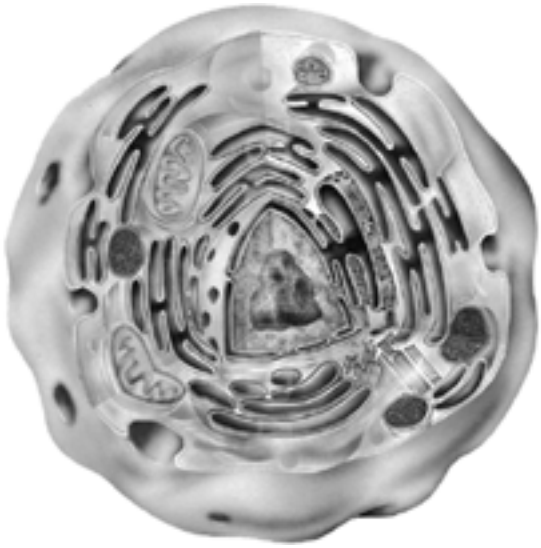

Figure 1:

Sobre la Base de datos

La autoría de esta Base de datos es de William Wolberg, Nick Street y Olvi Mangasarian. Las variables han sido calculadas gracias a la imagen digitalizada de una biopsia del tejido mamario y describen características del núcleo de las células. Las técnicas usadas están descritas en el artículo de K.P. Bennett y O.L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34

La base de datos ha sido descargada de la web Kaggle.com, en la dirección: http://bit.ly/2qse1el. También está disponible a través del servidor ftp de la UW CS. También puede encontrarse en el repositorio de Machine Learning de la UCI en la dirección: http://bit.ly/1L1zT4y

Información de las variables:
1) ID
2) Diagnóstico (M = Maligno, B = Benigno)
3-32) Diez características calculadas para cada núcleo de la célula:

a) El radio (radius) - La media de la distancia desde el centro hasta los puntos del perímetro
b) La textura (texture) - La desviación estándar de los valores en escala de grises
c) El perímetro (perimeter)
d) El área (area)
e) La suavidad (smoothness) - Variación local en las longitudes del radio
f) Compactación (compactness) - Perímetro^2/(área-1)
g) Concavidad (concavity) - Gravedad de las partes cóncavas del contorno
h) Puntos cóncavos (concave points) -Número de porciones cóncavas del contorno
i) Simetría (symmetry)
j) Dimensión fractal (fractal dimensión) - "Aproximación de los bordes costeros"-1

La media, el error estándar y la peor o la más grande de cada una de estas características han sido calculadas por ordenador para cada imagen, resultando en 30 variables diferentes. Todos los valores aparecen con cuatro cifras significativas. No hay valores perdidos. En total hay 357 casos benignos y 212 malignos.

K-MEANS

Aunque la finalidad del algoritmo K-Means no es la de la clasificación, la base de datos incorpora una variable que diferencia entre tumores benignos y malignos. Por ello voy a usar el algoritmo con ésta finalidad. Mi idea es acabar encontrando la configuración que mejor clasifique los datos. Que maximice los aciertos.

Lo primero que hago es borrar la memoria de R, cargar la base de datos y eliminar las variables cualitativas id y diagnosis:

```
datos <- read.csv(text=getURL("https://raw.githubusercontent.com/mrverde/cancer_wisconsin_DM_I/master/da
                  header = TRUE, stringsAsFactors = FALSE)
bdkmeans <- datos[ -c(1:2) ]
```

Para ver como se relacionan las variables entre sí, hago un diagrama de dispersión con el siguiente comando:

```
#plot(bdkmeans[,]) Silencio el gráfico porque es muy grande y da error. Lo adjunto aparte.
```

El resultado del diagrama de dispersión muestra una gran varianza entre algunas de las variables, así como la existencia de valores atípicos que pueden influir negativamente en nuestro modelo de clasificación.
Para eliminar estos efectos voy a probar diferentes transformaciones de los datos: obteniendo su logaritmo, estandarizándolos y eliminando los outliers igualando su valor a los datos de determinados percentiles.

```
#Calculo del logaritmo
bdkmeans <- log10(bdkmeans)
#Estandarizador
bdkmeans <- scale(bdkmeans)
bdkmeans <- data.frame(bdkmeans[,colSums(is.na(bdkmeans))<nrow(bdkmeans)])
#Eliminador de outliers
percentilup <- 85
percentildown <- 15
for (columna in colnames(bdkmeans)){
  up <-c("bdkmeans$",columna,"[bdkmeans$",columna,">quantile(bdkmeans$",columna,",",
         percentilup*0.01,")] <- quantile(bdkmeans$",columna,",", percentilup*0.01,")")
  down <-c("bdkmeans$",columna,"[bdkmeans$",columna,"<quantile(bdkmeans$",columna,", ",
           percentildown*0.01,")] <- quantile(bdkmeans$",columna,", ",
           percentildown*0.01,")")
  up <-paste(up, collapse="")
  down <-paste(down, collapse="")
  eval(parse(text=up))
  eval(parse(text=down))
}
```

En este punto si volvemos a dibujar el diagrama de dispersión y podemos ver los cambios:

```
#plot(bdkmeans[,]) Silencio el gráfico porque es muy grande y da error. Lo adjunto aparte.
```

Podemos comprobar en el diagrama de dispersión, como con los outliers al 5% los datos son heterocedásticos, y cómo se transforman en datos homocedásticos cuando eliminamos los outliers al 15%. En este punto ya los datos parecen correctos y procedo a hacer el K-Means. En cuanto al número de grupos le indico que cree dos, para ver si es capaz de clasificar por un lado los tumores malignos y por el otro los benignos. El K-Means es un algoritmo iterativo que resuelve un problema de optimización.

```
kmeans_cancer <-  kmeans(bdkmeans,2, iter.max =1000, nstart = 1000)
```

El resultado que devuelve es un mínimo local que no nos garantiza que la solución sea la mejor globalmente, por ello pueden obtenerse soluciones diferentes en función del punto en el que comencemos a iterar. Para intentar alcanzar un mínimo global, le indico que comience a iterar en mil puntos diferentes y que haga un máximo de mil iteraciones en cada uno de los inicios.

Creando una tabla de contingencia podemos comprobar cómo ha clasificado el algoritmo los grupos y su correspondencia con la diagnosis.

```
tabla <- table(datos$diagnosis, kmeans_cancer$cluster)
tabla
```

```
##
##       1   2
##   B 347  10
##   M  22 190
```

Después de esto podemos calcular el porcentaje de acierto que tuvo el algoritmo a la hora de agrupar los casos en los grupos benigno y maligno.

```
aciertos <- max(tabla[1,]) + max(tabla[2,])
fallos <- min(tabla[1,])+ min(tabla[2,])
por_ac = aciertos*100/569
por_fa = 100-por_ac
por_ac
```

```
## [1] 94
```

Este caso en concreto es el que mejor clasifica los datos. Hice además de éste otras pruebas con diferentes ajustes que paso a resumir en la siguiente tabla:

Normalizados	Logaritmo	Sin outliers al 5%	Sin outliers al 10%	Sin outliers al 15%	% de acierto
					85,41%
1					91,03%
2	1				93,05%
		1			86,82%
			1		87,34%
				1	89,10%
1		2			91,74%
1			2		93,67%
1				2	93,84%
2		1			91,22%
2			1		92,44%
2				1	93,32%
3	2	1			92,97%
3	2		1		93,14%
3	2			1	92,97%
2	1	3			93,50%
2	1		3		94,03%
2	1			3	94,37%**

Los números identifican el orden en el que se hicieron las transformaciones. **El modelo con mayor porcentaje de acierto
El mejor resultado se obtiene con las siguientes transformaciones de los datos:
1) La aplicación del logaritmo
2) La normalización de los datos
3) La eliminación de los outliers al 15%

Por último, he hecho una función que sirve para clasificar los nuevos casos. La función es muy simple, simplemente coge la posición de los centroides que devuelve el K-Means y comprueba cuál es el centroide más próximo: el de los tumores malignos o el de los tumores benignos. El código de la función es el siguiente:

```
clasificador <- function(nuevo_punto){
  if (dist(rbind(kmeans_cancer$centers[1],nuevo_punto))
      <dist(rbind(kmeans_cancer$centers[2],nuevo_punto))){
    return ("M")

  }else if (dist(rbind(kmeans_cancer$centers[2],nuevo_punto))
            <dist(rbind(kmeans_cancer$centers[1],nuevo_punto))){
    return ("B")
  }
}
```

Y podemos comprobar cómo clasifica los puntos con cualquier entrada de la base de datos:

```
clasificador(bdkmeans[100,])
```

```
## [1] "M"
```

```
clasificador(bdkmeans[507,])
```

```
## [1] "B"
```

Como puede comprobarse, el clasificador final que hemos obtenido es bastante bueno, ya que es capaz de acertar un 94,53% de los casos. En un trabajo posterior habría que mejorar el modelo e intentar reducir los casos en los que el modelo clasifica como benignos tumores malignos. Si bien estos casos son muy pocos -1,76%, 10 casos sobre un total de 569-, estos errores pueden ser fatales para la persona que los sufre. Estaríamos reduciendo sus posibilidades de sobrevivir. A su vez, sería interesante comprobar si esta clasificación errónea se debe a que estos tumores malignos se encuentran en un estado inicial de desarrollo que motiva sus valores atípicos. Otra posible vía de estudio sería centrarse en cada una de las variables de forma individual, y ver si alguna de ellas arroja valores que puedan alertarnos de que el tumor es maligno.