*Clase 11. Preprocesamiento de datos*

El preprocesamiento como concepto

Como dijimos anteriormente, hay una frase conocida en el mundo de la manipulación de datos: **“si entra basura, sale basura”**, en inglés conocido como “Garbage In, Garbage Out”, o sus iniciales GIGO. Los datos de entrada para un proceso de Data Science **no están, en general, preparados para ser analizados o para ser entrada o input de algoritmos**. Por esta razón, existe una serie de actividades que ayudan a “pulir” los datos de entrada, y prepararlos para que sirvan para el proceso de Data Science. A este conjunto de actividades lo denominamos **preprocesamiento de datos**. En esta clase veremos una introducción a algunas de las técnicas estadísticas de preprocesamiento de datos. Con estas técnicas atacaremos problemas comunes, y podremos **convertir los datos de entrada en información útil** para poder realizar un análisis preliminar y **preparar un input adecuado** para los algoritmos de Data Science.

Detección y tratamiento de outliers

## Outliers, repasando el concepto

Como introdujimos en la clase anterior, si consideramos una variable cuantitativa, es esperable que los datos tengan una cierta distribución. Es posible que los datos se acumulen en torno a un valor promedio, o bien que tengan algún grado de dispersión. No obstante, **hay casos donde algunos valores están “demasiado” alejados de la masa de datos central**. Esto puede ser indicador de alguna **situación anómala**, o bien puede ser simplemente **un dato mal cargado**. En todos los casos **es conveniente identificar estos datos extremos**, a los que denominamos **valores anómalos u outliers**, y **tomar acciones en consecuencia** a partir de su detección.

## Outliers en una variable

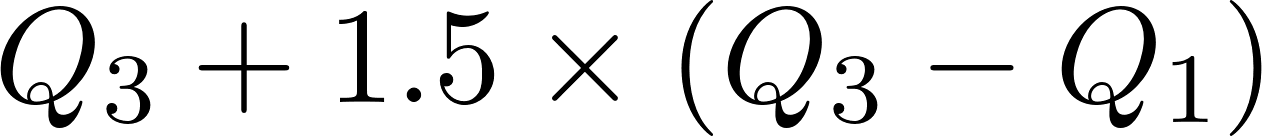
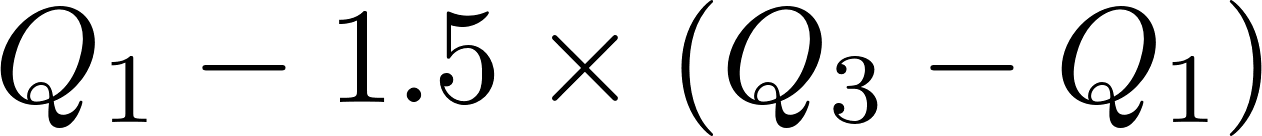
Tal como comenzamos a ver en la clase anterior, si existen outliers en una variable, **podemos verlos en el diagrama de caja y bigotes**. Recordemos los elementos de este gráfico a continuación



Recordemos aquí el significado de los bigotes. Todos los datos que están “más cerca” de la masa central de los datos son considerados inliers. Por su lado, aquellos que están “más alejados” de la masa central son considerados outliers. **Los bigotes marcan los valores de los inliers más extremos**.

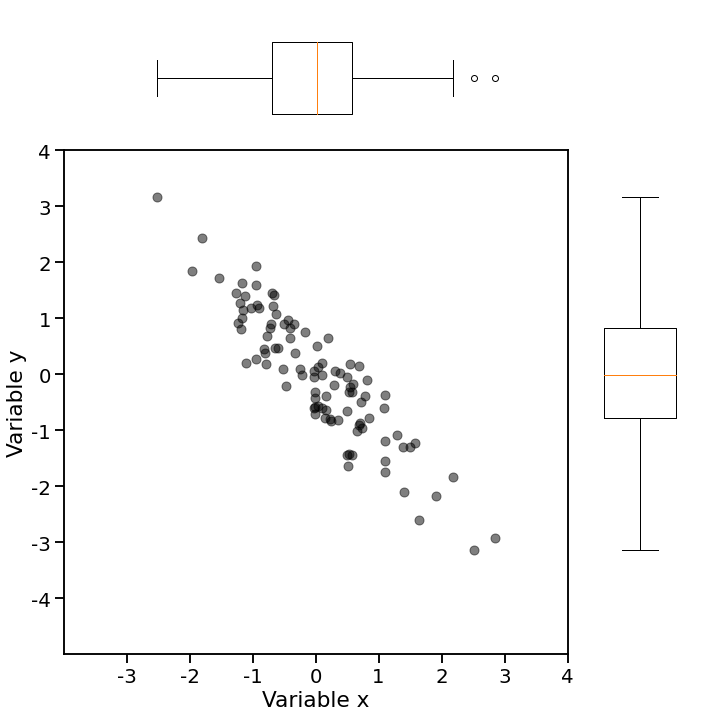
**¿Cómo se establece esta separación entre inliers y outliers?** Habíamos dicho que se establecía una barrera que marcaba el límite entre inliers y outliers, y que no aparece en el diagrama. Esta barrera se calcula de la siguiente forma: **se toma el límite de la caja, y se le añade la longitud de una caja y media. Esto se hace tanto para el límite superior de la caja como para el inferior**. Mostramos un esquema de esta operación a continuación.



La **longitud de la caja** es la diferencia entre los cuartiles 1 y 3 (llamados comúnmente Q1 y Q3), por eso se la denomina **rango intercuartil**. En fórmulas, hablamos de [](https://www.codecogs.com/eqnedit.php?latex=%20Q_3%20%2B%201.5%20%5Ctimes%20(Q_3%20-%20Q_1)%20#0) para el límite superior y [](https://www.codecogs.com/eqnedit.php?latex=%20Q_1%20-%201.5%20%5Ctimes%20(Q_3%20-%20Q_1)%20#0) para el límite inferior de la caja. **Todos los valores por fuera de esos límites son considerados outliers**. Este criterio se utiliza en la mayoría de los lenguajes de programación y herramientas de software estadístico.

## Outliers en dos dimensiones

En un primer acercamiento a la idea, puede parecer relativamente simple encontrar outliers en dos dimensiones: simplemente podrían graficarse diagramas de caja y bigotes para cada una de las dos variables. Esto puede funcionar adecuadamente la mayoría de las veces, pero existen distribuciones de variables que desafían este análisis. Veamos un pequeño ejemplo.



Para este ejemplo, un diagrama de caja y bigotes por cada variable no alcanza para analizar los outliers. Por otra parte, un diagrama de dispersión es de más ayuda, porque muestra los datos relacionados de acuerdo a las dos variables. Más aún, si añadimos un diagrama de caja y bigotes para cada variable dentro del gráfico de dispersión, podemos llegar a este análisis más detallado mostrando tanto relaciones como distribuciones individuales de cada variable. El código para este ejemplo en Python está disponible [en este enlace](https://colab.research.google.com/drive/1Mwq6rJ-E-yImO7Yw7mg7NonQy4e-Pi4N?usp=sharing).

Si miramos solamente los valores de la variable y, con su diagrama de caja y bigotes vertical, vemos que este diagrama **no tiene outliers**. Por su parte, si vemos solamente los valores de la variable x, notamos que su diagrama de caja y bigotes horizontal **tiene dos outliers en su extremo derecho**, que corresponden a los dos puntos del gráfico ubicados en la parte inferior derecha. En este caso, **sería conveniente analizar estos dos puntos en sus dos variables x e y para ver qué relación guardan con el resto de los datos**, y de esta forma determinar si realmente son extremos para el análisis o no. Veamos esta idea a continuación.

## Detectar e interpretar los outliers

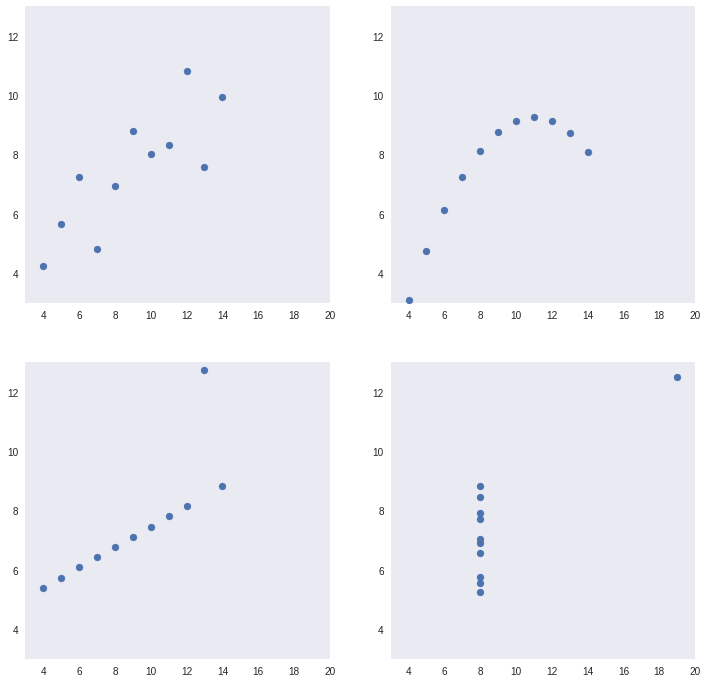
Muchas veces el trabajo con outliers consiste en realidad en **dos tareas bien diferenciadas**: la primera es **identificar y aislar** los puntos que tienen cualidades de outliers propiamente dichos, y la segunda es **tratar de interpretar qué representan esos outliers** para el resto de los datos. Esta clase trata fundamentalmente de la primera tarea a través de herramientas estadísticas y gráficas. No obstante, diremos algunas palabras sobre la interpretación de los outliers.

Esta última tarea es a menudo **más subjetiva** y requiere la participación de **un experto** que conozca el dominio de los datos analizados. Un ejemplo típico de la interpretación de los outliers es analizar los tiempos de llegada de los colectivos del transporte urbano en una parada determinada. Supongamos que tenemos un tiempo de llegada muy alto, o sea que hubo una demora muy grande. ¿Es este dato relevante para nuestro análisis? Quizá haya sido un caso excepcional donde se rompió el coche colectivo y generó una demora; o bien sea un horario donde, por tráfico o intensa afluencia de pasajeros, se generan demoras habituales; o bien es simplemente un dato mal cargado porque quizá el colectivo cambió de recorrido por una eventualidad no programada, y el tiempo de espera para esa parada se demoró hasta que el colectivo retomó su recorrido habitual. En cualquiera de los casos, el valor extremo de tiempo de llegada debe ser analizado, para ver si se considera, se modifica manualmente o se elimina. **En cualquiera de los casos, los outliers deben ser reportados y analizados con un experto del dominio de los datos**.

## Importancia de graficar los datos

Con lo anteriormente mencionado, no podemos dejar de remarcar la **importancia de graficar los datos**, de la manera que sea posible de acuerdo a las características del caso en particular, para poder **analizar sus relaciones y verificar si existen outliers o situaciones particulares**. Para esto, tengamos en cuenta lo aprendido hasta el momento, en particular las clases 5, 6 y 7.

Veamos un ejemplo singular. Miremos por un segundo los cuatro diagramas de dispersión que se muestran a continuación, graficados para cuatro conjuntos de datos denominados I, II, III y IV:



Es claro que los cuatro casos muestran conjuntos de datos sustancialmente diferentes. No obstante, observemos las salidas de Python para los cuatro datasets o conjuntos de datos:

| Coeficiente de correlación para:  Dataset I: 0.81642051634484 Dataset II: 0.8162365060002428 Dataset III: 0.8162867394895984 Dataset IV: 0.8165214368885028   (Promedio de los valores de x, promedio de los valores de y) para:  Dataset I: (9.0, 7.5) Dataset II: (9.0, 7.5) Dataset III: (9.0, 7.5) Dataset IV: (9.0, 7.5)   (Varianza de los valores de x, varianza de los valores de y) para:  Dataset I: (11.0, 4.13) Dataset II: (11.0, 4.13) Dataset III: (11.0, 4.12) Dataset IV: (11.0, 4.12) |
| --- |

Por extraño que parezca, los cuatro conjuntos de datos tienen casi el mismo coeficiente de correlación (con diferencias muy pequeñas), el mismo promedio para x, el mismo promedio para y, y análogamente con las varianzas de ambas variables. Si solamente hubiéramos analizado estos parámetros, no hubiéramos visto diferencias. Por suerte, añadimos los gráficos de dispersión al análisis, lo cual terminó de darnos la perspectiva que necesitábamos para comprender las diferencias entre los cuatro conjuntos de datos.

Este ejemplo, denominado “el cuarteto de Anscombe”, es conocido entre los profesionales de estadística, y está generado artificialmente para mostrar un caso extremo. No obstante, **no estamos exentos de encontrarnos con casos similares**, con lo que debemos estar **atentos a los indicadores estadísticos utilizados, las distribuciones de los datos y los gráficos correspondientes, todo en conjunto**, para poder entender de la mejor manera posible qué es lo que está pasando con los datos que son objeto de nuestro análisis.

Datos ausentes

## Datos ausentes y el mundo real

Cuando trabajamos con conjuntos de datos “de juguete” o “de laboratorio”, en general tenemos muchos detalles resueltos. **La tranquilidad de comenzar el proceso de análisis desde la parte estadística pura**, tomando directamente indicadores como media o mediana, y realizando las distribuciones correspondientes, **es un poco diferente a lo que puede llegar a pasar en el mundo real**. Veamos algunos ejemplos:

* Cuando pedimos responder una encuesta o formulario, **puede que alguna pregunta quede sin responder**. Una buena pregunta a hacernos aquí es ¿descartamos la encuesta porque le falta una respuesta, o tratamos de solucionar ese inconveniente para poder utilizar la información provista por el resto de las herramientas?
* Cuando recolectamos datos de manera automatizada o semi-automatizada, **puede ser que los datos estén mal cargados**, o no se les dio importancia a la hora de cargarlos. Los instrumentos electrónicos de uso no crítico pueden fallar, por ejemplo los equipos de GPS para rastrear los recorridos de los colectivos. ¿Qué hacemos si faltan datos? ¿Descartamos el día o tratamos de estimar los datos faltantes?
* **Puede ocurrir que se nos escape una tecla e ingresemos un número mal**, por ejemplo en una cuenta corriente de un cliente, y el sistema no lo chequea. Pero no nos damos cuenta del error, sino que seguimos operando en el sistema hasta que queremos medir estadísticas y preprocesar los datos. Recién ahí nos damos cuenta del error. ¿Qué hacemos? Sabemos que el dato está mal, pero en general no podemos simplemente borrar el cliente, porque tiene muchos datos relacionados. ¿Y entonces...?

Lo expuesto anteriormente habla de **situaciones donde se presentan datos ausentes**. Los datos ausentes **son parte de los datos del mundo real**. Son tan tangibles y habituales que los lenguajes de programación de Data Science como R y Python definen tipos especiales de datos para estas situaciones.

## ¿Qué hacemos con los datos ausentes?

Un problema con los datos ausentes es que **no todos los métodos y algoritmos brindan la posibilidad de trabajar con ellos**. Entonces tenemos dos grandes opciones: **descartarlos o imputarlos**. Afortunadamente, con Python tenemos gran parte del problema solucionado, porque la librería NumPy brinda funciones para trabajar específicamente con estas situaciones. Recordemos aquí la sección “Datos ausentes” de la Clase 4. En esta sección hablamos de los valores ausentes en Python, denominados NaN, y mencionamos los métodos que tiene Python para trabajar con datos ausentes:

* Las **operaciones vectorizadas ofrecen funciones que descartan los valores NaN**, tales como nansum, nanprod, nanmean, etc. Estas funciones se denominan nanfunctions. Es importante tener en cuenta que las nanfunctions ignoran los valores ausentes explícitamente, por lo que quitan los valores ausentes del conjunto de datos considerado.
* Si efectivamente queremos **quitar la observación completa (la fila de datos completa)** en el caso de que tenga al menos un dato ausente (por ejemplo, si no se contestó a una pregunta, entonces se descarta el formulario completo), podemos usar la función dropna(), que devuelve solamente aquellas filas de datos que no tienen ningún valor NaN.
* Finalmente, **si podemos comprobar que los valores faltantes no serán extremos**, podemos realizar una **imputación de datos**, esto es, **asignar un valor determinado al dato ausente**. Existen distintos niveles de complejidad para realizar imputaciones de datos, pero las más simples y todavía efectivas son las imputaciones por variable: una opción es completar los valores ausentes de una variable por la media de la distribución de la propia variable, y la otra opción es hacer lo mismo pero con la mediana de la distribución. Recomendamos la alternativa de la mediana en el caso en que la distribución de los datos existentes sea demasiado dispersa, por ser más **robusta o “resistente” a los valores extremos**. Recordar que la imputación propuesta se hace de manera separada por cada variable. En Python podemos usar la función fillna() para realizar la imputación de datos. Con fillna() se encontrarán los valores NaN y se reemplazarán por el argumento que se ingresa entre los paréntesis de la función.
* Otra forma de realizar la imputación, especialmente cuando hay pocos datos ausentes y la distribución es medianamente homogénea, o bien sigue una tendencia clara en una serie de tiempo, es la **interpolación por el promedio**. En este caso se toman los dos valores contiguos al dato ausente, y se le imputa a este dato ausente el promedio de dichos valores. La función interpolate(method = ‘linear’) de Pandas ayuda con este propósito. Este método de interpolación “lineal” precisamente recorrerá cada variable y realizará la imputación por el promedio de los valores contiguos.

Finalmente, recordemos que **los datos ausentes son siempre importantes porque pueden introducir irregularidades que distorsionan los resultados del análisis**. La recomendación es **trabajar fuertemente con un experto del dominio**, y verificar constantemente **que la eliminación de observaciones no traiga consecuencias negativas, y que la imputación arroje valores razonables**. Al igual que los outliers, los datos ausentes **deben ser siempre reportados y analizados** para entender su origen y tratar de minimizarlos.

Análisis de Componentes Principales.

## Un aliado para el preprocesamiento

En preprocesamiento de datos, es importante entender qué es lo que sucede cuando hay muchas variables interrelacionadas. Muchas veces serán los algoritmos de Data Science los que nos den la respuesta. Pero **no por esto podemos confiar en que solamente ingresando los datos ya sea suficiente para obtener resultados útiles**. Recordemos la idea de GIGO (“basura entra, basura sale”) presentado al comienzo de esta clase.

Ahora bien, supongamos que ya “limpiamos la basura”. ¿Es esto suficiente? Podría ser, pero sería conveniente hacer una revisión extra. **¿Qué tal si pudiéramos simplificar la información que tenemos por delante?** Veamos cómo hacer esto. Prestemos atención a la siguiente foto:



Podemos ver que **claramente tenemos dos personas**, probablemente adultas, tomándose de la mano. Podríamos inferir que la persona que está a la izquierda de la foto podría ser más alta que la de la derecha, pensando que están paradas sobre un mismo plano imaginario. Pensamos por un momento: ¿qué tal si pudiéramos hacer algo parecido con los datos en múltiples variables? Más aún, **¿qué tal si pudiéramos “ver la sombra” de los datos de tal forma de poder acomodarlos en una “foto” de dos dimensiones, y así poder verlos mejor?**

El análisis de componentes principales es una herramienta muy específica en el campo del álgebra multivariable. Si bien no vamos a ver su fundamento matemático aquí, lo importante de las componentes principales es su gran utilidad de aplicación para el preprocesamiento de datos.

Acercándonos al concepto

Veamos el [siguiente video](https://www.youtube.com/watch?v=wKSrdUNEBUk), donde podemos ver al reconocido inventor y pionero Nikola Tesla. Pero **OJO! solamente podemos verlo bien si nos colocamos en un punto específico**. Las componentes principales funcionan de una manera análoga: **disponen a los puntos en el lugar donde mejor podemos ver su “sombra”.**

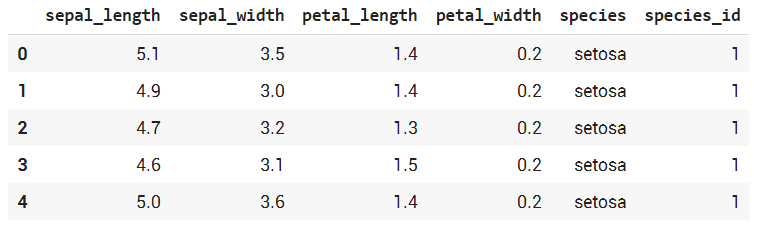
Consideremos el siguiente conjunto de datos:

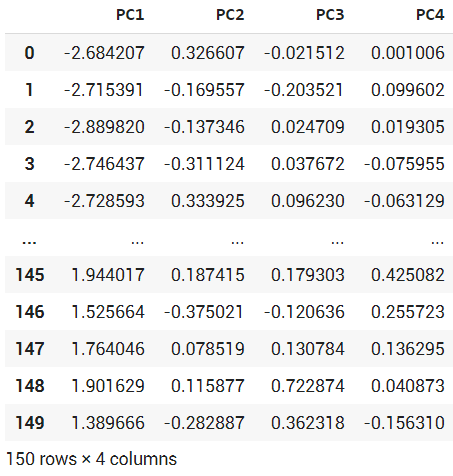


Este es el conocido conjunto de datos Iris, con 150 observaciones de 3 tipos de flores diferentes: setosa, versicolor y virginica. Existen 50 observaciones por cada tipo de flor, medidas en cuatro variables: largo y ancho del sépalo, y largo y ancho del pétalo. En este gráfico podemos observar diagramas de dispersión entre todos los pares de variables. Cada tipo de flor se muestra con un color diferente. Podemos intuir que las flores de tipo setosa, marcadas en color azul, están mayormente separadas del resto. Esto significa que los valores de sus variables parecen ser en general diferentes de las demás flores.

Ahora bien, ¿cómo podríamos reducir tanta información? Aplicar componentes principales es como **encender una fuente de luz, que busca la sombra mejor proyectada por los datos**, con una pérdida de información que intenta ser lo más pequeña posible.

Esta sombra estará proyectada en la misma cantidad de variables que las originales. Esto significa que al aplicar componentes principales para una cantidad determinada de variables, obtendremos la misma cantidad de componentes. **Las componentes son nuevas variables, cada una de las cuales surge de realizar un cálculo con todas las variables originales**. En los gráficos siguientes pueden verse primero las variables originales, y luego las componentes principales





Con respecto a lo dicho anteriormente, para este caso tenemos 4 variables: sepal\_length, sepal\_width, petal\_length y petal\_width. Si aplicamos componentes principales obtendremos 4 componentes, a las que llamaremos PC1, PC2, PC3 y PC4. Cada elemento de PC1 es el resultado de una operación entre las 4 variables originales. Asimismo con PC2, PC3 y PC4.

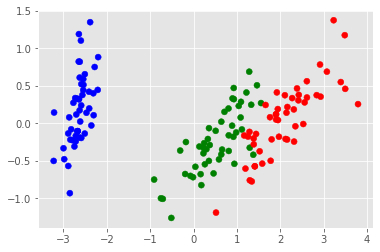
¿Qué acabamos de hacer? Simplemente **creamos un nuevo conjunto de datos** (las componentes principales) a partir del conjunto de datos. Los elementos para este nuevo conjunto de datos son los mismos 150 elementos del conjunto original. **Lo que cambia son las variables que los describen**. Estas nuevas variables están calculadas de tal forma que **cada una de ellas puede explicar la variabilidad de los datos** (su medida de dispersión o su “sombra”) **con distinto grado de importancia**. La PC1 explicará el mayor porcentaje de los datos, la PC2 un poco menos, y así en orden decreciente. De esta forma, **las componentes principales están ordenadas por su grado de importancia.**

¿Cómo se mide esta importancia? Disponemos de un indicador específico para esto, el **“explained variance ratio” o tasa de variabilidad explicada**, que mostramos en la siguiente tabla:

| PC1 | PC2 | PC3 | PC4 |
| --- | --- | --- | --- |
| 92.46 | 5.30 | 1.71 | 0.51 |

De esta forma, la PC1 explica el 92% de los datos, mientras que la PC2 explica el 5.30% y así sucesivamente. Si lo pensamos de esta manera, quiere decir que la PC1 y la PC2 juntas explican el 97.76% de los datos. Siguiendo con esta idea, estaríamos perdiendo solamente un 2.24% de información. Python provee todas estas funciones en el paquete scikit-learn, que veremos más adelante.

Si tenemos dos nuevas variables que explican más del 97% de los datos originales, **es una selección de variables muy conveniente para graficar en dos dimensiones**. Veamos el gráfico:



Es claro en el gráfico que existe un grupo de datos visiblemente separado del resto, que corresponde a las flores del tipo setosa, tal como veíamos anteriormente. De esta forma **tenemos información resumida de un conjunto de datos de 4 variables originales en 2 componentes principales que muestran claramente la separación de los datos en dos subconjuntos bien definidos**.

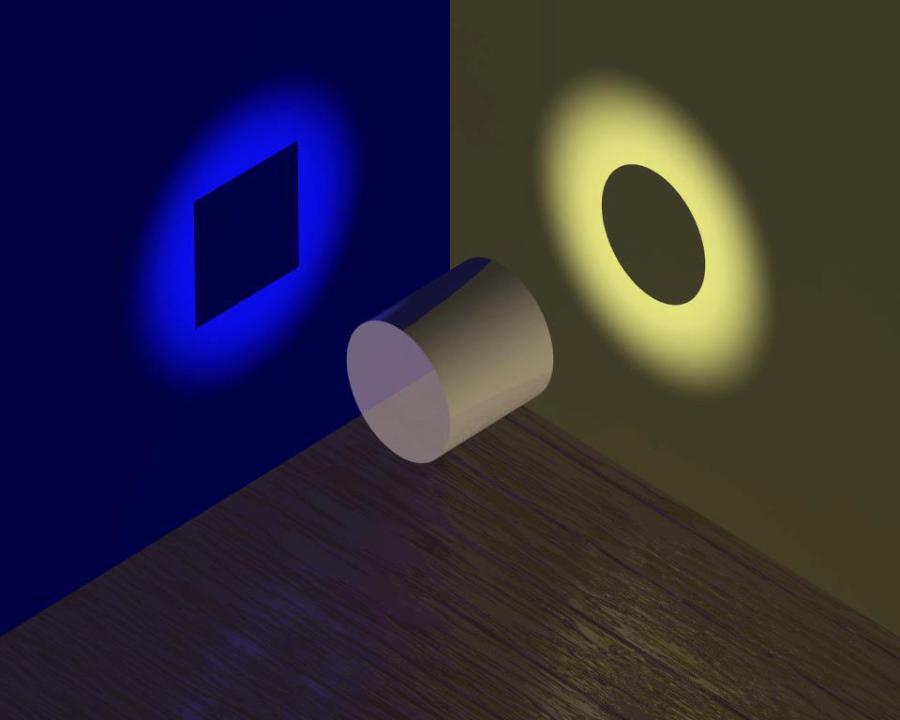
## Comentarios finales

Es necesario tener en cuenta dos cuestiones al usar componentes principales. La primera cuestión es fundamental: **las componentes principales no tienen significado**. No tienen unidad de medida, ni señalan ninguna característica de los datos. **Es un error tratar de darles una interpretación.** En ciertas bibliografías se induce a pensar que existe algún tipo de relación entre las componentes principales y las variables originales, pensando en encontrar alguna otra forma de caracterizar los datos. Aquí **no recomendamos realizar esa interpretación, sino más bien pensar que son otra forma de visualizar los datos**.

Entonces, ¿para qué sirven? Fundamentalmente para encontrar algunos detalles en el conjunto de datos que pueden pasar desapercibidos cuando estamos analizando muchas variables en conjunto. Señalamos dos de las situaciones más comunes:

* **Si un dato es outlier en las componentes principales, también será outlier en el conjunto de datos original**. Esta es una excelente herramienta para encontrar valores demasiado alejados de la masa central de datos.
* **Si un conjunto de datos conforma un grupo separado y bien diferenciado en las componentes principales** (como las flores de tipo setosa de nuestro ejemplo), entonces **también conformará un grupo diferenciado en las variables originales.** A partir de esta propiedad también las componentes principales se utilizan muy a menudo para aplicar algoritmos de agrupamiento o clustering.

La segunda cuestión es **utilizar las componentes principales con precaución.** Veamos el siguiente gráfico:



De acuerdo a dónde pongamos la fuente de luz tendremos resultados diferentes e igualmente válidos. **Si los datos no tienen una forma muy definida, nos encontraremos en una situación similar para las componentes principales. Por lo tanto no serán representativas de los datos, o sea que no servirán para graficarlos en dos dimensiones.** Para darnos cuenta de esta situación basta con sumar la variabilidad explicada por las primeras dos componentes: si su suma no alcanza un valor demasiado grande (podemos pensar en 80% o 90%) posiblemente nos encontremos en esta situación y la visualización de los datos no arrojará resultados útiles. En el caso extremo, por ejemplo puntos que en 3 dimensiones forman una esfera, donde si apuntamos una fuente de luz vemos lo mismo desde todas las direcciones, un análisis de componentes principales arrojará una tasa de explicación de variabilidad idéntica para cada componente. Esto es, si hay 3 variables habrá entonces 3 componentes, y cada una de ellas explicará el 33,33% de los datos.

FIN