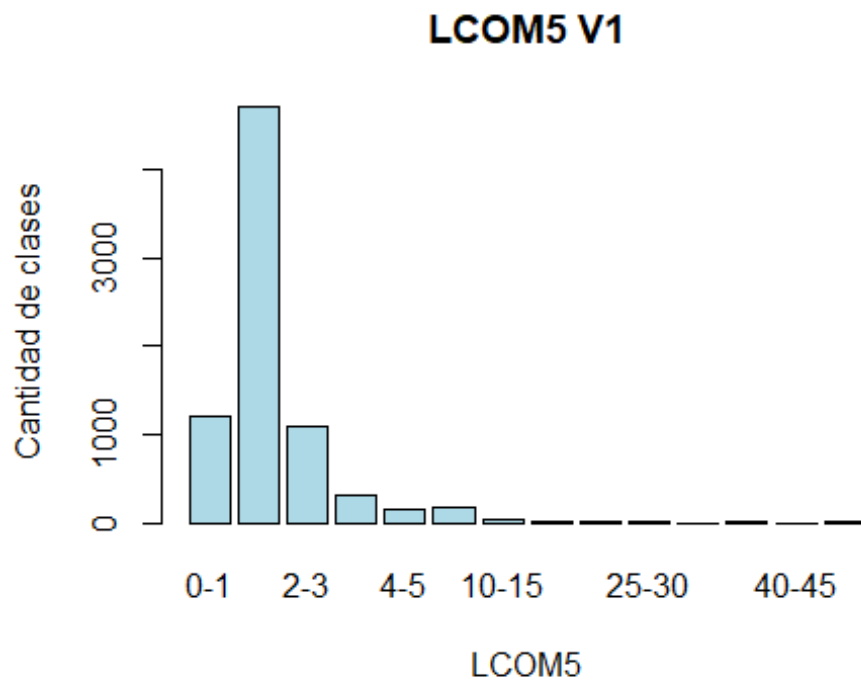
## Histogram of datasetClassV1\$CBO



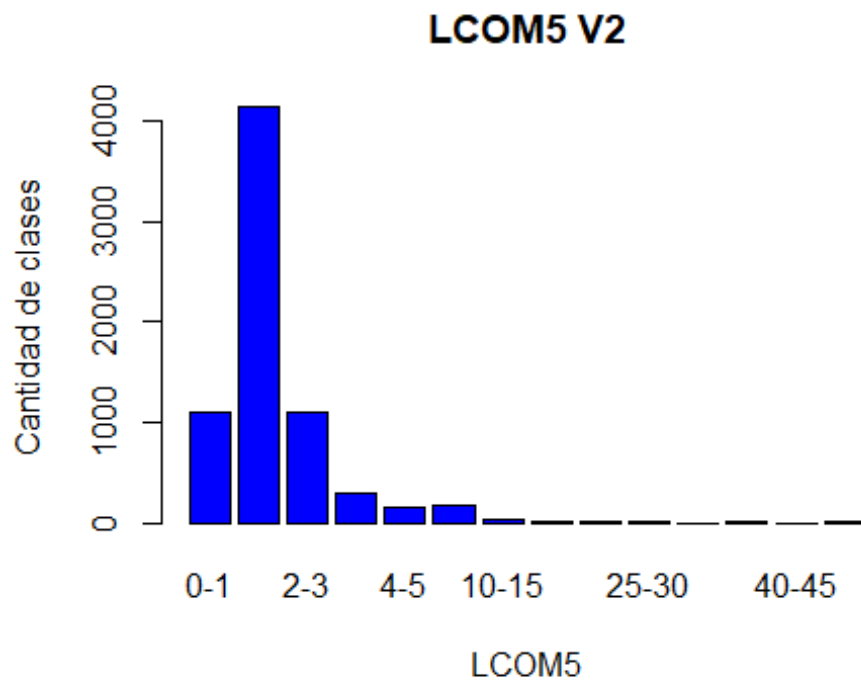
### Gráficas de barras con el LCOM5 de las clases de las 3 versiones

```
freqLCOM5_classV1 <- table(cut(LCOM5_classV1, breaks =
c(0,1,2,3,4,5,10,15,20,25,30,35,40,45,55), right = F, include.lowest =
T), exclude = NULL)
freqLCOM5_classV2 <- table(cut(LCOM5_classV2, breaks =
c(0,1,2,3,4,5,10,15,20,25,30,35,40,45,55), right = F, include.lowest =
T), exclude = NULL)
freqLCOM5_classV3 <- table(cut(LCOM5_classV3, breaks =
c(0,1,2,3,4,5,10,15,20,25,30,35,40,45,55), right = F, include.lowest =
T), exclude = NULL)
rownames(freqLCOM5_classV1) <- c('0-1', '1-2', '2-3', '3-4', '4-5', '5-
10', '10-15', '15-20', '20-25', '25-30', '30-35', '35-40', '40-45', '45-55')
rownames(freqLCOM5_classV2) <- c('0-1', '1-2', '2-3', '3-4', '4-5', '5-
10', '10-15', '15-20', '20-25', '25-30', '30-35', '35-40', '40-45', '45-55')
rownames(freqLCOM5_classV3) <- c('0-1', '1-2', '2-3', '3-4', '4-5', '5-
10', '10-15', '15-20', '20-25', '25-30', '30-35', '35-40', '40-45', '45-55')

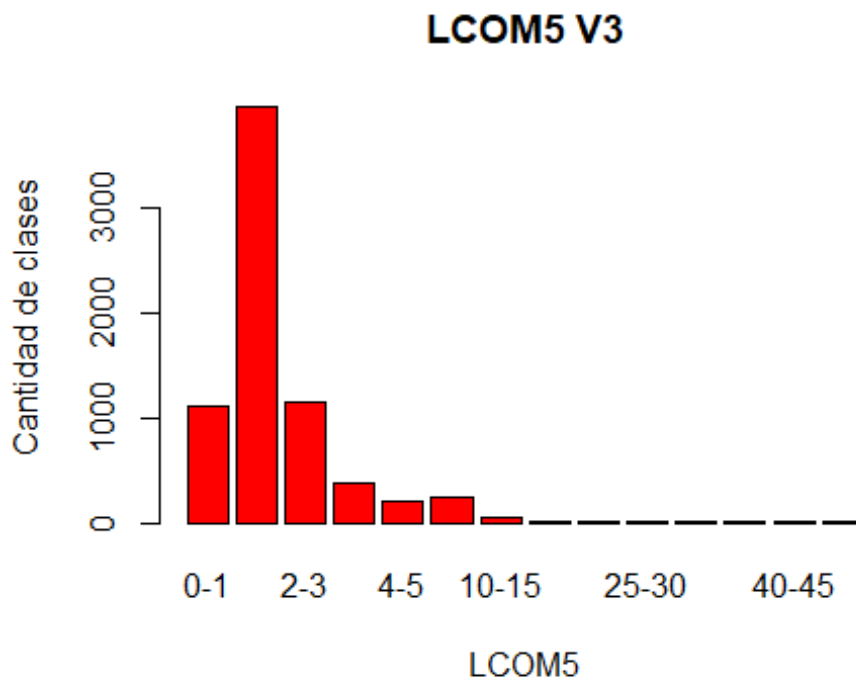
barplot(freqLCOM5_classV1,
        main = 'LCOM5 V1',
        col = 'lightblue',
        xlab = 'LCOM5',
        ylab = 'Cantidad de clases ')
```



```
barplot(freqLCOM5_classV2,  
        main = 'LCOM5 V2',  
        col = 'blue',  
        xlab = 'LCOM5',  
        ylab = 'Cantidad de clases')
```



```
barplot(freqLCOM5_classV3,  
        main = 'LCOM5 V3',  
        col = 'red',  
        xlab = 'LCOM5',  
        ylab = 'Cantidad de clases')
```

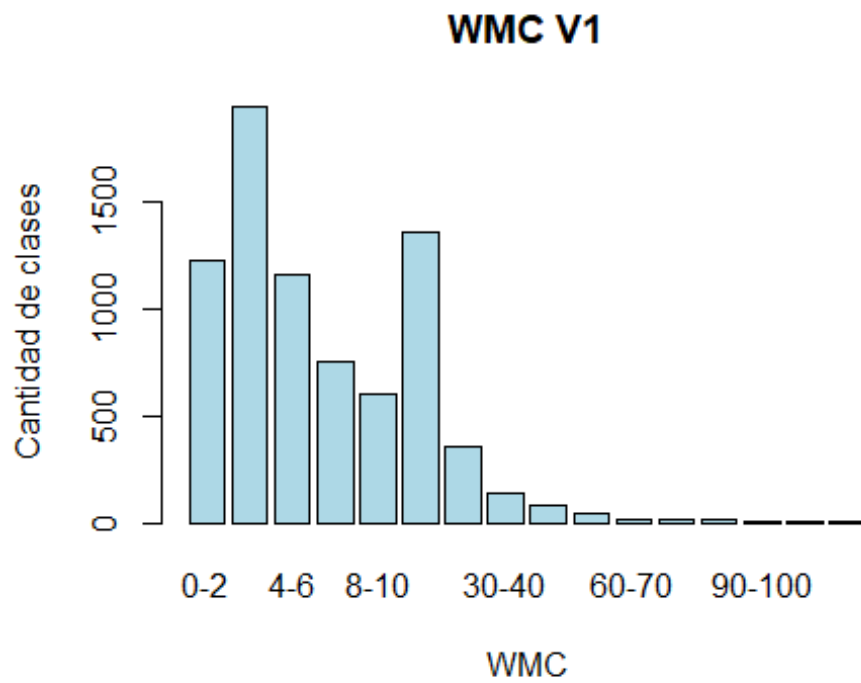


#### Gráficas de barras con el WMC de las clases de las 3 versiones

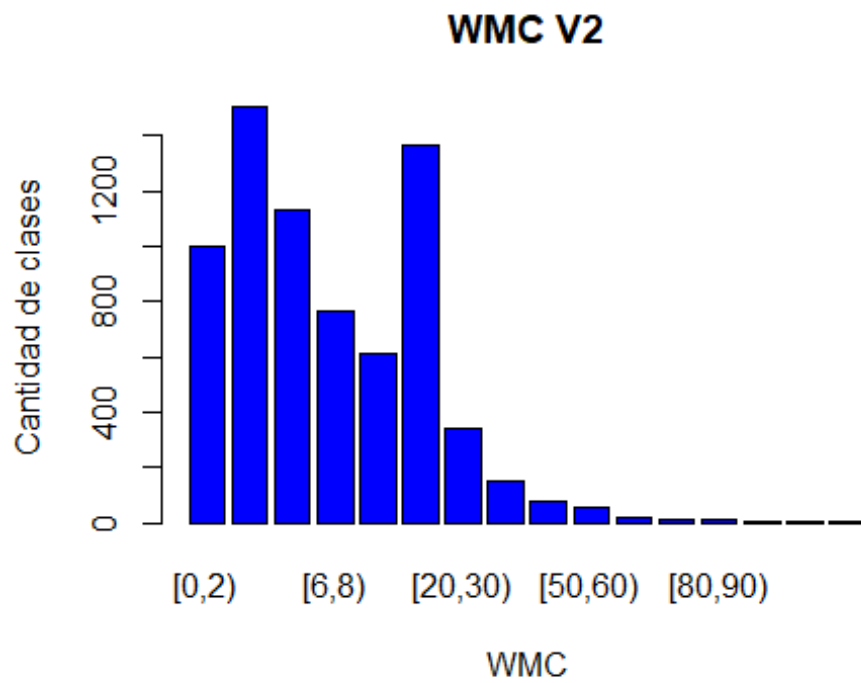
```
freqWMC_classV1 <- table(cut(WMC_classV1, breaks =
c(0,2,4,6,8,10,20,30,40,50,60,70,80,90,100,110,125), right = F,
include.lowest = T), exclude = NULL)
freqWMC_classV2 <- table(cut(WMC_classV2, breaks =
c(0,2,4,6,8,10,20,30,40,50,60,70,80,90,100,110,125), right = F,
include.lowest = T), exclude = NULL)
freqWMC_classV3 <- table(cut(WMC_classV3, breaks =
c(0,2,4,6,8,10,20,30,40,50,60,70,80,90,100,110,125), right = F,
include.lowest = T), exclude = NULL)

rownames(freqWMC_classV1) <- c('0-2', '2-4', '4-6', '6-8', '8-10', '10-
20', '20-30', '30-40', '40-50', '50-60', '60-70', '70-80', '80-90', '90-
100', '100-110', '110-125')
rownames(freqWMC_classV1) <- c('0-2', '2-4', '4-6', '6-8', '8-10', '10-
20', '20-30', '30-40', '40-50', '50-60', '60-70', '70-80', '80-90', '90-
100', '100-110', '110-125')
rownames(freqWMC_classV1) <- c('0-2', '2-4', '4-6', '6-8', '8-10', '10-
20', '20-30', '30-40', '40-50', '50-60', '60-70', '70-80', '80-90', '90-
100', '100-110', '110-125')

barplot(freqWMC_classV1,
main = 'WMC V1',
col = 'lightblue',
xlab = 'WMC',
ylab = 'Cantidad de clases ')
```

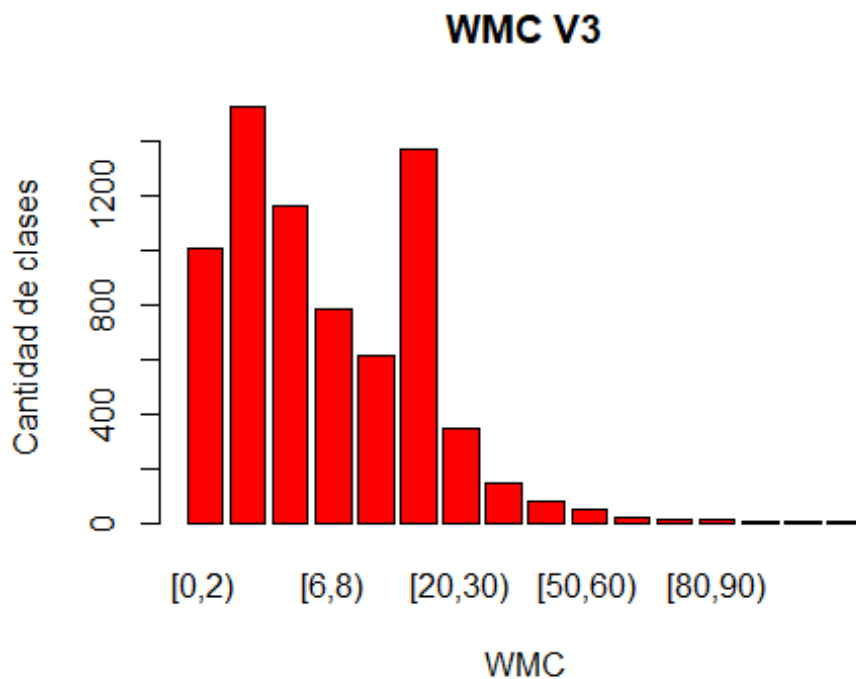


```
barplot(freqWMC_classV2,  
        main = 'WMC V2',  
        col = 'blue',  
        xlab = 'WMC',  
        ylab = 'Cantidad de clases')
```



```
barplot(freqWMC_classV3,  
        main = 'WMC V3',  
        col = 'red',  
        xlab = 'WMC',  
        ylab = 'Cantidad de clases')
```





### Medias de las metricas que emplearé

Medias de todas las columnas que voy a emplear para analizar el código: CBO, LOC, LCOM5 y WMC para las métricas de clases.

- LOC -> La media de las líneas de código de cada version analizada.
- CBO -> La media del CBO de cada verion analizada. Cuanto más acoplamiento, mayor CBO, se da en una clase, más difícil será reutilizarla.

Se propone como indicador del esfuerzo necesario para el mantenimiento y en el testeo.

- LCOM5 -> La media del LCOM5.

Se puede crear un indicador ya que un valor alto de LCOM implica falta de cohesión, es decir, escasa similitud de los métodos.

- WMC -> La media del WMC de las versiones analizadas. Mide la complejidad de una clase. Se debe mantener un valor MWC lo más bajo posible.

Según he visto en apuntes de clase, se sugiere un umbral de 40 o 20, dependiendo si las clases son o no de interface de usuario respectivamente. Crearé un indicador en base a estos datos.

```
mediaLOC_ClassV1 <- mean(LOC_classV1)
mediaLOC_ClassV2 <- mean(LOC_classV2)
mediaLOC_ClassV3 <- mean(LOC_classV3)
```

```
mediaCBO_ClassV1 <- mean(CBO_classV1)
mediaCBO_ClassV2 <- mean(CBO_classV2)
mediaCBO_ClassV3 <- mean(CBO_classV3)

mediaLCOM5_ClassV1 <- mean(LCOM5_classV1)
mediaLCOM5_ClassV2 <- mean(LCOM5_classV2)
mediaLCOM5_ClassV3 <- mean(LCOM5_classV3)

mediaWMC_ClassV1 <- mean(WMC_classV1)
mediaWMC_ClassV2 <- mean(WMC_classV2)
mediaWMC_ClassV3 <- mean(WMC_classV3)
```

## Indicadores

En el siguiente apartado se crearan y explicarán 3 indicadores, que según mi criterio, tienen sentido.

### Indicador CBO

Tal como se ha visto en los apartados anteriores, cuanto mayor CBO, mayor acoplamineto. Esto quiere decir que las clases con un CBO alto necesitan un mayor esfuerzo para su mantenimiento y testeo.

La media del CBO en las 3 versiones es :

```
mediaCBO_ClassV1
## [1] 6.402133
mediaCBO_ClassV2
## [1] 6.693663
mediaCBO_ClassV3
## [1] 6.681186
```

Tambien observo que el máximo de CBO en las 3 versiones es:

```
max_CBO_ClassV1
## [1] 321
max_CBO_ClassV2
## [1] 225
max_CBO_ClassV3
## [1] 242
```

Estos datos tan elevados sólo están presentes en alrededor de 10 clases de cada version. Aún asi hay muchos valores altos. El indicador que propongo es que toda clase que doble la media de CBO se almacenará en un subconjunto para poder revisarse posteriormente.

Este indicador está hecho con la media del analisis de la versión 1.

```
mediaCBO_doble <- (mediaCBO_ClassV1 * 2)
Clases_a_Revisar_CBO_Alto <- datasetClassV1 [datasetClassV1$CBO >
mediaCBO_doble ,]
```

### Indicador LCOM

Medias de LCOM:

```
mediaLCOM5_ClassV1
## [1] 1.350501
mediaLCOM5_ClassV2
## [1] 1.409491
mediaLCOM5_ClassV3
## [1] 1.543628
```

Como se puede observar la media de LCOM es de 1.35 - 1.40 - 1.54 en las 3 versiones analizadas. Un valor alto de LCOM implica falta de cohesión, es decir, escasa similitud de los métodos.

Al analizar los datos, ordenados por el valor de LCOM, he podido ver que el valor máximo es 51 y va descendiendo hasta llegar a un valor de 5 a partir del elemento 175 de 7500 elementos. Esto es un 2.33% de las clases analizadas.

Viendo los datos anteriores, propongo un indicador que guarde las clases, que superen el 5 como valor de LCOM en un subconjunto para poder estudiar las clases a posterior

```
Clases_a_Revisar_LCOM_Alto <- datasetClassV1 [datasetClassV1$LCOM5 > 5 ,]
```

### Indicador WMC

Las medias del WMC de las 3 versiones analizadas son:

```
mediawMC_ClassV1
## [1] 8.026922
mediawMC_ClassV2
## [1] 8.683575
mediawMC_ClassV3
```

```
## [1] 8.668962
```

Tal como se puede ver en los apuntes de la asignatura, se sugiere un umbral de 40 o 20, dependiendo si las clases son o no de interface de usuario respectivamente.

Mediante este indicador se almacenarán en el subconjunto indicado todas las clases que tengan un WMC mayor a 20.

```
Clases_aRevisar_WMC_Alto <- datasetClassV1 [datasetClassV1$WMC > 20 ,]
```

### Datos tras la ejecucion de los indicadores

A continuación se pueden ver los subconjuntos creados a partir de la ejecución de los indicadores, sobre los datos analizados de la version 1.

```
Clases_a_Revisar_CBO_Alto
```

```
## # A tibble: 1,143 × 102
##   ID      Name  LongName Parent Component Path   Line Column EndLine
##   <chr>   <chr>   <chr>     <chr>   <chr>   <chr> <dbl>  <dbl>  <dbl>
##   <dbl>
## 1 L2852  XooPl... org.son... L124    L103    "C:\... 87      1      191
## 2
## 2 L2970  XooPl... org.son... L124    L103    "C:\... 37      1       67
## 2
## 3 L7787  DbCli... org.son... L46621  L103    "C:\... 91      1      523
## 2
## 4 L43505 MyBat... org.son... L46621  L103    "C:\... 151     1      336
## 2
## 5 L7866  DbTes... org.son... L46621  L103    "C:\... 62      1      354
## 2
## 6 L71974 SQDat... org.son... L46621  L103    "C:\... 58      1      236
## 2
## 7 L185366 AppSt... org.son... L1850... L103    "C:\... 46      1       81
## 2
## 8 L188688 Proce... org.son... L1850... L103    "C:\... 66      1      382
## 2
## 9 L188877 Sched... org.son... L1850... L103    "C:\... 53      1      490
## 2
## 10 L189383 Proce... org.son... L1850... L103    "C:\... 52      1      368
## 2
## # ... with 1,133 more rows, and 92 more variables: CC <dbl>, CCL <dbl>,
## #   CCO <dbl>, CI <dbl>, CLC <dbl>, CLLC <dbl>, LDC <dbl>, LLDC <dbl>,
## #   LCOM5 <dbl>, NL <dbl>, NLE <dbl>, WMC <dbl>, CBO <dbl>, CBOI
## #   <dbl>,
## #   NII <dbl>, NOI <dbl>, RFC <dbl>, AD <dbl>, CD <dbl>, CLOC <dbl>,
## #   DLOC <dbl>, PDA <dbl>, PUA <dbl>, TCD <dbl>, TCLOC <dbl>, DIT
## #   <dbl>,
## #   NOA <dbl>, NOC <dbl>, NOD <dbl>, NOP <dbl>, LLOC <dbl>, LOC <dbl>,
```

```
## # `NA` <dbl>, NG <dbl>, NLA <dbl>, NLG <dbl>, NLM <dbl>, NLPA <dbl>, ...
```

```
Clases_a_Revisar_LCOM_Alto
```

```
## # A tibble: 157 × 102
```

```
## ID Name LongName Parent Component Path Line Column EndLine  
EndColumn
```

```
## <chr> <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
<dbl>
```

```
## 1 L46716 Datab... org.son... L46621 L103 "C:\... 59 1 462  
2
```

```
## 2 L47957 Datab... org.son... L46621 L103 "C:\... 64 1 544  
2
```

```
## 3 L71694 DBSes... org.son... L46621 L103 "C:\... 55 1 522  
2
```

```
## 4 L72037 Pagin... org.son... L46621 L103 "C:\... 31 1 104  
2
```

```
## 5 L192858 Props... org.son... L1898... L103 "C:\... 38 1 219  
2
```

```
## 6 L195766 Markd... org.son... L1955... L103 "C:\... 26 1 120  
2
```

```
## 7 L224950 WsTes... org.son... L2138... L103 "C:\... 37 1 101  
2
```

```
## 8 L42546 CeTas... org.son... L9492 L103 "C:\... 33 1 105  
2
```

```
## 9 L43523 ReadO... org.son... L43931 L103 "C:\... 39 1 94  
2
```

```
## 10 L45028 Stand... org.son... L43740 L103 "C:\... 66 3 98  
4
```

```
## # ... with 147 more rows, and 92 more variables: CC <dbl>, CCL <dbl>, CCO <dbl>,  
CI <dbl>,
```

```
## # CI <dbl>, CLC <dbl>, CLLC <dbl>, LDC <dbl>, LLDC <dbl>, LCOM5 <dbl>,  
NL <dbl>,
```

```
## # NL <dbl>, NLE <dbl>, WMC <dbl>, CBO <dbl>, CBOI <dbl>, NII <dbl>,  
NOI <dbl>,
```

```
## # NOI <dbl>, RFC <dbl>, AD <dbl>, CD <dbl>, CLOC <dbl>, DLOC <dbl>,  
PDA <dbl>,
```

```
## # PDA <dbl>, PUA <dbl>, TCD <dbl>, TCLOC <dbl>, DIT <dbl>, NOA <dbl>,  
NOC <dbl>,
```

```
## # NOC <dbl>, NOD <dbl>, NOP <dbl>, LLOC <dbl>, LOC <dbl>, `NA` <dbl>,  
NG <dbl>,
```

```
## # NG <dbl>, NLA <dbl>, NLG <dbl>, NLM <dbl>, NLPA <dbl>, NLPM <dbl>, ...
```

```
Clases_aRevisar_WMC_Alto
```

```
## # A tibble: 605 × 102
```

```
## ID Name LongName Parent Component Path Line Column EndLine  
EndColumn
```

```
## <chr> <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
<dbl>
```

```
## 1 L46716 Databa... org.son... L46621 L103 "C:\... 59 1 462
```

```

2
## 2 L43502 Defaul... org.son... L46621 L103 "C:\... 51 1 208
2
## 3 L47957 Databa... org.son... L46621 L103 "C:\... 64 1 544
2
## 4 L48184 Abstra... org.son... L46621 L103 "C:\... 63 1 523
2
## 5 L48417 CoreTe... org.son... L46621 L103 "C:\... 52 1 197
2
## 6 L49149 BatchS... org.son... L46621 L103 "C:\... 33 1 213
2
## 7 L7787 DbClie... org.son... L46621 L103 "C:\... 91 1 523
2
## 8 L49150 DbSess... org.son... L46621 L103 "C:\... 32 1 199
2
## 9 L54854 Delega... org.son... L46621 L103 "C:\... 36 1 210
2
## 10 L71694 DBSess... org.son... L46621 L103 "C:\... 55 1 522
2
## # ... with 595 more rows, and 92 more variables: CC <dbl>, CCL <dbl>,
CCO <dbl>,
## # CI <dbl>, CLC <dbl>, CLLC <dbl>, LDC <dbl>, LLDC <dbl>, LCOM5
<dbl>,
## # NL <dbl>, NLE <dbl>, WMC <dbl>, CBO <dbl>, CBOI <dbl>, NII <dbl>,
## # NOI <dbl>, RFC <dbl>, AD <dbl>, CD <dbl>, CLOC <dbl>, DLOC <dbl>,
## # PDA <dbl>, PUA <dbl>, TCD <dbl>, TCLOC <dbl>, DIT <dbl>, NOA
<dbl>,
## # NOC <dbl>, NOD <dbl>, NOP <dbl>, LLOC <dbl>, LOC <dbl>, `NA`
<dbl>,
## # NG <dbl>, NLA <dbl>, NLG <dbl>, NLM <dbl>, NLPA <dbl>, NLPM <dbl>,
...

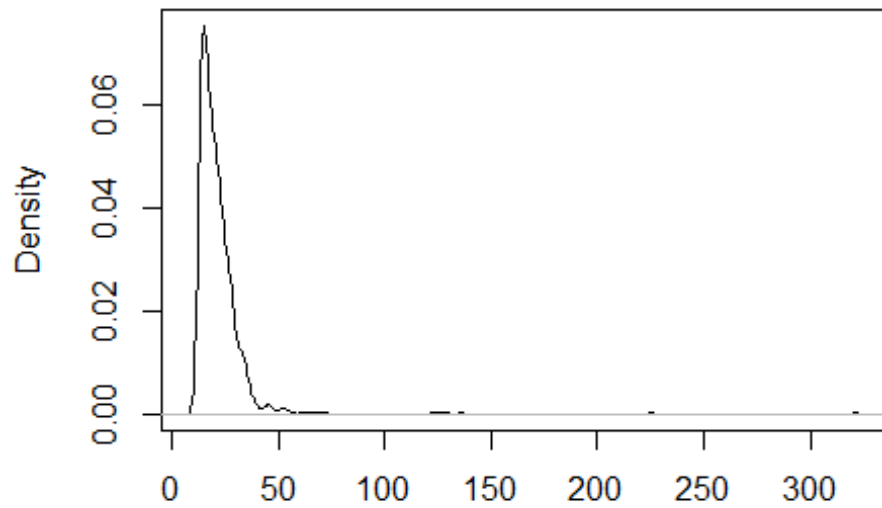
```

Algunas formas de analizar los datos obtenidos serían un gráfico de densidad o un gráfico de sectores.

Densidad del CBO en las clases maracadas como CBO alto.

```
plot(density(Clases_a_Revisar_CBO_Alto$CBO), main="Densidad de CBO")
```

## Densidad de CBO



N = 1143 Bandwidth = 1.643

```
freqLCOM_alto_classV1 <- table(cut(Clases_a_Revisar_LCOM_Alto$LCOM5,  
breaks = c(0,2,4,6,8,10,20,30,40,55), right = F, include.lowest = T),  
exclude = NULL)  
  
rownames(freqLCOM_alto_classV1) <- c('0-2', '2-4', '4-6', '6-8', '8-10', '10-  
20', '20-30', '30-40', '40-55')  
pie(table(freqLCOM_alto_classV1), main="LCOM")
```

## LCOM

