

# Introducción a la Ciencia de Datos

## Introducción a la Ciencia de Datos

La primera pregunta que cabe hacernos es: ¿Qué es la Ciencia de Datos?

Resulta claro que, si estamos participando de este curso, todos tenemos alguna forma de respuesta.

Cómo todo concepto que está de moda viene siendo bastante usado y abusado. Por ese motivo es que, siento, nos conviene precisar el concepto.

Empecemos explorando las cosas que habitualmente se consideran relacionadas:

- Datamining
- Datawarehouse
- Estadísticas
- Bases de datos
- Big Data
- Machine learning
- Knowledge discovery

No nos caben dudas que estas cosas están relacionadas con los contenidos que nos interesa explorar. ¿Es eso todo? Seguro que la respuesta es NO.

No sólo porque siempre hay para agregar un sinfín de material técnico. Sino porque también es preciso considerar disciplinas más "blandas" que son necesarias para la marcha de los negocios y que tienen que ver con realizar una comunicación eficaz, con la comprensión de los distintos procesos internos de las organizaciones, con las habilidades conceptuales y los planteos de carácter estratégico.

Vamos a considerar entonces dentro de la "Ciencia de datos" a todas las herramientas necesarias para realizar los negocios de manera inteligente.

- Alcanzando los mejores resultados posibles.
- Tomando las decisiones con criterios racionales.

¿En que se distingue esto de la práctica, varias veces milenaria, de los negocios?

Desde hace mucho tiempo los seres humanos venimos ejerciendo el comercio. Como en el resto de nuestras vidas no necesariamente todo lo que hacemos se guía por decisiones racionales. Muchas veces y con gran éxito los negocios se conducen "por instinto"

La explicación del instinto comercial queda fuera de los alcances del curso. Existe y ha llevado a varios casos a éxitos resonantes. Menos resonantes han resultado los fracasos de un gran número de emprendedores que trataron de aplicar herramientas intuitivas y se lanzaron en aventuras sin validación racional.

La diferente representación delante de nuestra imaginación de estos múltiples fracasos ignotos y lo muy presentes que tenemos éxitos resonantes (Microsoft, Facebook, etc) nos hacen a veces dudar del criterio racional para los negocios.

El carácter excepcional de las excepciones es lo que debe confirmarnos en la validez de seguir criterios racionales.

Niveles en los que opera la inteligencia de negocios

De acuerdo a Katz (Buscar cita Harvard Business Review) existen dentro de las organizaciones tres niveles:

- Nivel técnico
- Nivel de mandos medios
- Nivel de dirección

La inteligencia de negocios opera al servicio de los tres niveles.

Es un gran desafío para la gente del nivel técnico entender lo que se cocina en el nivel de dirección. Pueden sin gran dificultad asomarse a los problemas y el trabajo de los mandos medios pero les resultan prácticamente incomprensibles los correspondientes al nivel de dirección.

Siguiendo a Katz vamos a tratar de explicarlo a partir de las habilidades fundamentales de cada nivel:

Nivel	Habilidades Principales	Habilidades Secundarias
<b>Técnico</b>	Técnicas	Interpersonales
<b>Mando Medio</b>	Interpersonales	Técnicas
<b>Alta Dirección</b>	Conceptuales	Interpersonales

Para todos nosotros es muy fácil imaginarnos que son las habilidades técnicas. Escribir un programa, hacer una consulta eficiente, tornear una pieza, etc.

Aún estando plenamente en el nivel técnico tenemos una clara imagen de lo que son las habilidades interpersonales, aunque más no sea por cuando las echamos en falta en nuestros jefes. Liderar, persuadir, inspirar son habilidades interpersonales que se vuelven claves en el nivel de los mandos medios.

No ocurre lo mismo con las habilidades conceptuales que son la clave de la Alta Dirección. Estas habilidades consisten en ver los procesos de creación de valor y en manejarlos adecuadamente para maximizar los beneficios actuales y futuros.

Las habilidades conceptuales tienen en cuenta los ciclos de duración de los negocios, que pueden involucrar varios años, y manejan las incertidumbres que corresponde a ese tipo de horizonte temporal.

La inteligencia de negocios debe operar a los tres niveles:

- En el nivel técnico se construyen ETL, se realizan análisis automatizados.
- El nivel de mando medio se especializa en traducir los problemas de negocios a análisis que puedan ejecutarse y, a la vuelta, en la interpretación y comunicación de los resultados hallados.
- El nivel conceptual trabaja la estrategia de la compañía. Se encarga de plantear las pruebas necesarias para conocer con mínima inversión si un nuevo negocio es o no

**rentable.** También debe aportar a la predicción del tamaño que ese negocio podrá alcanzar en el mercado y, por lo tanto, del horizonte temporal. Las funciones de BI se vuelven fuertemente entrelazadas con las de la gente de Marketing y/o Comercial. Hasta que la organización no supera un dado tamaño mínimo no es posible tenerlas diferenciadas a todo nivel. **El nivel que debe predominar en cada organización depende del tipo de negocio.**

Tradicionalmente si los clientes y proveedores son pocos las negociaciones son 1 a 1 y la Dirección Comercial tiende a prevalecer (operaciones del tipo B2B). Si los clientes son muchos (B2C) se espera un predominio de la dirección de Marketing.

BI siempre tiene un lugar en el juego, pero, sin embargo, su papel se ve reforzado cuando existe o puede generarse mucha información previa al contacto con los clientes de manera de optimizar el proceso de venta en las operaciones del tipo B2C.

En este punto necesitamos hacer un esfuerzo extra para lograr asomarnos al nivel conceptual. **La pregunta básica es: ¿Cómo gana dinero mi empresa? Las respuestas pueden ser muy variadas:**

- Comprando barato y vendiendo más caro (empresa de tipo comercial)
- Comprando materias primas, transformándolas y vendiéndolas (empresa de tipo industrial)
- Contratando personal y sistemas para que brinden servicios. (empresa de servicios)
- Cobrando primero, pagando después y aprovechando la renta del dinero de otros (empresa de tipo financiero)

Resulta clave, entonces, saber cómo es el proceso de generación de valor de la propia organización. Esto nos dará pautas sobre donde conviene concentrar los esfuerzos analíticos.

### Ejercicio 1.1.1:

Para la organización para la cual trabaja:

- Describa el proceso de generación de valor que utiliza.
- Asigne un valor económico a cada uno de los factores de costo para una unidad de venta
- Para cada valor económico asignado indique un margen de variación razonable
- Encuentre el factor cuya variabilidad tiene mayor influencia en el resultado final.

Armados de información del tipo del ejercicio 1.1.1 podemos saber donde conviene concentrar los esfuerzos analíticos. En el nivel conceptual no se trata ya de hacer cuentas y, ni siquiera, de interpretar los resultados, sino de saber dónde poner el foco y como plasmar en estrategias las conclusiones.

### Introducción a Datawarehousing

Muchas compañías han ido acumulando la información de sus sistemas transaccionales a lo largo del tiempo. Dentro de esa información están presentes los hábitos de compra, las estrategias de fraude, las probabilidades de incobrabilidad, y muchas más circunstancias que son muy importantes para la operación de la empresa.

Los criterios con los que se almacenaron los datos en las aplicaciones transaccionales montadas sobre bases de datos relacionales han seguido, en el mejor de los casos, las buenas prácticas condensadas en las formas normales.

Estas buenas prácticas ponían foco en asegurar la consistencia de la información y en el ahorro de espacio de almacenamiento que era, años atrás, prohibitivamente caros.

Además se perseguía que las operaciones cotidianas fueran ágiles sin considerar. Las operaciones masivas de extracción de información podían poner en peligro las operaciones cotidianas. Se imponía entonces, previo a la ejecución concertada de cualquier esfuerzo analítico mudar la información a un reservorio independiente.

Otro problema muy común pasa porque los operadores de los sistemas transaccionales tienen su foco puesto en que el día a día del negocio discurra sin mayores interrupciones y no sienten que reflejar adecuadamente la realidad en los sistemas sea una parte importante de sus trabajos. No llenan la información como si después fuera a ser usada para algo más allá del negocio entre manos. Esta actitud genera dos defectos que pueden resultar muy caros desde la perspectiva del analista como ser los faltantes de información y las estructuras horizontales.

Muchas veces un operador deja de completar o completa parcialmente la información requerida. Los sistemas diseñados con foco en la agilidad tampoco previenen esta práctica y los datos se pierden para siempre.

También resulta decepcionantemente común que los operadores alteren el modo de usar un sistema guardando distinto tipo de información en los campos. Son fenómenos del tipo: "Ah, antes del 2000 guardábamos el código postal en el campo observaciones", o bien, "No, la localidad recién está normalizada desde el 2003, antes era un campo libre y, para los clientes ya conocidos se dejaba en blanco"

Muchos esfuerzos de análisis se estrellaron contra estos problemas. La lección duramente aprendida es que antes de aplicar las herramientas analíticas es preciso preparar los datos. Además de la preparación conviene almacenarlos en una plataforma independiente con una estructura diseñada desde el principio para agilizar la tarea analítica.

Una de las primeras cosas que los esquemas de almacenamiento pensados para el análisis incorporan es la desnormalización de los datos. De esa forma es posible construir todos los índices necesarios para que puedan calcularse en forma directa los agregados.

Algunos sistemas van más allá y guardan agregados pre-calculados. Esto les permite evacuar las consultas en forma más rápida pero a costa de una mayor complejidad de actualización.

Los sistemas transaccionales, por otra parte, suelen almacenar la última versión de la información correspondiente. En muchos casos "pisan" datos que pueden ser relevantes. Podría darse el caso de figurar en cero la deuda de un cliente pero no almacenar la información sobre si pagó puntualmente, pagó en forma tardía o se le condonó la deuda por motivos impositivos.

Desde la perspectiva de un DW la historia del cliente es muy relevante.

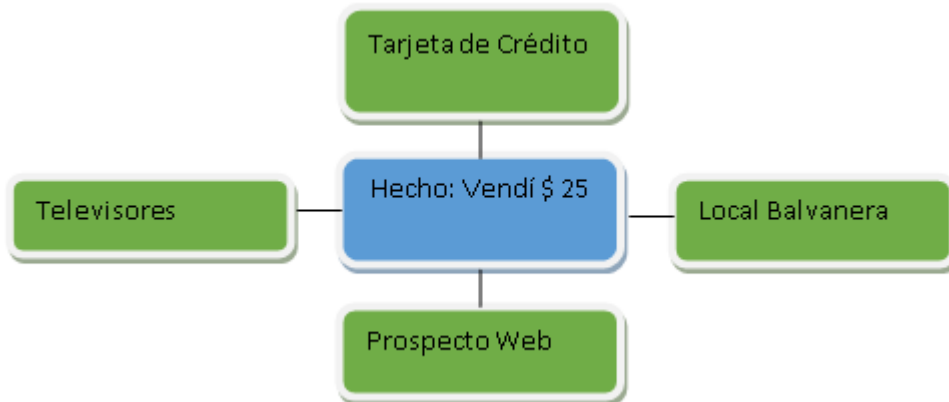
Resumiendo:

Aplicación transaccional (OLTP)	DW
3ra forma normal	2da forma normal para cada dimensión
Modelo de datos complicado y eficiente	Modelo de datos simple y redundante

Se pierden los cambios	No se registran cambios, se agrega información
Optimizado para la operación	Optimizado para la lectura

Diseño de DW:

Estrella:



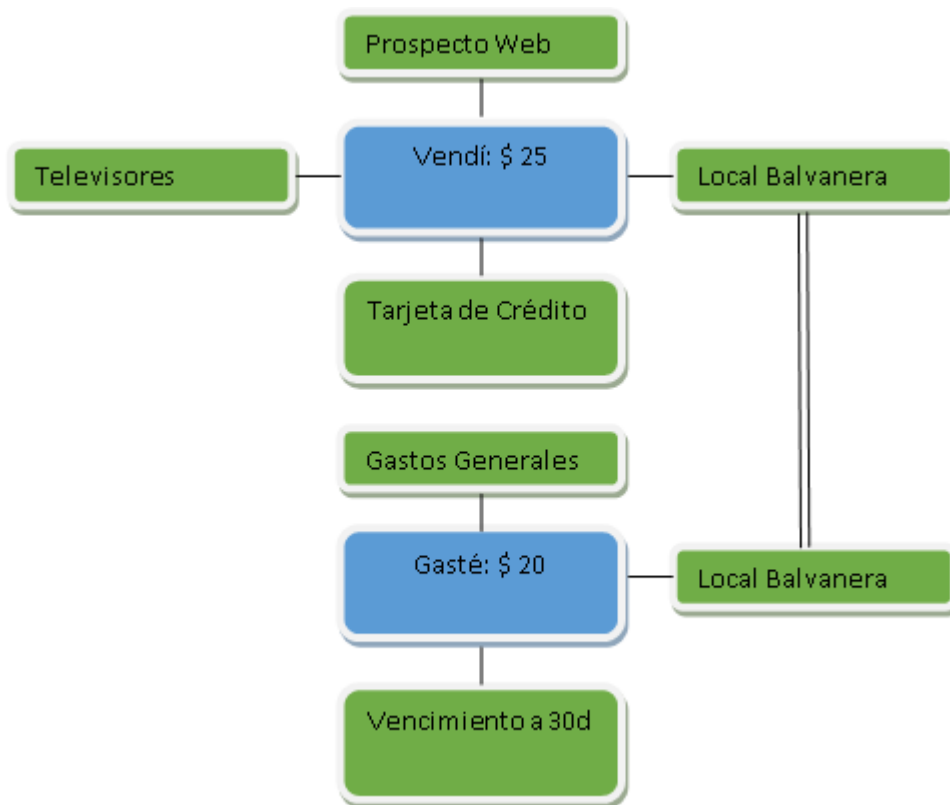
El centro lo ocupa el hecho relevante, por ejemplo el importe vendido. A su alrededor se ponen todos los atributos que tienen que ver con este hecho.

Constelaciones

Pero, es posible, que no alcance con una sola tabla de hechos para representar toda la información relevante de la compañía. Obviamente se puede trabajar con dos tablas de hechos que no compartan ninguna dimensión (atributo)



Sin embargo es mucho más probable que, dentro de la misma organización las temáticas estén relacionadas y las tablas de hechos compartan atributos:



Al atributo compartido se lo denomina, en la jerga, agujero de gusano.

Al combinar varias constelaciones de diferentes compañías se puede realizar lo que se llama una "galaxia"

El diseño de los repositorios de datos constituye el objeto específico del módulo "Técnicas de Almacenamiento y procesamiento de datos" que forma parte del curso más avanzado que se dicta sobre BI en la modalidad virtual.

#### Introducción al Datamining

Hacer Data Mining consiste en explorar, por medios automáticos o semi-automáticos, grandes volúmenes de datos con la finalidad de descubrir reglas y patrones significativos. El objetivo que se propone es brindar información al negocio ayudando a las empresas a mejorar sus operaciones a través de un mayor entendimiento de su contexto y posibilidades:

¿Qué clientes es más probable que acepten una oferta? ¿Qué clientes tienen mayor probabilidad de dejar de pagar? ¿Qué clientes tienen las mayores probabilidades de pedir la baja del servicio? ¿Qué demanda puedo esperar para mis productos para el año próximo?

Estos análisis se basan en los datos pasados y pueden usarse para predecir el futuro. La calidad de los datos y el conocimiento del negocio son la clave de cualquier análisis. Los

datos deben ser entendidos y cuidados como un activo que le permitirá a las organizaciones diferenciarse proporcionando más y mejores servicios.

¿Por qué minería de datos?

Las empresas tienen grandes volúmenes de datos recolectados y almacenados:

- Datos generados por aplicaciones en la nube, redes sociales.
- Compras en negocios.
- Comercio electrónico.
- Transacciones bancarias / Tarjetas de Crédito.
- Sistemas de monitoreo: sensores, web logs, etc.

Muchas veces hay información escondida en los datos que no resulta evidente a los ojos de un analista, o por una simple cuestión de volumen jamás llega a ser analizada.

La desproporción entre el volumen de información y la cantidad de analistas disponibles crece con el tiempo impulsada por una cada vez mayor presión competitiva.

Los datos, por lo tanto, deben ser entendidos como un activo que le facilitará a las organizaciones brindar más y mejores servicios, predecir tendencias futuras, anticiparse a ellas, etc.

Ciclo de vida de los datos:

El datamining empieza por los datos aunque, rara vez, los datos se generan bajo control del equipo de análisis.

En general los datos que van a ser objeto de la minería nacen como subproducto de las acciones de otros sectores que hacen un uso muy distinto. Los sectores relacionados con la operación del negocio tienen objetivos de corto plazo. No les interesa registrar consistentemente a lo largo de años las mismas cosas con los mismos criterios. Pueden resolver muchas cuestiones con agilidad recurriendo al expediente de anotar atributos clave en campos libres del tipo "observaciones"

La minería de este último tipo de datos, si bien no es imposible, es, por este motivo, más compleja y trabajosa.

Entonces, a la dificultad planteada por la modalidad de almacenamiento de los datos en las aplicaciones transaccionales se suman las consecuencias de esta divergencia en el criterio de carga de los mismos. Por todo esto se impone extraerlos, transformarlos y cargarlos en un depósito específico (o Datawarehouse)

El proceso de ETL (extracción, transformación y carga) constituye entonces otro paso en el ciclo de vida de los datos. Las herramientas, técnicas y desafíos de los ETL constituyen el foco del módulo "Técnicas de Limpieza y carga de datos" que forma parte del curso de BI más avanzado que brinda la UTN en la modalidad virtual.

Una vez que los datos se encuentran limpios y cargados en un almacén específico puede comenzar el proceso de análisis. El grueso de este curso se focaliza en las técnicas más conocidas y usadas para llevar a cabo ese análisis.

Realizar análisis con datamining no termina el ciclo de vida de la actividad. Además hay que presentar los resultados a los tomadores de decisiones. Las herramientas de datamining contienen facilidades para la presentación de resultados. Si bien un gráfico



puede valer más que mil palabras, por si mismos, en particular frente a ojos no entrenados en su interpretación pueden, además, necesitar las palabras que lo expliquen. Un buen analista necesitará que sus resultados sean comunicados por sí o por otro. Deben ser volcados a lenguaje de negocios y marcar con énfasis los límites del conocimiento adquirido.

Normalmente se considera inútil tratar de responder la pregunta - ¿Qué queda afuera? - de un dado objeto de estudio ya que el universo de cosas que quedan afuera es, normalmente, infinito. Sin embargo, para facilitar la tarea de trazar una frontera conceptual ayuda indicar las cosas que están afuera, aunque más no sea, las que más podrían prestarse a confusión.

- Datamining NO es un producto de software que puede comprarse.
- No es una solución ni mágica ni instantánea a los múltiples problemas de negocio.
- No es un fin en sí mismo, sino un proceso.
- No es un dogma de fe, es una disciplina con sustento matemático y estadístico.
- Muchas veces para desplegar su pleno alcance requiere interactuar con los procesos operativos de la organización para prevenir la pérdida de información

Las técnicas de datamining pueden clasificarse como:

1) Descriptivas:

Identifican patrones que explican o resumen aspectos de los datos por ejemplo:

- Agrupamiento
- Reglas de asociación
- Correlaciones

2) Predictivas:

Tratan de inferir el comportamiento de una variable objetivo a partir de una multiplicidad de atributos por ejemplo:

- Regresión
- Árboles de decisión
- Redes neuronales

Todas estas herramientas y otras se cubrirán en los módulos correspondientes de este curso.

Uno de los términos que se usa habitualmente como sinónimo de minería de datos es "análisis inteligente de datos" o también "descubrimiento de conocimientos"

En todos los casos se trata de encontrar conocimiento que sea:

- válido: representa la realidad
- novedoso: aporta algo desconocido
- potencialmente útil: la organización podrá usar el conocimiento a favor de sus objetivos

- comprensible: es importante que sean interpretables en términos que tengan sentido para los expertos

Relaciones con otras disciplinas:

Ningún campo de la actividad humana existe en el vacío. Forma parte de un entramado complejo del cual es útil tener conciencia ya que otras disciplinas siempre aportan métodos y herramientas.

En el caso que nos ocupa, la minería de datos usa conceptos y herramientas de:

- Bases de datos
- Carga y limpieza de información
- Toma de decisiones
- Aprendizaje automático
- Probabilidad y estadísticas
- Técnicas de visualización
- Procesamiento en Paralelo

Todo proyecto de minería de datos suele cubrir las siguientes fases:

- Relevamiento de la información disponible y de las necesidades del negocio
- Extracción, limpieza y transformación
- Minería de datos propiamente dicha
- Evaluación e interpretación
- Comunicación al negocio

Introducción al "Knowledge Discovery"

¿Qué es el conocimiento?

Podemos pensarlo como la capacidad para describir con reglas simples fenómenos complejos.

En el campo que nos ocupa los fenómenos complejos se encuentran descriptos en detalle, caso por caso, en grandes conjuntos de datos.

El primer paso para descubrir reglas simples es aplicar los criterios de la estadística descriptiva, promedios, dispersiones, percentiles, etc. Esto nos puede ayudar a sacar conclusiones muy elementales usando sólo la mera observación.

Una vez agotados estos medios y, en especial, frente a problemas donde el número de variables relevantes supera las dos o tres necesitamos ayuda para nuestra capacidad humana de correlacionar cosas.

Armados con técnicas de minería de datos podemos buscar reglas de diverso tipo que nos permitan alcanzar una mejor comprensión del problema entre manos.

Uno de los problemas más comunes es que las herramientas nos lleven a reglas que son, en el fondo, un ajuste casual a la muestra bajo análisis y que no representen en absoluto al universo del que queremos aprender. Esto es llamado "sobre ajuste" (overfitting)

Hay dos remedios para esto. El primero es trabajar con varias muestras, inferir sobre una y luego asegurarse que los resultados se ajustan también a las otras. El segundo es más delicado ya que pasa por exigir que la regla tenga sentido dentro del marco teórico del problema.

Este último tipo de verificación requiere que el analista entienda las dependencias entre las variables y no sea ajeno a la problemática de negocios que está abordando. Esto no implica que no debe tener abierta la mente a la posibilidad de encontrar resultados que contradigan su sentido común. Simplemente, de encontrarlos, deberá ser aún más cuidadoso en su verificación así como persuasivo en su comunicación.

Estos resultados que contradicen el sentido común suelen ser los que agregan más valor al negocio cuando pueden justificarse apropiadamente y venderse al management de manera creíble.

Ejemplo:

La empresa X basa su actividad en la explotación de un conjunto de compradores recurrentes. La situación crediticia de buena parte de los compradores recurrentes es complicada y no siempre consiguen cubrir el importe de las compras que inician.

Para la compañía una compra que no puede facturar por falta de crédito implica mucho trabajo con ningún beneficio.

Por ese motivo se procedió a eliminar de la base de prospectos a todos los compradores recurrentes que nunca hubieran conseguido concretar una operación.

Esto produjo una disminución del nivel de compras no concretadas, pero, una disminución aún mayor del nivel total de compras.

Un análisis más detallado arrojó que:

Probabilidad de concretar dado que concretó siempre = 65%

Probabilidad de concretar dado que no concretó nunca = 35%

Sin embargo el esfuerzo necesario para iniciar una venta también cambiaba:

Esfuerzo necesario para iniciar una venta a los que concretaban siempre = 5 horas

Esfuerzo necesario para iniciar una venta a los que no concretaban nunca = 2 horas

Por lo tanto vendiendo sólo a los clientes con el mejor puntaje de concreción podía esperar una venta cada  $5 / 0.65 = 7.7$  horas mientras que para los peores prospectos serían  $2/0.35 = 5.7$  horas lo que resulta mucho más económico.

El descubrimiento de esta pieza de información que contradecía el sentido común establecido resultó particularmente valiosa ya que permitió expandir el negocio de la empresa recuperando clientes perdidos al tiempo que se aumentaba el margen promedio de ganancia.

### Introducción a las herramientas de OLAP y Tableros de Comando

#### Herramientas OLAP

A partir del ejemplo anterior empezamos a asomarnos a la problemática que significa tener para cada prospecto distintos atributos. En nuestro caso eran:

- Frecuencia con la que concretó sus compras
- Esfuerzo necesario para llegar a una venta

Podemos imaginarnos que en una operación real los clientes podrían ser clasificados por muchos más parámetros:

- País
- Estado
- Ciudad
- Nivel de crédito
- Antigüedad
- Volumen de operaciones
- Industria a la que se dedica

y una infinidad de etcéteras.

A un analista le puede interesar agrupar los clientes con diferentes criterios y ver como se reparten los demás. Si una ciudad concentra muchos de poca antigüedad habla de crecimiento.

Si la misma ciudad concentra muchos de escaso nivel de crédito entonces podría ese crecimiento ser peligroso.

Si el crecimiento en general se concentra en una industria entonces una ciudad muy relacionada con esa industria podría también mostrar un crecimiento importante que no estaría directamente ligado a una acción específica del equipo correspondiente.

Para investigar todas estas combinaciones resulta útil poder mostrar muy rápidamente todas las combinaciones posibles de estos totales de clientes.

Esto se puede, por supuesto, resolver con una consulta a la base de clientes que totalice, cada vez, lo que se le pide. Sin embargo puede resultar mucho más rápido tener todos los subtotales pre - calculados y agruparlos según sea necesario.

En esto consiste una herramienta OLAP.

Cada atributo puede formar parte de una jerarquía:

- 1) País 1
  - a) Estado 1
    - i) Ciudad 1
    - ii) Ciudad 2
  - b) Estado 2
    - i) Ciudad 3
- 2) País 2
  - a) Estado 3
    - i) Ciudad 4
    - ii) Ciudad 5

Cada jerarquía está asociada a un único concepto. (En el caso precedente sería "ubicación")

Otra jerarquía podría ser la antigüedad con Año, Trimestre, Mes

El nivel de crédito, la industria a la que se dedica y el volumen de operaciones no formarían parte de ninguna jerarquía.

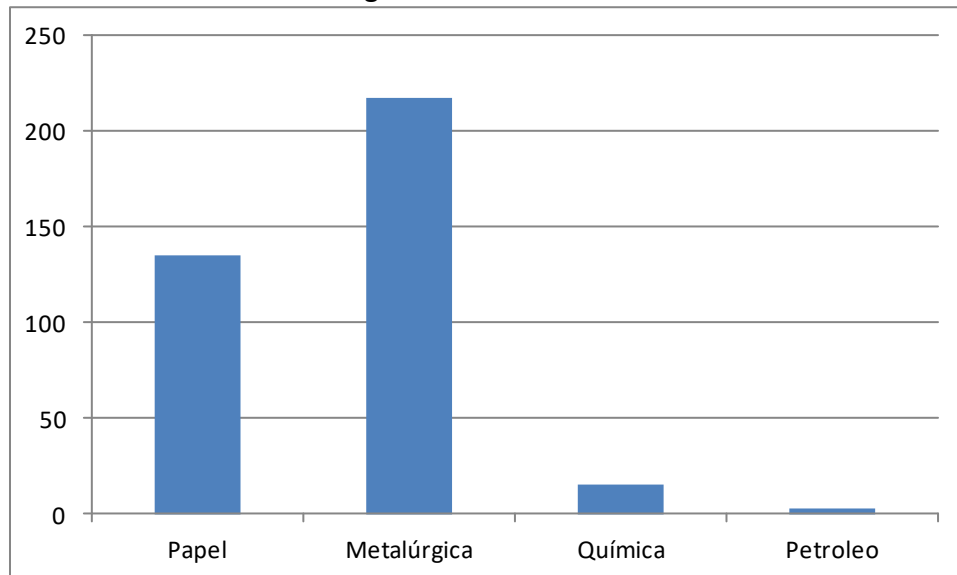
Cada jerarquía y cada atributo no jerarquizado forman lo que se llama una dimensión.

El problema que venimos tratando se representaría en 5 dimensiones:

- Dos corresponden a jerarquías (Ubicación y Antigüedad)

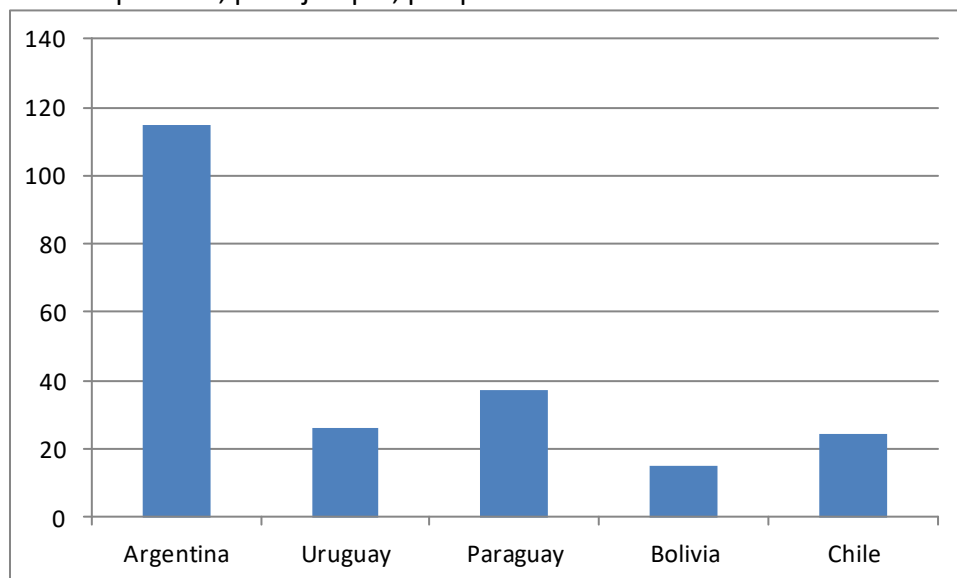
- Tres corresponden a atributos no jerarquizados (Nivel de Crédito, Volumen de operaciones, Industria)

El analista parte de un total de clientes y puede separarlos según cualquier dimensión. Al hacerlo convierte un número en un gráfico:

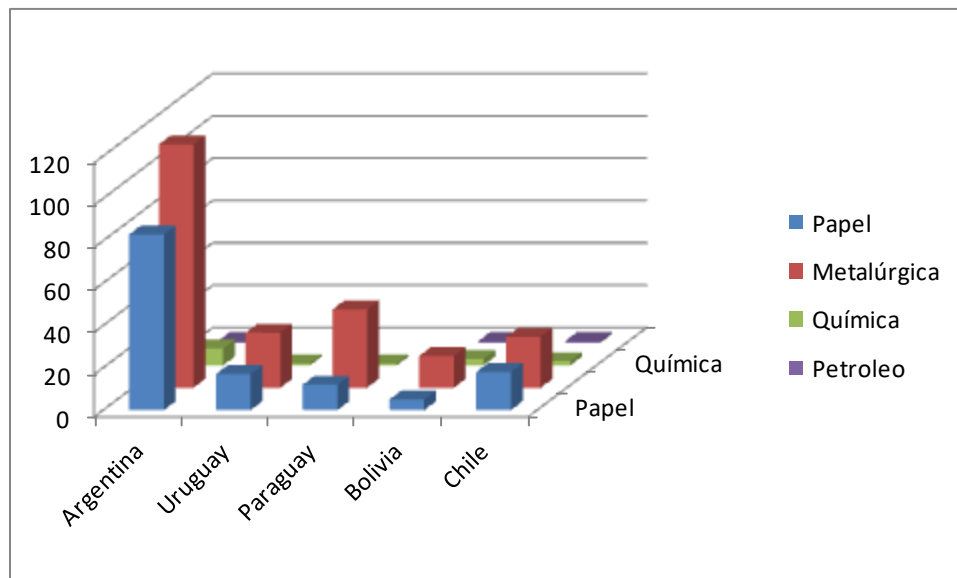


Esta acción es conocida como "drill down" (perforar)

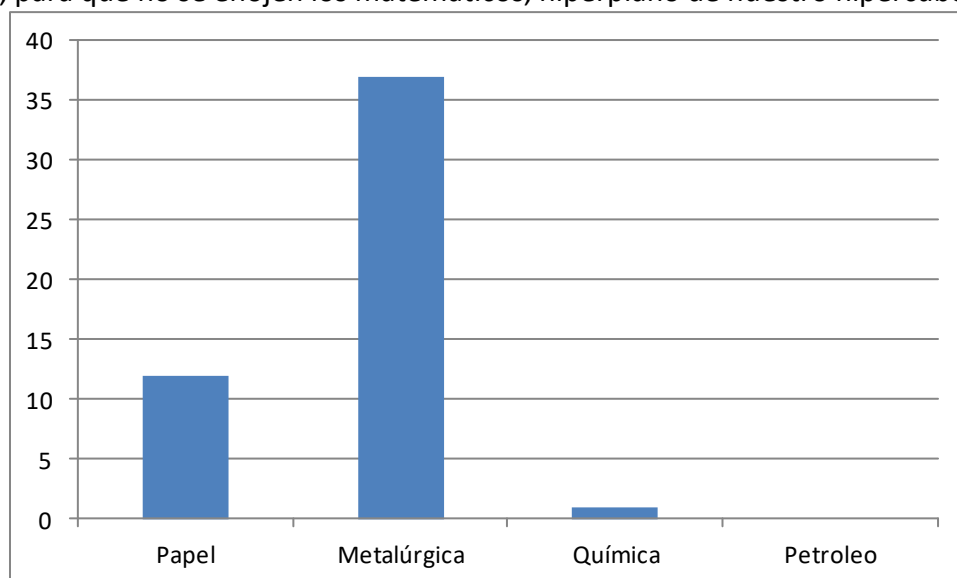
La herramienta OLAP puede ahora pararse en una industria (por ejemplo la metalúrgica y continuar con el proceso, por ejemplo, por país:



El efecto combinado puede mostrarse como:



Un analista atento podría percatarse de la importancia de Paraguay, para la cual no estaba preparado y pedirle a la herramienta OLAP que le muestre sólo este país cortando una "feta" o, para que no se enojen los matemáticos, hiperplano de nuestro hipercubo:



Esta operación se llama, en la jerga, "slicing"

Y así, mediante estas operaciones a las que la herramienta responde instantáneamente va procediendo el trabajo del analista.

Existen diversos conceptos alrededor del OLAP:

MOLAP representa "multi-dimensional online analytical processing". Es exactamente lo que se entiende habitualmente por OLAP. El aspecto multidimensional viene de la forma en la que se optimiza el almacenamiento de los totales precalculados.

El principal problema de MOLAP pasa por la necesidad de actualizar todo el cubo cada vez que se agrega información.

Otro problema se presenta cuando la cantidad de dimensiones involucradas es muy alta. Un síntoma de esta circunstancia es que el tamaño que ocupa el cubo se vuelve comparable con el que tiene la información original.

Algunas soluciones comerciales manejan el problema de tener muchas dimensiones mejor que otras. Se debe tener en cuenta este aspecto al elegir una solución concreta para un problema concreto.

Algunos productos introducen redundancia en los datos como para acelerar los tiempos de respuesta.

### ROLAP

ROLAP trabaja directamente sobre las bases de datos relacionales (de ahí le viene la R)

La principal limitación puede venir de la velocidad de respuesta. Simplemente se aprovecha la capacidad del SQL para componer las consultas necesarias para obtener en tiempo real los datos necesarios para la operación pedida.

### HOLAP

La H viene de Híbrido. HOLAP trata de combinar las ventajas de MOLAP y ROLAP. Pre procesa una parte de la información y gana ventajas de las técnicas de compresión aunque construye consultas para evitar los problemas que derivan de dimensiones muy poco pobladas.

En general mantienen una buena escalabilidad y reducen la cantidad de trabajo y almacenamiento necesarios para los procesos de actualización.

Otros conceptos mencionados suelen ser:

WOLAP: OLAP basado en la Web.

DOLAP: OLAP para computadoras de escritorio. (Por oposición a servidores)

No hay nada equivalente al ODBC para SQL en el sentido de un lenguaje estándar para conectarse a distintos servidores OLAP. Lo más similar fue la especificación de OLE DB para OLAP generada por Microsoft en 1997 y el lenguaje de consultas MDX.

### Tableros de comando:

Una empresa, de mediana en adelante, suele depender para su gestión exitosa, de múltiples factores. La falla de unos pocos puede arruinar el resultado de la mayoría.

Resulta imposible, para el equipo de gestión, consultar cada uno con la frecuencia necesaria si no están racionalmente agrupados en una sola herramienta.

Este es el desafío que intentan resolver los tableros de comando.

Una forma habitual de organizar un tablero de comando es asignando un indicador a cada área funcional de la organización.

En el tablero de comando de un gerente general se debería ver entonces un indicador para el área comercial, uno para finanzas, uno para operaciones y así siguiendo adelante.

Un primer problema es cómo combinar los operadores de distintas áreas para producir un resultado global.

Una primera aproximación es reducir todos los indicadores de todas las áreas a un valor entre 0 y 1 y tomar luego el promedio entre todos.

Esto tiene un problema grave. Permite que un buen desempeño en lo financiero, por ejemplo, "compense" un grave descalabro comercial. Dependiendo del tipo de organización estas compensaciones pueden tener más o menos sentido. En líneas generales no conviene dejar que se produzcan porque tienden a tapar problemas de gestión.

Una forma de evitar esto es haciendo que los indicadores se multipliquen entre sí. Se mantienen dentro del intervalo de 0 a 1 pero cualquiera de ellos que fracase estrepitosamente arrastra al conjunto.

Siguiendo con el mismo criterio el indicador de cada área podría también a su vez calcularse como un producto de sus factores principales.

Conviene que el gerente de cada área esté de acuerdo con el esquema de indicadores y valores de referencia propuestos para su sector. Este acuerdo debe formalizarse a la hora de construir el tablero de comando así como la "receta" precisa que se usará para el cálculo de cada factor.

La revisión en conjunto con el equipo gerencial del tablero de comando puede constituir el punto de partida para las reuniones periódicas de seguimiento.

También es posible construir los indicadores a lo largo de los procesos que animan a una organización y no por áreas. Esto dependerá de cada organización en particular.

Otra alternativa organizacional es la gestión por proyectos. En el caso de una empresa que se dedica, por ejemplo, a ejecutar proyectos para otras corresponderá entonces un tablero también proyectizado.

Si resultara necesario se puede dar a un indicador más peso que a otros afectándolo con una potencia.

Uno de los cuidados más difíciles a la hora de construir un tablero de comando es asegurarse que una situación mala para la organización no termine registrada como un buen resultado. Esto es particularmente crítico con el factor tiempo.

Es relativamente simple gestionar de manera de comerse reservas de capital no susceptible de registración contable mejorando el ejercicio actual pero haciendo inviable el futuro de la organización:

Ejemplos de estos intangibles son:

- Crédito de la empresa en el mercado
- Buen espíritu de los empleados y reserva de talentos
- Saber hacer del personal capacitado

El crédito de una empresa en el mercado le permite tomar recursos de sus proveedores a tasas muy convenientes y sin garantías. Esto resulta funcionalmente equivalente a tener capital propio aunque, desde la perspectiva contable esta posibilidad de crédito no alcanza expresión alguna.

Obviamente es muy delicado pensar como se expone un valor de esta naturaleza. No hacerlo prácticamente impulsa al equipo de gestión a consumir estas reservas para mejorar los parámetros que si se miden, en particular, cuando de ellos dependen su remuneración.

Algo similar pasa con un equipo motivado y con la reserva de talentos. La falta de actualización salarial va a ir en contra de ambas realidades. Sin embargo el efecto contable inmediato va a ser positivo. Liquidar el buen clima laboral y evaporar la reserva



de talentos puede parecer entonces, desde una perspectiva contable y cortoplacista un gran negocio. Un tablero de comando bien balanceado debiera evitar esta tentación al tener en cuenta estas realidades en su justa medida.

Finalmente, un tablero de control debe ser simple. La realidad detrás del mismo es lo suficientemente compleja como para darnos el lujo de agregarle nada innecesario. Es necesario que se pueda hacer "drill down" de los distintos indicadores para poder investigar el origen de los problemas que se vayan presentando.

Mientras la organización no alcance un estadio de madurez informativa tal que los números que dan cuenta de la acción son únicos, íntegros y controlados no resultará posible armar un tablero de control útil ya que cualquier conclusión será atacable desde la propia fuente.

Herramientas relacionadas con Business Intelligence:

Podemos intentar clasificar la multitud de herramientas disponibles de acuerdo a múltiples criterios:

1. De acuerdo a la forma de licenciamiento
  - a. Herramientas propietarias
  - b. Herramientas gratuitas
2. De acuerdo al fuerte de la herramienta
  - a. Herramientas de visualización
  - b. Herramientas de análisis

Ejemplos notables de herramientas propietarias son las de:

1. SAS
2. SPSS
3. Spotfire
4. Qlikview

Como herramientas gratuitas podemos citar a:

1. Weka
2. R
3. PSPP
4. Rapidminer
5. Orange

Cómo más orientado a la visualización que al análisis podemos destacar a Qlikview mientras que el resto tiene un fuerte componente analítico.

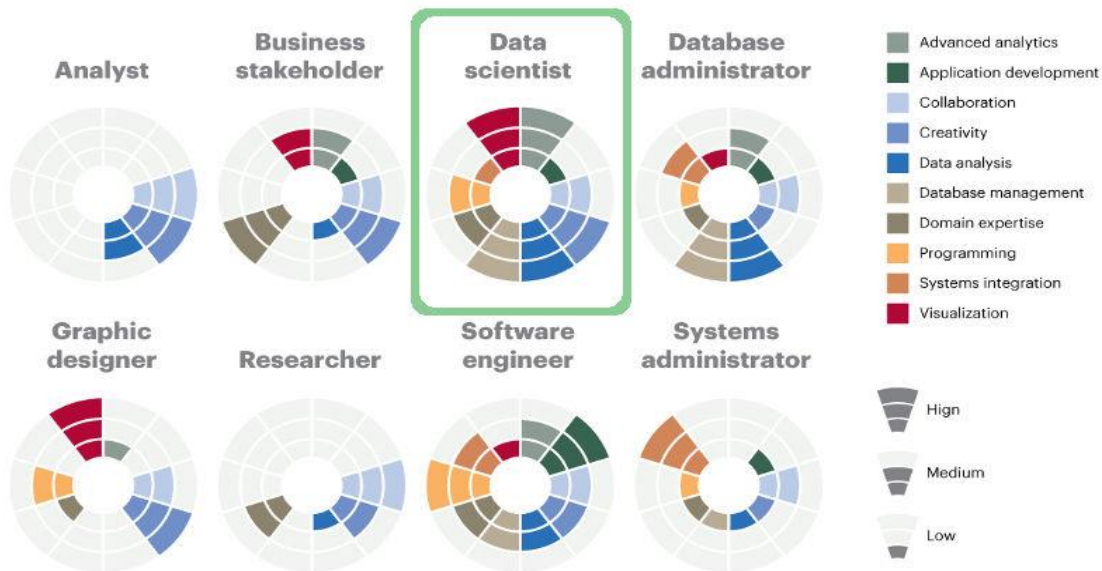
A lo largo del resto del curso iremos planteando herramientas para ser resueltas con las herramientas correspondientes.

### Taxonomía de las competencias de un científico de datos

Un científico de datos funciona como un equipo de una sola persona. Necesita manejar a un nivel avanzado una multiplicidad de temas.

Podemos recurrir al siguiente gráfico para hacernos una idea de la circunstancia:

**Needed skills by role for effective cross-functional IT and data science collaboration**



Source: A.T. Kearney analysis

Mientras que para las demás profesiones alcanza con dominar una o dos habilidades para el científico de datos necesitamos un mínimo de 5.

Esto puede verse como una grave complicación. Sin embargo se trata de una oportunidad ya que nos permite convertirnos en científicos de datos viniendo desde muy distintas profesiones.

Este es el motivo que causa que no pueda existir un científico de datos "junior" pues seguramente se trata de un profesional que ya era "senior" en otra de las profesiones listadas en el gráfico precedente.

### Instalando y cargando paquetes en R

La instalación de R es un proceso muy simple:

#### Descarga:

<https://cran.r-project.org/bin/windows/base/R-3.5.1-win.exe>

Una vez descargado ejecutarlo

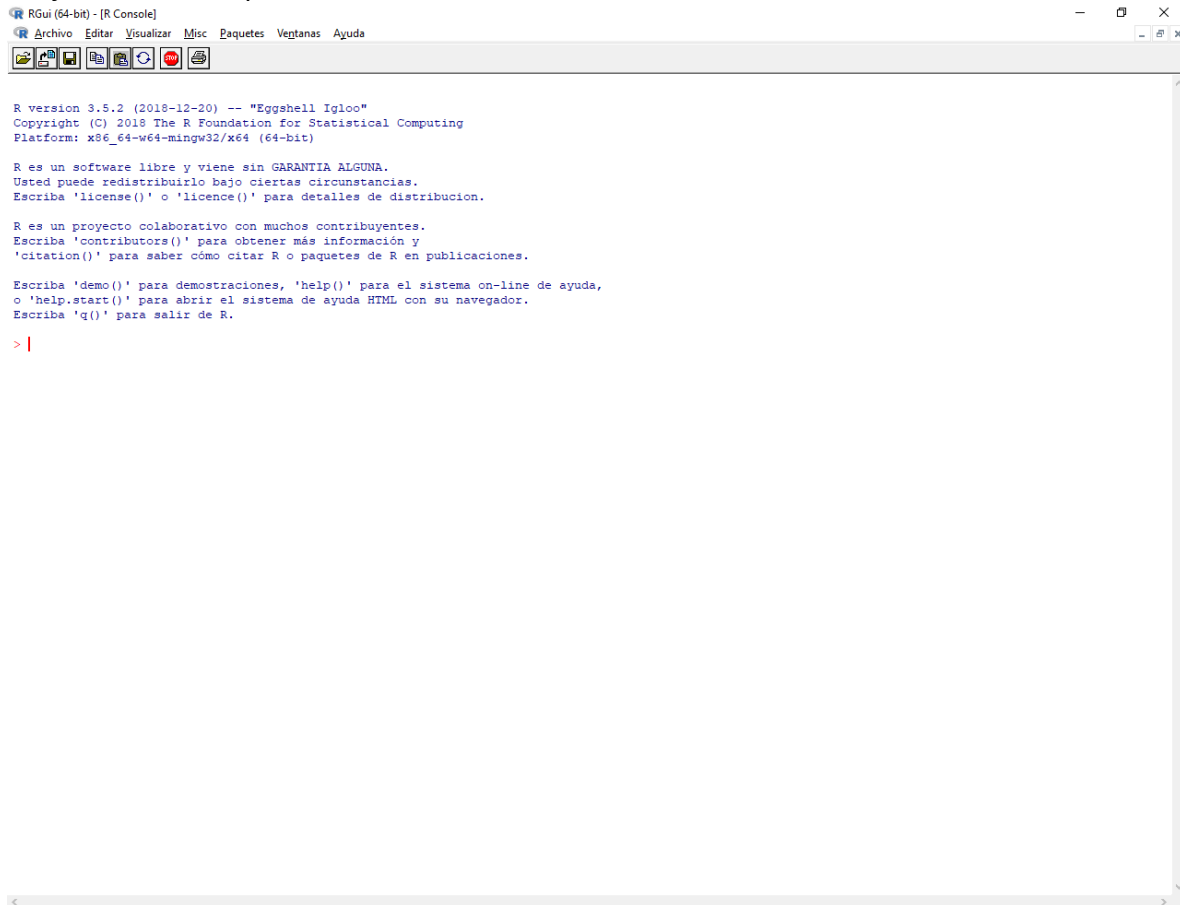
### Instalación:

Pide permisos de administrador.

Pide elegir el language de instalación

Seguir todas las opciones por defecto.

Al ejecutar R nos aparece:



```
R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R es un software libre y viene sin GARANTIA ALGUNA.
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribucion.

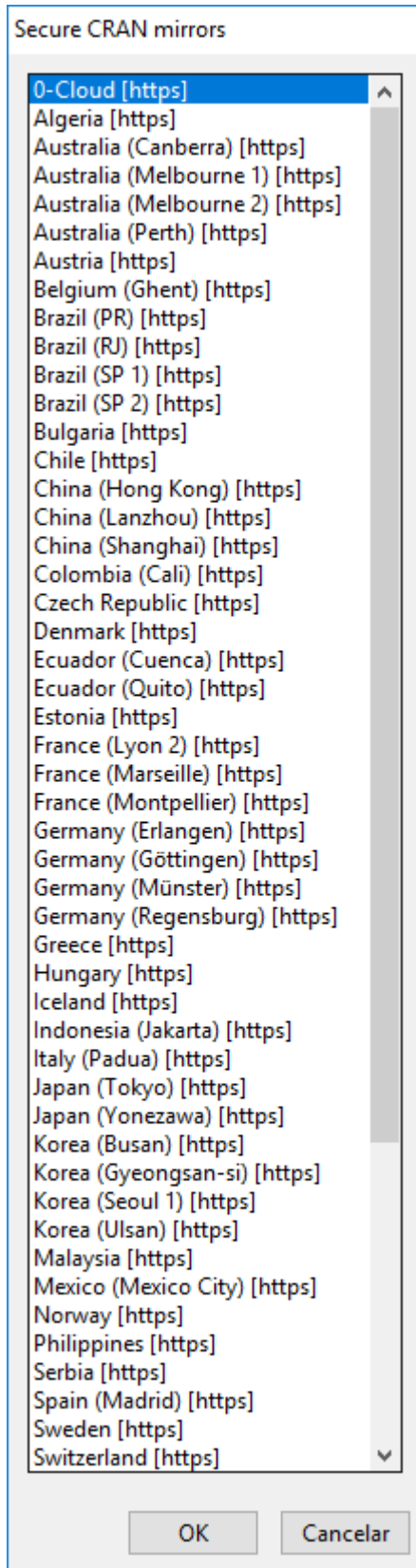
R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

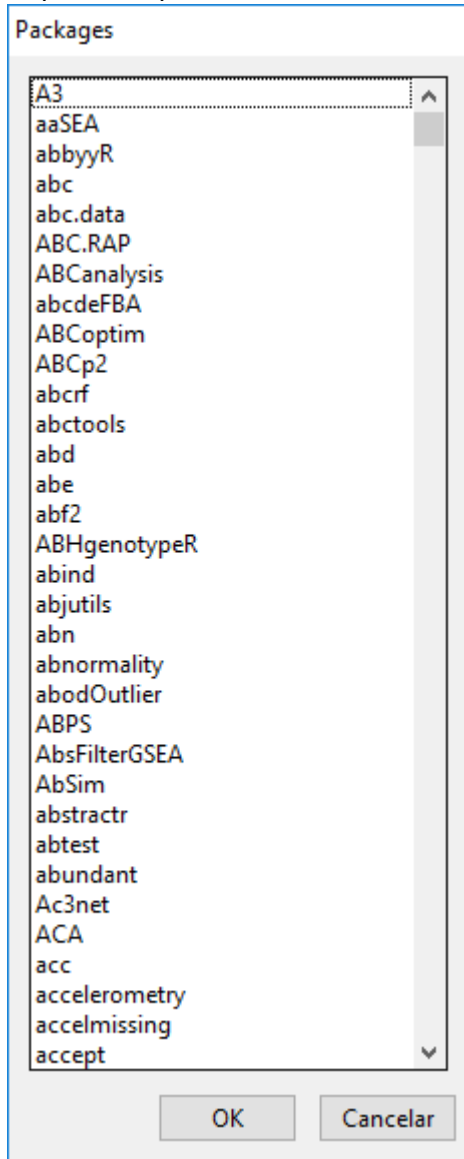
> |
```

Para descargar paquetes debemos dirigirnos (por sorprendente que parezca) a la opción paquetes del menú y elegir “instalar paquetes”

Lo primero que nos pide es seleccionar uno de los múltiples espejos desde los que podemos descargar los paquetes:



Una vez que elegimos un espejo nos despliega todos los paquetes que ese espejo tiene disponibles para nosotros:



Como podemos ver por el tamaño del botón de desplazamiento la cantidad de paquetes disponibles es inmensa. Esta es la principal riqueza de R. Hay muchísimo trabajo cristalizado en estos paquetes. Por añadidura, han sido usados tantas veces y por tantos profesionales que ha habido tiempo para pulir los bugs que pudieran haber tenido en el principio.

### Primeros pasos en R:

Una vez que se tiene R instalado es importante conocer comandos básicos de uso de terminal. Estos nos permitirán ubicarnos y trabajar más cómodos. A continuación una lista básica para empezar a movernos en el entorno R:

- saber en qué directorio estamos: `getwd()`
- fijar el directorio de trabajo: `setwd('nombre del directorio')`
- listar las variables: `ls()`
- ver qué paquetes están disponibles: `available.packages()`
- instalar un paquete (la lista de paquetes puede encontrarse en la página de CRAN): `install.packages("nameOfPackage")`
- llamar a un paquete instalado para usarlo en un desarrollo `library(nameOfPackage)`
- llamar funciones de paquetes instalados: `require` (a diferencia de `install.packages` que instala toda la librería, este comando sólo carga la función que se le indica)
- acceder a la ayuda (por ejemplo para la función `rnorm`)

```
?rnorm
```

- Buscar archivos de ayuda

```
help.search("rnorm")
```

- Obtener los argumentos de una función (por ejemplo `rnorm`)

```
args("rnorm")
```

```
## function (n, mean = 0, sd = 1)
## NULL
```

- ver el código fuente de una función

```
rnorm
## function (n, mean = 0, sd = 1)
## .Call(C_rnorm, n, mean, sd)
## <bytecode: 0x9d01c74>
## <environment: namespace:stats>
```

## Vamos a la fuente: los datos

Anteriormente mencionamos que lo más importante para un científico de datos es formular la pregunta correcta para el problema de negocio en cuestión. Seguido en importancia se encuentran los datos en sí, es decir, qué datos son relevantes, a cuáles datos puedo acceder, en qué formato y cómo conviene almacenarlos, qué criterios conviene usar para limpiarlos, etc.

Ahora bien, ¿qué son los datos? Empezaremos por mencionar algunos conceptos que debemos tener en cuenta a la hora de hablar de datos. Más adelante volveremos sobre los tipos de datos específicos que existen en R y los analizaremos en más detalle a través de ejemplos.

- **Datos:** valores cualitativos o cuantitativos de variables que pertenecen a un conjunto de items (o población, en el sentido estadístico). No siempre se encuentran bien estructurados y pueden almacenarse en diferentes formatos (archivos de texto, bases de datos relacionales y no relacionales, etc).
- **Variables:** mediciones/características de un item. Pueden ser categóricas (por ejemplo el color de una prenda de vestir) o cuantitativas (por ejemplo el ancho de la prenda). R posee funciones específicas para trabajar con todos los tipos de datos, sean estos numéricos, de cadena (*strings*), fechas, etc. Es muy importante notar que de acuerdo al tipo de dato, R emplea distintos **objetos** para almacenarlos en forma eficiente (y útil). Por ejemplo, los datos pueden organizarse en vectores, matrices, listas, *corpus* (es un conjunto de documentos de texto, muy utilizado en minería de texto), y uno de los más útiles en R, los **dataframes**. A lo largo del curso volveremos seguido sobre este último objeto, pues es una de las características más poderosas de R.

R cuenta con un gran número de repositorios de datos a los cuales podemos acceder para, por ejemplo, practicar! También son muy útiles para testear nuestros desarrollos, porque son datos muy conocidos y al utilizarlos en nuestros desarrollos podemos asegurarnos de que los mismos son correctos.

Para acceder a ellos, basta cargar la librería `datasets`. Luego, con el comando `data` podemos cargar el conjunto de datos que querramos. Por ejemplo, si queremos cargar el dataset llamado `iris`, que contiene datos sobre las medidas de pétalos y tallos de tres tipos de flores, escribimos

```
library(datasets)
data(iris)
```

y ya tendremos cargado el set. Nos adelantamos un poco usando el comando `head` que nos permite ver los primeros miembros de un set de datos. A lo largo del curso volveremos a encontrar a este comando por ser muy útil. Veamos qué tiene el conjunto de datos `iris`:

```
head(iris)
```

##	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	5.1	3.5	1.4	0.2	setosa

```
## 2      4.9      3.0      1.4      0.2 setosa
## 3      4.7      3.2      1.3      0.2 setosa
## 4      4.6      3.1      1.5      0.2 setosa
## 5      5.0      3.6      1.4      0.2 setosa
## 6      5.4      3.9      1.7      0.4 setosa
```

Vemos entonces que pudimos cargar el set de datos iris.

## Operaciones básicas en R

A continuación, veremos algunas operaciones básicas en R, como por ejemplo crear una secuencia de números consecutivos o que se repitan. A lo largo del curso iremos encontrando estos comandos una y otra vez; la intención de discutirlos aquí es simplemente presentarlos para que podamos comenzar a manejarnos con confianza dentro del entorno de R.

Vamos a definir primero algunos vectores que nos servirán de ejemplo de muchos comandos importantes. Hay otras maneras de definir vectores (y matrices, etc.) que veremos en la próxima unidad; por ahora nos quedaremos con la manera más usual.

```
a <- c(1,2,3,4)
a
## [1] 1 2 3 4

b <- seq(5,9)
b
## [1] 5 6 7 8 9

c <- seq(5,9,by=2)
c
## [1] 5 7 9
```

Observando la salida de estos comandos notamos varias cosas importantes. Para definir el vector a, utilizamos el comando `c()`, que toma como argumento una lista de objetos (que en este caso son números, del 1 al 4).

El comando `c()` es un concatenador de objetos; en el primer ejemplo está concatenando los números 1 al 4, pero podría estar concatenando diversos objetos, como ser caracteres o cadenas de caracteres (*strings*), algo que iremos viendo a lo largo del curso. Para entender mejor este operador, veamos el siguiente ejemplo:

Tomamos el vector a y queremos agregarle el número 5 al final o al principio:

```
d <- c(a,5)
d
## [1] 1 2 3 4 5
```



ó

```
d <- c(5,a)
d
```

```
## [1] 5 1 2 3 4
```

Podríamos también agregarle otro vector, por ejemplo b:

```
d <- c(a,b)
d
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

Antes de continuar, notemos brevemente que valen todas las operaciones estándar sobre estos vectores, como por ejemplo la suma:

```
e <- a+b
```

```
## Warning in a + b: longer object length is not a multiple of shorter
object
## length
```

```
e
```

```
## [1] 6 8 10 12 10
```

También podemos comparar vectores (en general podemos comparar objetos de la misma clase; volveremos sobre este punto importante más adelante). Veamos un ejemplo de comparación utilizando el operador == (igual a):

```
a==b
```

```
## Warning in a == b: longer object length is not a multiple of shorter
## object length
```

```
## [1] FALSE FALSE FALSE FALSE FALSE
```

nos devuelve 'FALSE, FALSE, FALSE, FALSE', lo que indica que el primer elemento de a no es igual al primer elemento de b, el segundo elemento de a no es igual al segundo elemento de b, etc. Entonces, para vectores, vemos que la comparación es uno a uno: primer elemento con primer elemento, y así.

Ya que estamos, y a modo introductorio porque esto lo volveremos a ver en gran detalle en las unidades que siguen, veamos qué sucede con cadenas de caracteres, definidas entre comillas (simples o dobles). Si tenemos dos cadenas:

```
cad_1 <- 'hola'
cad_1
```

```
## [1] "hola"
```

y

```
cad_2 <- 'que tal'
cad_2
```

```
## [1] "que tal"
```

podemos concatenarlas con el comando **c()** de la misma forma que antes:

```
cad_3 <- c(cad_1,cad_2)
cad_3
```

```
## [1] "hola"      "que tal"
```

Sigamos con los vectores numéricos b y c. Aquí, utilizamos el comando **seq(x,y,opciones)** para definir dos vectores, b y c. Este comando sirve para generar secuencias, en el caso de b desde el 5 hasta el 9 (yendo de a 1 que es el comportamiento por default), y en el caso de c desde 5 hasta 9, pero yendo de a dos en dos.

Otros tres comandos muy útiles (y que uno termina usando casi todo el tiempo) son los que siguen:

```
long <- length(a)
long
```

```
## [1] 4
```

que da el número de elementos del vector,

```
suma <- sum(a)
suma
```

```
## [1] 10
```

que da la suma de sus elementos, y

```
uni <- unique(a)
uni
```

```
## [1] 1 2 3 4
```

que devuelve los elementos únicos del vector (en este caso nos devuelve el vector original porque no hay elementos repetidos). Otro ejemplo de **unique**:

```
f <- c(a,1,2)
f
```

```
## [1] 1 2 3 4 1 2
```

```
uni <- unique(f)
uni
```

```
## [1] 1 2 3 4
```

Los números 1 y 2 en f (que es la concatenación de a con 1 y 2) están repetidos, y por eso sólo aparecen una vez al aplicar **unique**. Es importante notar que esto mismo se aplica a cadenas de caracteres. Por ejemplo:

```
f <- c("hola","chau","que tal","hola","hola","chau")
f
## [1] "hola"      "chau"      "que tal"   "hola"      "hola"      "chau"
uni <- unique(f)
uni
## [1] "hola"      "chau"      "que tal"
```

En la unidad 2 volveremos sobre estos comandos y muchos otros relativos al manejo de distintos tipos de datos como vectores, matrices, dataframes y cadenas, entre otros y veremos también cómo acceder y manipular los elementos de estos objetos.

### Lectura de archivos

Para terminar esta introducción, veremos como levantar datos de un archivo en formato csv (*comma separated values*) y guardar esos datos en una variable para su posterior uso. En la unidades siguientes iremos viendo cómo levantar datos de distintos tipos de archivos y otras fuentes, pero aquí comenzamos con el formato csv como ejemplo sencillo y útil. En muchos casos los datos de entrada están en este formato, por lo cual es muy útil saber cómo manejarse con este tipo de archivos.

Si bien hay varias formas equivalentes de levantar datos de un archivo csv, que veremos en la próxima unidad, tal vez la forma más directa de leerlo sea utilizar el comando **read.csv** que ya viene precargado. Supongamos que tenemos un archivo llamado *prueba.csv* en el directorio actual. Queremos leerlo y guardarlo en la variable datos. Para ello, escribimos `datos <- read.csv("prueba.csv")`

Algunos puntos importantes para tener en cuenta, que son generales a R e iremos viendo en todo el curso. Notemos que asignamos una variable utilizando el operador `<-`. Esto se interpreta de la misma forma que un signo `=`; incluso podríamos usar un signo `=` para esta asignación pero el operador `<-` (y su contraparte `->`) son más generales. Notemos también que en la función **read.csv** hemos incluido el nombre del archivo entre comillas dobles (podrían ser simples también) por tratarse de un *string*.