Junio 2018

Cristina Gil Martínez

# ANÁLISIS DE COMPONENTES PRINCIPALES

Apuntes personales sobre análisis de componentes principales, con un ejemplo en R aplicado sobre datos de expresión génica.

# CONTENIDO

# INTRODUCCIÓN

El análisis de componentes principales (*principal component analysis*) o *PCA* es una de las técnicas de aprendizaje **no supervisado**, las cuales suelen aplicarse como parte del análisis exploratorio de los datos. A diferencia de los métodos de aprendizaje supervisado, donde contamos con un grupo de variables o características ($X_1$, $X_2$, ..., $X_p$) medidas sobre un conjunto de observaciones *n*, con la intención de obtener predicciones sobre una variable respuesta *Y* asociada, en los no supervisados solo contamos con un número de variables de las cuales nos interesa conocer o de las que queremos extraer información, por ejemplo, sobre la existencia de subgrupos entre las variables u observaciones.

Una de las aplicaciones de *PCA* es la **reducción de dimensionalidad** (variables), perdiendo la menor cantidad de información (varianza) posible: cuando contamos con un gran número de variables cuantitativas posiblemente correlacionadas (indicativo de existencia de información redundante), *PCA* permite reducirlas a un número menor de variables transformadas (componentes principales) que expliquen gran parte de la variabilidad en los datos. Cada dimensión o componente principal generada por *PCA* será una **combinación lineal** de las variables originales, y serán además independientes o no correlacionadas entre sí. Los componentes principales generados pueden utilizarse a su vez en métodos de aprendizaje supervisado, como regresión de componentes principales o *partial least squares* (ver capítulo Métodos de Regularización*)*.

*PCA* también sirve como herramienta para la **visualización de datos**, mediante la reducción de la dimensionalidad. Supóngase que quisiéramos representar *n* observaciones con medidas sobre *p* variables ($X_1$, $X_2$, ..., $X_p$) como parte de un análisis exploratorio de los datos. Lo que podríamos hacer es examinar representaciones bidimensionales, sin embargo, existen un total de $\binom{p}{2} = p(p-1)/2$ posibles representaciones entre pares de variables, y si el número de variables es muy alto, estas representaciones se harían inviables, además de que posiblemente la información contenida en cada una sería solo una pequeña fracción de la información total contenida en los datos.

*PCA* puede considerarse como una rotación de los ejes del sistema de coordenadas de las variables originales a nuevos ejes ortogonales, de manera que estos ejes coincidan con la dirección de máxima varianza de los datos.

*NOTA: PCA no requiere la suposición de normalidad multivariante de los datos.*

# Álgebra matricial

A continuación se explican de manera básica dos conceptos sobre álgebra matricial requerida en *PCA*, para entender mejor el cálculo de los componentes principales: **eigenvectores** y **eigenvalores** de una matriz.

## EIGENVECTORES Y EIGENVALORES

Los eigenvectores y eigenvalores corresponden a números y vectores asociados a matrices cuadradas. Dada una matriz *A* de *n* x *n*, su eigenvector ($\vec{v}$) es una matriz *n* x 1 tal que

$$A \cdot \vec{v} = \lambda \cdot \vec{v}$$

donde el número λ es el eigenvalor, un valor escalar real asociado con el eigenvector.

Siempre que sean compatibles en tamaño, podemos multiplicar dos matrices entre sí. Los **eigenvectores**, autovectores o vectores propios son un caso especial de esta operación entre una matriz y un vector, siendo los eigenvectores de una matriz todos aquellos que, al ser multiplicados por esta matriz, resulten en el mismo vector o en un múltiplo entero de él. Considerando el siguiente ejemplo

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 6 \\ 4 \end{pmatrix} = \begin{pmatrix} 24 \\ 16 \end{pmatrix} = 4 \times \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

el vector resultante $\binom{24}{16}$ es múltiplo entero del vector original: 4 x $\binom{6}{4}$, lo que es igual a decir que el vector resultante es 4 veces el vector original. Es por tanto, un eigenvector de la matriz $\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$.

Entre las **propiedades** de los eigenvectores se encuentran:

- Solo las matrices cuadradas tienen eigenvectores, pero no todas las matrices cuadradas los tienen. Dada una matriz *n* x *n* con eigenvectores, el número existente de ellos es *n*.
- Un eigenvector escalado, es decir, si se multiplica por cierto valor antes de multiplicarlo por una matriz, el eigenvector continuará manteniendo su propiedad, ya que solo se cambia su longitud, no su dirección (es la dirección y no la longitud la que determina la propiedad de eigenvector de un determinado vector). Con respecto al ejemplo anterior, el vector se ha podido escalar de la siguiente forma:

$$2 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

- Independientemente del número de dimensiones, todos los eigenvectores de una matriz son perpendiculares. Esto significa que podemos expresar los datos respecto a estos eigenvectores.

Es frecuente escalar los eigenvectores para que tengan una longitud de 1, de manera que todos tengan la misma longitud. Siguiendo con el ejemplo anterior,

$$\begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

es un eigenvector, cuya longitud es

$$\sqrt{(3^2 + 2^2)} = \sqrt{13}$$

El escalado de este eigenvector se llevaría a cabo dividiéndolo entre esta candidad:

$$\begin{pmatrix} 3 \\ 2 \end{pmatrix} \div \sqrt{13} = \begin{pmatrix} 3/\sqrt{13} \\ 2/\sqrt{13} \end{pmatrix}$$

Cada uno de los componentes principales generados por *PCA* se corresponde a un eigenvector (dirección).

Por otro lado, los **eigenvalores** o valores propios son los valores con los que se multiplica el eigenvector y que dan lugar al vector original. En el ejemplo anterior, el eigenvalor asociado al eigenvector se corresponde con el valor 4. Los eigenvalores miden la cantidad de variabilidad retenida por cada componente principal (siendo mayores para la primera componente principal que para el resto), por lo que pueden usarse para determinar el número de componentes principales a retener.

- Un eigenvalor > 1 indica que la componente principal explica más varianza de lo que lo hace una de las variables originales, estando los datos estandarizados.

## ESTANDARIZACIÓN DE LAS VARIABLES

El cálculo de los componentes principales depende de las unidades de medida empleadas en las variables. Es por tanto importante, antes de aplicar *PCA*, estandarizar las variables para que tengan **media 0 y desviación estándar 1**, ya que, de lo contrario, las variables con mayor varianza dominarían al resto, aunque en el caso en que las variables estén medidas en las mismas unidades, podemos optar por no

estandarizarlas. La estandarización se lleva a cabo restando a cada observación la media y dividiendo entre la desviación estándar de la variable a la que pertenece:

$$\frac{x_i - media(x)}{\mathrm{sd}(x)}$$

Esto contrasta con otras técnicas de aprendizaje supervisado y no supervisado donde la estandarización de las variables no tiene efecto sobre el resultado, como por ejemplo, en regresión lineal.

# CÁLCULO DE LOS COMPONENTES PRINCIPALES

Como se ha dicho anteriormente, los componentes principales son una combinación lineal normalizada de las variables originales de un set de datos. Al calcularse sobre variables estandarizadas, los componentes principales son autovectores que **se toman de la matriz de correlaciones** (donde los elementos de la diagonal son igual a 1), no de covarianzas (ya que con variables estandarizadas ambas matrices coinciden). Generalmente, se podrán obtener tantas componentes principales distintas como variables disponibles. La elección se realiza de manera que la primera componente principal sea la que mayor varianza recoja; la segunda debe recoger la máxima variabilidad no recogida por la primera, y así sucesivamente, eligiendo un número que recoja un porcentaje suficiente de varianza total.

El objetivo es identificar las combinaciones lineales que mejor representan las variables $X_1, ..., X_p$. Sean ($\mathbf{Z_1, Z_2, ..., Z_M}$) M < *p* combinaciones lineales de las *p* variables originales, es decir

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

donde $\boldsymbol{\phi_{1m}, \phi_{2m}, ..., \phi_{pm}}$ son las constantes o ***loadings*** de los componentes principales (por ejemplo, $\phi_{11}$ correspondería al primer *loading* de la primera componente principal). Los *loadings* dan idea sobre qué peso tiene cada variable en cada componente. Cada vector de *loadings* [$\phi_{1m}, \phi_{2m}, ..., \phi_{pm}$] , de longitud igual a *p*, define además la dirección en el espacio sobre el cual la varianza de los datos es mayor.

La combinación lineal se normaliza para no inflar la varianza, por lo que

$$\sum_{j=1}^{p} \phi_{j1}^2 = 1$$

es decir, la suma de cuadrados de los *loadings* se iguala a 1.

La **primera componente principal** ($Z_1$) es aquella cuya dirección refleja o contiene la mayor variabilidad en los datos (por lo que esta componente será la que más información contenga). Este vector define la línea lo más próxima posible a los datos y que minimiza la suma de las distancias perpendiculares entre cada dato y la línea representada por la componente (usando como medida de cercanía el promedio de la distancia euclídea al cuadrado):

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \ldots + \phi_{p1} x_{ip}$$

donde $\phi_{11}$ corresponde al primer *loading* de la primera componente principal.

En otras palabras, el vector de *loadings* de la primera componente principal resuelve el problema de optimización

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left( \sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

Si las variables no se estandarizaran antes de aplicar *PCA*, los *loadings* de las variables con mayor varianza serían muy altos con respecto al resto, además de que una componente principal estaría muy influenciada por esta variable.

La **segunda componente principal** ($Z_2$) será una combinación lineal de las variables, que recoja la segunda dirección con mayor varianza de los datos, pero que no esté correlacionada con $Z_1$. Esta condición es equivalente a decir que la dirección de $Z_2$ (vector $\phi_2$) ha de ser **perpendicular** u **ortogonal** respecto a $Z_1$ (vector $\phi_1$).

Una vez generados los componentes principales, se pueden representar uno frente a otro para obtener visualizaciones de los datos.

## PROPORCIÓN DE VARIANZA EXPLICADA

La varianza total presente en los datos se define matemáticamente como

$$\sum_{j=1}^{p} \text{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$$

mientras que la varianza explicada por el *m*-ésimo componente principal se corresponde con

$$\frac{1}{n}\sum_{i=1}^{n} z_{im}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2$$

por lo que, la proporción de varianza explicada (***PVE***) del *m*-ésimo componente principal es igual a

$$\frac{\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2}{\sum_{j=1}^{p}\sum_{i=1}^{n}x_{ij}^2}$$

siendo igual a un número positivo. La suma de todas las *PVE* de todas los *M* componentes principales será igual a 1.

En un *PCA* nos interesa conocer la proporción de varianza explicada por cada uno de los componentes principales, o dicho de otra manera, cuanta información presente en los datos se pierde por la proyección de las observaciones sobre los primeros componentes principales. Como se explicó anteriormente, cada eigenvalor se corresponde con la varianza del componente $Z_i$ definido por el eigenvector $\vec{v}_i$

$$Var(Z_i) = \lambda_i$$

por lo que la proporción de varianza total que explica la componente $Z_i$ será

$$\frac{\lambda_i}{\sum_{i=1}^{p}\lambda_i} = \frac{\lambda_i}{\sum_{i=1}^{p}Var(x_i)}$$

donde la suma de las varianzas de los componentes principales y las variables originales son iguales. Multiplicando esta proporción por 100 obtendríamos el porcentaje.

Sumando todos los autovalores $\lambda_i$ obtendremos la varianza total de todos los componentes

$$\sum_{i=1}^{p} Var(Z_i) = \sum_{i=1}^{p}\lambda_i$$

# Número óptimo de componentes principales

Partiendo de un set de datos con *n* observaciones y *p* variables, el número de componentes principales distintos será de

$$min(n-1, p)$$

No existe un método objetivo para escoger el número de componentes principales que son suficientes para un análisis, por lo que depende del juicio del analista y del problema en cuestión. Si explican suficiente variabilidad y el objetivo es la visualización de los datos, no se suelen escoger más de tres componentes principales, para así facilitar la representación gráfica y la interpretación. Si este no es el principal objetivo, o si lo que se pretende es determinar el número óptimo de componentes que expliquen un mínimo porcentaje de varianza o que queramos utilizar para un análisis supervisado, como por ejemplo regresión de componentes principales, una manera objetiva de determinar el número de componentes es mediante **validación cruzada**.

# EJEMPLO EN R

PCA:

```
library(stats)
```

- **prcomp()** -> Forma rápida de implementar PCA sobre una matriz de datos.
- **princomp()**

```
library(FactoMineR)
```

- **PCA()** -> PCA con resultados más detallados. Los valores ausentes se reemplazan por la media de cada columna. Pueden incluirse variables categóricas suplementarias. Estandariza automáticamente los datos.

```
library(factoextra)
```

- **get_pca()** -> Extrae la información sobre las observaciones y variables de un análisis PCA.
- **get_pca_var()** -> Extrae la información sobre las variables.
- **get_pca_ind()** -> Extrae la información sobre las observaciones.

*Visualizaciones*:

```
library(FactoMineR)
```

- **fviz_pca_ind()** -> Representación de observaciones sobre componentes principales.
- **fviz_pca_var()**-> Representación de variables sobre componentes principales.
- **fviz_screeplot()** -> Representación (gráfico barras) de eigenvalores.
- **fviz_contrib()**-> Representa la contribución de filas/columnas de los resultados de un pca.

Los métodos de análisis no supervisado suelen aplicarse para el análisis de datos genómicos. El set de datos `NCI60` contiene información sobre datos de microarray de líneas celulares de cáncer: medidas sobre expresión de 6830 genes (variables) sobre 64 líneas celulares cancerígenas (observaciones). El formato de este set es una lista con dos elementos: una matriz con los valores de expresión génica (*data*) y un vector con el nombre de los tipos de cáncer (*labs*).

En este ejemplo se muestra la aplicación de *PCA* para encontrar patrones o agrupaciones mediante la representación de las dos primeras componentes principales, que puede utilizarse en conjunción con posteriores análisis de *clustering*. Siendo este un análisis no supervisado, no haremos uso de la información sobre el tipo de cáncer de cada línea celular.

NOTA: Recordar que *PCA* solo puede aplicarse a datos numéricos. Si los datos contienen variables categóricas, deben ser convertidas a numéricas.

# Exploración de datos

```
library(ISLR)

names(NCI60)

## [1] "data" "labs"

datos.nci <- NCI60$data
dim(datos.nci)

## [1]   64 6830

head(datos.nci)[, 1:6]

 ##           1         2         3         4         5             6
## V1 0.300000  1.180000  0.550000  1.140000 -0.265000 -7.000000e-02
## V2 0.679961  1.289961  0.169961  0.379961  0.464961  5.799610e-01
## V3 0.940000 -0.040000 -0.170000 -0.040000 -0.605000  0.000000e+00
## V4 0.280000 -0.310000  0.680000 -0.810000  0.625000 -1.387779e-17
## V5 0.485000 -0.465000  0.395000  0.905000  0.200000 -5.000000e-03
## V6 0.310000 -0.030000 -0.100000 -0.460000 -0.205000 -5.400000e-01


# Tipos de cáncer distintos en el set de datos
unique(NCI60$labs)

 ##  [1] "CNS"         "RENAL"       "BREAST"      "NSCLC"       "UNKNOWN"
##  [6] "OVARIAN"     "MELANOMA"    "PROSTATE"    "LEUKEMIA"    "K562B-repro"
## [11] "K562A-repro" "COLON"       "MCF7A-repro" "MCF7D-repro"


# Número de muestras por tipo de cáncer
table(NCI60$labs)

 ##
##      BREAST         CNS       COLON K562A-repro K562B-repro    LEUKEMIA
##           7           5           7           1           1           6
## MCF7A-repro MCF7D-repro    MELANOMA       NSCLC     OVARIAN    PROSTATE
##           1           1           8           9           6           2
##       RENAL     UNKNOWN
##           9           1


 # Media de la expresión de cada gen (muestra de los 10 primeros).
# (MARGIN = 2 para que se aplique la función a las columnas)
apply(X = datos.nci, MARGIN = 2, FUN = mean)[1:10]

 ##             1             2             3             4             5
## -0.0190634141 -0.0278131014 -0.0199227886 -0.3286727886  0.0260928356
##             6             7             8             9            10
##  0.0067178370  0.0196865850 -0.0231259139  0.0007803366  0.0192372951
```

```
# Varianza de la expresión de cada gen (muestra de los 10 primeros)
apply(X = datos.nci, MARGIN = 2, FUN = var)[1:10]
```

```
 ##         1         2         3         4         5         6         7
8
## 0.1947740 0.5737041 0.1877537 1.1922566 0.2352962 0.1228028 0.1374062 0.1146699

##         9        10
## 0.1842025 0.4116286
```

# Cálculo de componentes principales

Es importante estandarizar las variables (genes) para que tengan desviación estándar igual a 1 antes de aplicar *PCA*, aunque sería razonable argumentar que es mejor no estandarizarlos.

Existen varias funciones con las que aplicar *PCA* en R. Algunos ejemplos se muestran a continuación:

## FUNCIÓN `prcomp()`

Por defecto, la función `prcomp()` centra las variables para que tengan media de 0. Con el argumento `scale = TRUE` indicamos que queremos escalar las variables para que tengan desviación estándar igual a 1.

```
pca.nci <- prcomp(datos.nci, scale = TRUE)

names(pca.nci)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

Los elementos `"center"` y `"scale"` se corresponden con las medias y las desviaciones estándar originales de las variables previo escalado e implementación del *PCA*. La matriz `"rotation"` proporciona los *loadings* de los componentes principales (cada columna contiene el vector de *loadings* de cada componente principal). La función los denomina matriz de rotación ya que si multiplicáramos la matriz de datos por `datos.genes$rotation`, obtendríamos las coordenadas de los datos en el nuevo sistema rotado de coordenadas. Estas coordenadas se corresponden con los *scores* de los componentes principales.

*Nota: En álgebra lineal, una matriz de rotación es la matriz que representa una rotación en el espacio euclídeo.*

```r
# Muestra de los primeros 6 elementos del vector de loadings de los 5 primeros com
ponentes
head(pca.nci$rotation)[, 1:5]
```

```
##            PC1          PC2         PC3          PC4          PC5
## 1 -0.010682370  0.001324406 0.008503514 -0.003524094 -0.010126893
## 2 -0.002312078  0.001675266 0.010256593  0.002603645 -0.011400802
## 3 -0.005879750 -0.006289434 0.010055415 -0.010681458  0.010264980
## 4  0.003278071  0.002666138 0.008361513 -0.007475761  0.011248268
## 5 -0.007677535 -0.002508097 0.013820836  0.009509144  0.004094756
## 6  0.002266671 -0.009677933 0.010818283 -0.012751147 -0.007196820
```

```r
dim(pca.nci$rotation)
```

```
## [1] 6830   64
```

Hay un total de 64 componentes principales distintas, ya que en general pueden haber *min(n - 1, p)* componentes en un set de datos *n* x *p*. En este caso *min(*6829, 64*)* = 64.

Podemos acceder a los vectores de los *scores* de la siguiente manera:

```r
head(pca.nci$x)[,1:5]
```

```
##           PC1       PC2          PC3          PC4        PC5
## V1 -19.68245  3.527748  -9.7354382   0.8177816 -12.511081
## V2 -22.90812  6.390938 -13.3725378  -5.5911088  -7.972471
## V3 -27.24077  2.445809  -3.5053437   1.3311502 -12.466296
## V4 -42.48098 -9.691742  -0.8830921  -3.4180227 -41.938370
## V5 -54.98387 -5.158121 -20.9291076 -15.7253986 -10.361364
## V6 -26.96488  6.727122 -21.6422924 -13.7323153   7.934827
```

Otro de los *outputs* de la función `prcomp()` es la desviación estándar de cada componente principal:

```r
pca.nci$sdev
```

```
##  [1] 2.785347e+01 2.148136e+01 1.982046e+01 1.703256e+01 1.597181e+01
##  [6] 1.572108e+01 1.447145e+01 1.354427e+01 1.314400e+01 1.273860e+01
## [11] 1.268672e+01 1.215769e+01 1.183019e+01 1.162554e+01 1.143779e+01
## [16] 1.100051e+01 1.065666e+01 1.048880e+01 1.043518e+01 1.032194e+01
## [21] 1.014608e+01 1.005439e+01 9.902655e+00 9.647656e+00 9.507638e+00
## [26] 9.332529e+00 9.273200e+00 9.090046e+00 8.981173e+00 8.750031e+00
## [31] 8.599622e+00 8.447375e+00 8.373048e+00 8.215787e+00 8.157313e+00
## [36] 7.974655e+00 7.904462e+00 7.821271e+00 7.721562e+00 7.586035e+00
## [41] 7.456193e+00 7.344380e+00 7.104489e+00 7.013055e+00 6.958385e+00
## [46] 6.866265e+00 6.807439e+00 6.647630e+00 6.616068e+00 6.407926e+00
## [51] 6.219838e+00 6.203258e+00 6.067065e+00 5.918049e+00 5.912333e+00
## [56] 5.735386e+00 5.472610e+00 5.292148e+00 5.021174e+00 4.683979e+00
## [61] 4.175673e+00 4.082121e+00 4.041243e+00 2.148247e-14
```

La varianza explicada por cada componente principal la obtenemos elevando al cuadrado la desviación estándar:

```
# Varianza explicada por cada componente
pca.nci$sdev^2

##  [1] 7.758157e+02 4.614486e+02 3.928508e+02 2.901080e+02 2.550986e+02
##  [6] 2.471524e+02 2.094230e+02 1.834472e+02 1.727647e+02 1.622718e+02
## [11] 1.609529e+02 1.478095e+02 1.399534e+02 1.351533e+02 1.308230e+02
## [16] 1.210113e+02 1.135644e+02 1.100148e+02 1.088930e+02 1.065424e+02
## [21] 1.029429e+02 1.010908e+02 9.806257e+01 9.307726e+01 9.039519e+01
## [26] 8.709610e+01 8.599223e+01 8.262894e+01 8.066147e+01 7.656305e+01
## [31] 7.395349e+01 7.135815e+01 7.010794e+01 6.749915e+01 6.654176e+01
## [36] 6.359512e+01 6.248052e+01 6.117227e+01 5.962252e+01 5.754792e+01
## [41] 5.559482e+01 5.393991e+01 5.047377e+01 4.918294e+01 4.841912e+01
## [46] 4.714560e+01 4.634123e+01 4.419098e+01 4.377236e+01 4.106152e+01
## [51] 3.868639e+01 3.848041e+01 3.680928e+01 3.502331e+01 3.495568e+01
## [56] 3.289465e+01 2.994946e+01 2.800683e+01 2.521219e+01 2.193966e+01
## [61] 1.743625e+01 1.666371e+01 1.633165e+01 4.614964e-28
```

Como es de esperar, la varianza explicada es mayor en la primera componente que en las subsiguientes.


```
# Varianza explicada por cada componente
pca.nci$sdev^2

 ##  [1] 7.758157e+02 4.614486e+02 3.928508e+02 2.901080e+02 2.550986e+02
##  [6] 2.471524e+02 2.094230e+02 1.834472e+02 1.727647e+02 1.622718e+02
## [11] 1.609529e+02 1.478095e+02 1.399534e+02 1.351533e+02 1.308230e+02
## [16] 1.210113e+02 1.135644e+02 1.100148e+02 1.088930e+02 1.065424e+02
## [21] 1.029429e+02 1.010908e+02 9.806257e+01 9.307726e+01 9.039519e+01
## [26] 8.709610e+01 8.599223e+01 8.262894e+01 8.066147e+01 7.656305e+01
## [31] 7.395349e+01 7.135815e+01 7.010794e+01 6.749915e+01 6.654176e+01
## [36] 6.359512e+01 6.248052e+01 6.117227e+01 5.962252e+01 5.754792e+01
## [41] 5.559482e+01 5.393991e+01 5.047377e+01 4.918294e+01 4.841912e+01
## [46] 4.714560e+01 4.634123e+01 4.419098e+01 4.377236e+01 4.106152e+01
## [51] 3.868639e+01 3.848041e+01 3.680928e+01 3.502331e+01 3.495568e+01
## [56] 3.289465e+01 2.994946e+01 2.800683e+01 2.521219e+01 2.193966e+01
## [61] 1.743625e+01 1.666371e+01 1.633165e+01 4.614964e-28
```


Como es de esperar, la varianza explicada es mayor en la primera componente que en las subsiguientes.

```
summary(pca.nci)

## Importance of components%s:
##                              PC1      PC2      PC3      PC4      PC5      PC6
## Standard deviation     27.8535 21.48136 19.82046 17.03256 15.97181 15.72108
## Proportion of Variance  0.1136  0.06756  0.05752  0.04248  0.03735  0.03619
## Cumulative Proportion   0.1136  0.18115  0.23867  0.28115  0.31850  0.35468
##                              PC7      PC8      PC9     PC10     PC11     PC12
## Standard deviation     14.47145 13.54427 13.14400 12.73860 12.68672 12.15769
## Proportion of Variance  0.03066  0.02686  0.02529  0.02376  0.02357  0.02164
## Cumulative Proportion   0.38534  0.41220  0.43750  0.46126  0.48482  0.50646
##                             PC13     PC14     PC15     PC16     PC17     PC18
## Standard deviation     11.83019 11.62554 11.43779 11.00051 10.65666 10.48880
## Proportion of Variance  0.02049  0.01979  0.01915  0.01772  0.01663  0.01611
## Cumulative Proportion   0.52695  0.54674  0.56590  0.58361  0.60024  0.61635
```

```
##                            PC19    PC20     PC21    PC22    PC23    PC24
## Standard deviation     10.43518 10.3219 10.14608 10.0544 9.90265 9.64766
## Proportion of Variance  0.01594  0.0156  0.01507  0.0148 0.01436 0.01363
## Cumulative Proportion   0.63229  0.6479  0.66296  0.6778 0.69212 0.70575
##                            PC25    PC26    PC27   PC28    PC29    PC30    PC31
## Standard deviation     9.50764 9.33253 9.27320 9.0900 8.98117 8.75003 8.59962
## Proportion of Variance 0.01324 0.01275 0.01259 0.0121 0.01181 0.01121 0.01083
## Cumulative Proportion  0.71899 0.73174 0.74433 0.7564 0.76824 0.77945 0.79027
##                            PC32    PC33    PC34    PC35    PC36    PC37    PC38
## Standard deviation     8.44738 8.37305 8.21579 8.15731 7.97465 7.90446 7.82127
## Proportion of Variance 0.01045 0.01026 0.00988 0.00974 0.00931 0.00915 0.00896
## Cumulative Proportion  0.80072 0.81099 0.82087 0.83061 0.83992 0.84907 0.85803
##                            PC39    PC40    PC41   PC42    PC43   PC44    PC45
## Standard deviation     7.72156 7.58603 7.45619 7.3444 7.10449 7.0131 6.95839
## Proportion of Variance 0.00873 0.00843 0.00814 0.0079 0.00739 0.0072 0.00709
## Cumulative Proportion  0.86676 0.87518 0.88332 0.8912 0.89861 0.9058 0.91290
##                          PC46    PC47    PC48    PC49    PC50    PC51    PC52
## Standard deviation     6.8663 6.80744 6.64763 6.61607 6.40793 6.21984 6.20326
## Proportion of Variance 0.0069 0.00678 0.00647 0.00641 0.00601 0.00566 0.00563
## Cumulative Proportion  0.9198 0.92659 0.93306 0.93947 0.94548 0.95114 0.95678
##                            PC53    PC54    PC55    PC56    PC57   PC58    PC59
## Standard deviation     6.06706 5.91805 5.91233 5.73539 5.47261 5.2921 5.02117
## Proportion of Variance 0.00539 0.00513 0.00512 0.00482 0.00438 0.0041 0.00369
## Cumulative Proportion  0.96216 0.96729 0.97241 0.97723 0.98161 0.9857 0.98940
##                            PC60    PC61    PC62    PC63      PC64
## Standard deviation     4.68398 4.17567 4.08212 4.04124 2.148e-14
## Proportion of Variance 0.00321 0.00255 0.00244 0.00239 0.000e+00
## Cumulative Proportion  0.99262 0.99517 0.99761 1.00000 1.000e+00
```

## FUNCIÓN PCA()

```
library(FactoMineR)

pca2.nci <- PCA(X = datos.nci, scale.unit = TRUE, ncp = 64, graph = FALSE)
```

El objeto *PCA* generado contiene los siguientes elementos en forma de lista:

```
print(pca2.nci)

## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 64 individuals, described by 6830 variables
## *The results are available in the following objects:
##
##    name               description
## 1  "$eig"             "eigenvalues"
## 2  "$var"             "results for the variables"
## 3  "$var$coord"       "coord. for the variables"
## 4  "$var$cor"         "correlations variables - dimensions"
## 5  "$var$cos2"        "cos2 for the variables"
## 6  "$var$contrib"     "contributions of the variables"
## 7  "$ind"             "results for the individuals"
## 8  "$ind$coord"       "coord. for the individuals"
## 9  "$ind$cos2"        "cos2 for the individuals"
```

```
## 10 "$ind$contrib"     "contributions of the individuals"
## 11 "$call"            "summary statistics"
## 12 "$call$centre"     "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"      "weights for the individuals"
## 15 "$call$col.w"      "weights for the variables"
```

```
head(pca2.nci$eig)
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
## comp 1   775.8157              11.358942                           11.35894
## comp 2   461.4486               6.756203                           18.11514
## comp 3   392.8508               5.751842                           23.86699
## comp 4   290.1080               4.247554                           28.11454
## comp 5   255.0986               3.734972                           31.84951
## comp 6   247.1524               3.618630                           35.46814
```

Como es de esperar, el eigenvalor (varianza explicada) es mayor en la primera componente que en las subsiguientes.

# Representación

(Para ver ejemplos sobre las modificaciones posibles sobre las representaciones de un PCA, visitar el siguiente enlace).

Cabe destacar que la representación gráfica de las observaciones y las variables es distinta: las observaciones se representan mediante sus proyecciones, mientras que las variables se representan mediante sus correlaciones. La correlación entre un componente y una variable estima la información que comparten -> *loadings*, por lo que las variables se pueden representar como puntos en el espacio de los componentes utilizando sus *loadings* como coordenadas.

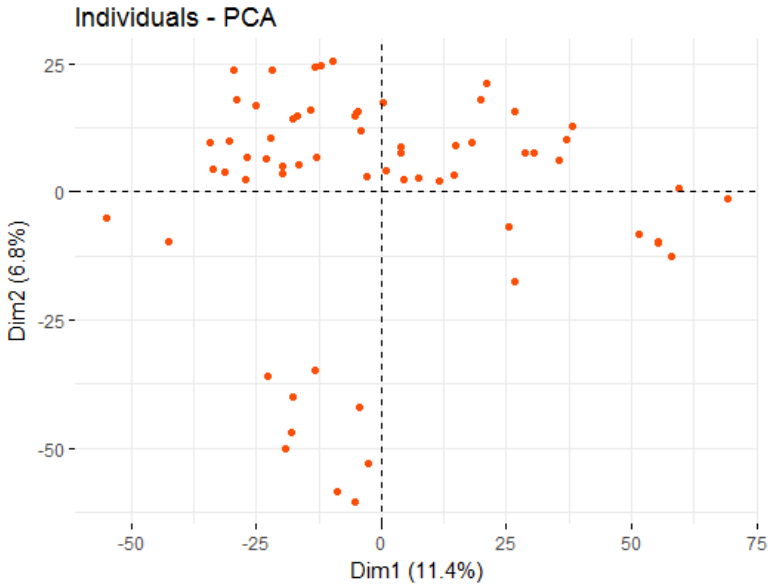
## OBSERVACIONES

Se muestran dos ejemplos para representar las observaciones sobre las dos primeras componentes principales:

```
library(factoextra)

fviz_pca_ind(pca.nci, geom.ind = "point",
             col.ind = "#FC4E07",
             axes = c(1, 2),
             pointsize = 1.5)

 # axes 1 y 2 se corresponden con PC1 y PC2, pudiendo escoger otros
```

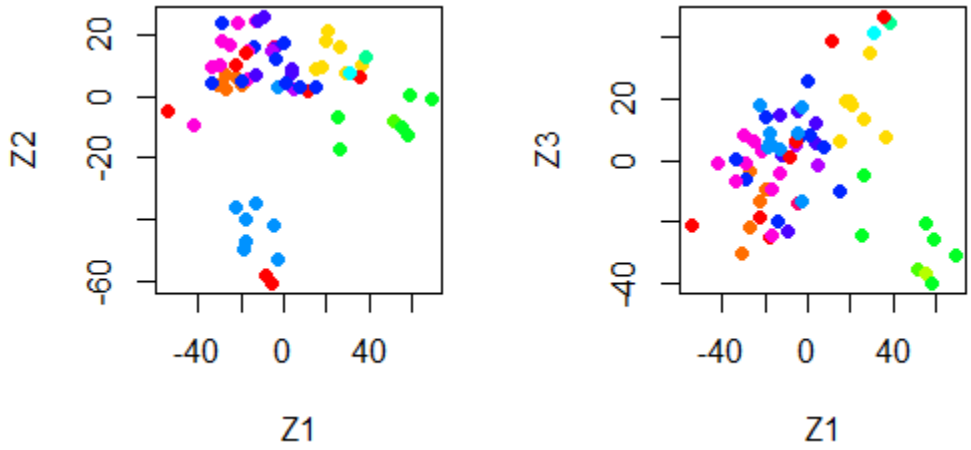



Teniendo información sobre las líneas correspondientes a cada tipo de cáncer, podemos representar cada grupo en un color distinto. Para esto creamos una función que asigne un color distinto a cada elemento de un vector numérico, para representar con colores cada una de las 64 líneas en base al tipo de cáncer de cada una:

```
colores <- function(vec){
# la función rainbow() devuelve un vector que contiene el número de colores distin
tos
col <- rainbow(length(unique(vec)))
return(col[as.numeric(as.factor(vec))])
}

par(mfrow = c(1,2))
# Observaciones sobre PC1 y PC2
plot(pca.nci$x[,1:2], col = colores(NCI60$labs),
     pch = 19,
     xlab = "Z1",
     ylab = "Z2")
# Observaciones sobre PC1 y PC3
plot(pca.nci$x[,c(1, 3)], col = colores(NCI60$labs),
     pch = 19,
     xlab = "Z1",
     ylab = "Z3")
```

En

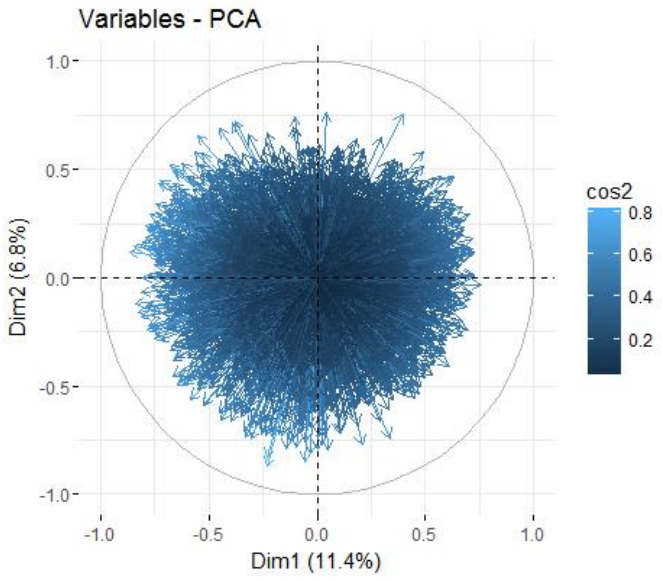



general, las observaciones que pertenecen a un tipo de cáncer tienden a situarse en una distancia próxima en esta representación de dos dimensiones con las tres primeras componentes principales. Esto indica que las líneas celulares del mismo tipo de cáncer tienden a tener unos niveles de expresión génica similares.

## VARIABLES

Para representar las variables sobre las dos primeras componentes principales podemos utilizar la función `fviz_pca_var()` del paquete `factoextra`. La correlación entre una variable y un componente principal se utiliza como la coordenada de dicha variable sobre el componente principal. De esta manera podemos obtener un gráfico de correlación de variables:

```
fviz_pca_var(pca.nci, col.var = "cos2",
             geom.var = "arrow",
             labelsize = 2,
             repel = FALSE)
```



En este tipo de gráfico, además de indicarse el % de varianza explicada por el primer (Dim1) y segundo componente (Dim2), las variables positivamente correlacionadas se agrupan juntas o próximas, mientras que las negativamente correlacionadas se representan en lados opuestos del origen o cuadrantes opuestos. Además, la distancia entre las variables y el origen mide la calidad de la representación de las variables (mayor cuanto más próxima a la circunferencia o círculo de correlación, siendo éstas las que más contribuyen en los dos primeros componentes). La calidad de esta representación se mide por el valor al cuadrado del coseno (*cos2*) del ángulo del triángulo formado por el punto del origen, la observación y su proyección sobre el componente. Para una variable dada, la suma del *cos2* sobre todos los componentes principales será igual a 1, y si además la variable es perfectamente representable por solo los dos primeros componentes principales, la suma de *cos2* sobre estos dos será igual a 1. Variables posicionadas cerca del origen puede ser un indicativo de que serían necesarios más de dos componentes principales para su representación.

Con la función `get_pca_var()` del paquete `factoextra` podemos extraer los resultados de las variables a partir de un objeto pca:

```
var <- get_pca_var(pca.nci)


var

 ## Principal Component Analysis Results for variables
##  ===================================================
##   Name       Description
## 1 "$coord"   "Coordinates for the variables"
## 2 "$cor"     "Correlations between variables and dimensions"
## 3 "$cos2"    "Cos2 for the variables"
## 4 "$contrib" "contributions of the variables"
```

El *cos2* de las variables y su contribución serían equivalentes.

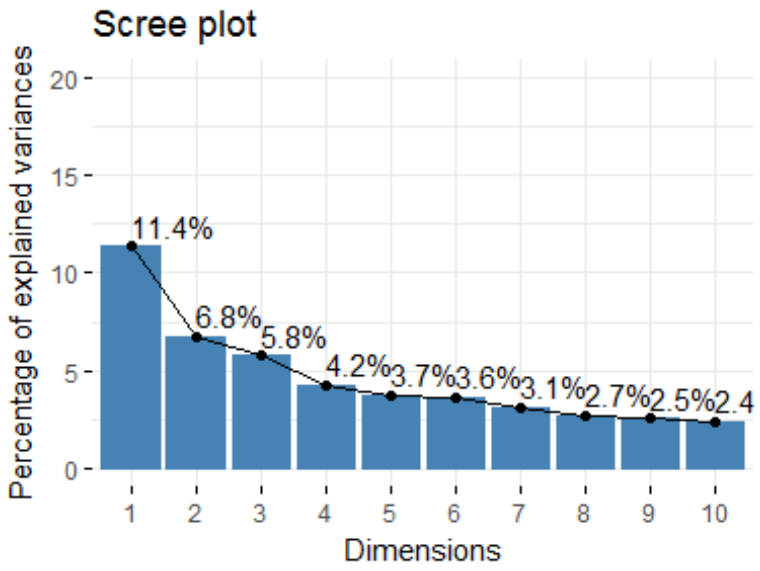
# Elección del número de componentes principales

Podemos hacer uso de multitud de representaciones a la hora de escoger el número óptimo de componentes principales:

## *SCREE PLOT*

Una forma es generando un *scree plot* que represente los eigenvalores ordenador de mayor a menor. Con la función `fviz_screeplot()` del paquete `factoextra` podemos obtener esta representación, sin importar qué función hemos utilizado para generar los componentes principales.

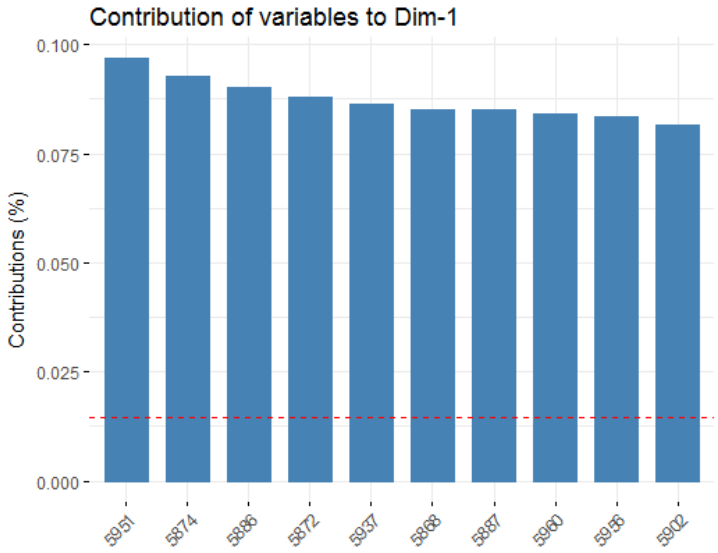
```
fviz_screeplot(pca.nci, addlabels = TRUE, ylim = c(0, 20))
```




## CONTRIBUCIÓN DE VARIABLES

Si contamos con un gran número de variables, podríamos decidir mostrar solo aquellas con mayor contribución.

```
# Top 10 variables que más contribuyen a PC1
fviz_contrib(pca.nci, choice = "var", axes = 1, top = 10)
```




La línea roja discontinua indica el valor medio de contribución. Para una determinada componente, una variable con una contribución mayor a este límite puede considerarse importante a la hora de contribuir a esta componente. En la representación anterior, el gen 5951 es la que más contribuye a la PC1.

## PROPORCIÓN DE VARIANZA EXPLICADA Y ACUMULADA

Para calcular la proporción de varianza explicada (*PVE*) por cada componente principal, simplemente dividimos la varianza explicada de cada uno entre la varianza total de todos:

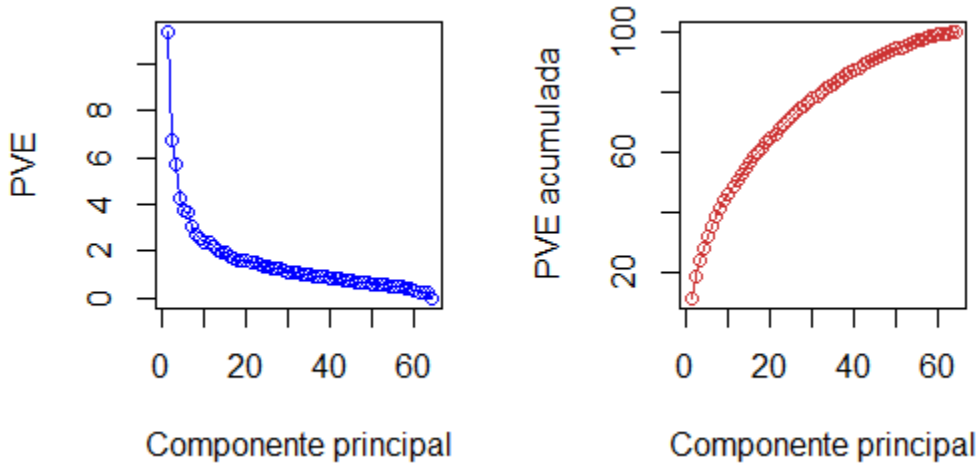
```
PVE <- 100*pca.nci$sdev^2/sum(pca.nci$sdev^2)


PVE
##  [1] 1.135894e+01 6.756203e+00 5.751842e+00 4.247554e+00 3.734972e+00
##  [6] 3.618630e+00 3.066222e+00 2.685903e+00 2.529498e+00 2.375869e+00
## [11] 2.356558e+00 2.164122e+00 2.049097e+00 1.978818e+00 1.915417e+00
## [16] 1.771761e+00 1.662730e+00 1.610759e+00 1.594333e+00 1.559919e+00
## [21] 1.507217e+00 1.480099e+00 1.435762e+00 1.362771e+00 1.323502e+00
## [26] 1.275199e+00 1.259037e+00 1.209794e+00 1.180988e+00 1.120982e+00
## [31] 1.082774e+00 1.044775e+00 1.026471e+00 9.882745e-01 9.742571e-01
## [36] 9.311145e-01 9.147953e-01 8.956409e-01 8.729506e-01 8.425758e-01
## [41] 8.139798e-01 7.897498e-01 7.390010e-01 7.201016e-01 7.089184e-01
## [46] 6.902723e-01 6.784953e-01 6.470130e-01 6.408838e-01 6.011935e-01
## [51] 5.664186e-01 5.634028e-01 5.389352e-01 5.127863e-01 5.117962e-01
## [56] 4.816201e-01 4.384987e-01 4.100561e-01 3.691390e-01 3.212249e-01
## [61] 2.552891e-01 2.439782e-01 2.391163e-01 6.756902e-30
```

La primera componente principal explica el 11,35% de la varianza, mientras que la segunda solo un 6,75%.

```r
par(mfrow = c(1,2))

plot(PVE, type = "o",
     ylab = "PVE",
     xlab = "Componente principal",
     col = "blue")
plot(cumsum(PVE), type = "o",
     ylab = "PVE acumulada",
     xlab = "Componente principal",
     col = "brown3")
```




De manera conjunta, los primeros 7 componentes principales explican en torno al 40% de la varianza de los datos, lo cual no es una cantidad muy alta.

# BIBLIOGRAFÍA

*An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics)*

A tutorial on Principal Components Analysis, Lindsay I Smith February 2002

Abdi, Hervé, and Lynne J. Williams. 2010. "Principal Component Analysis." *John Wiley and Sons, Inc. WIREs Comp Stat* 2: 433–59. http://staff.ustc.edu.cn/~zwp/teach/MVA/abdi-awPCA2010.pdf.