Preprocesamiento y estructuración de datos II

Tema 3. (continuación)

unir LA UNIVERSIDAD EN INTERNET

Profesor: Luis F. Ruiz

¿Qué veremos hoy?

01 ¿Por qué R y Python para preprocesamiento?

02 El reto de integrar múltiples fuentes.

03 Preparación de datos para dashboards.

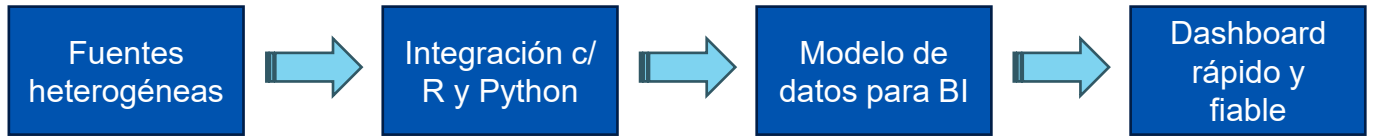
04 Resumen y conclusiones.

05 Ejemplos Tableau para Actividad 1.

¿Por qué R y Python para preprocesamiento?

- ✓ Más del 70 % del tiempo de un proyecto de visualización es preparar datos.
- ✓ Herramientas visuales + código ➔ combinación óptima.
- ✓ R (tidyverse: dplyr, tidyr, readr) y Python (pandas, numpy) para:
  - ➢ Integrar múltiples fuentes (CSV, SQL, APIs).
  - ➢ Automatizar limpiezas y validaciones.
  - ➢ Generar DataFrames ya estructurados para el dashboard.

Dibujar flujos del tipo …



3.5. El reto de integrar múltiples fuentes

Integrar datos no es simplemente concatenar tablas, sino ***definir relaciones, homogeneizar formatos y garantizar integridad referencial.***

- **Escenario típico de empresa:**
  - Excel/CSV locales (reportes manuales).
  - Bases de datos (ERP, CRM, e-commerce).
  - APIs (redes sociales, analítica web, tipos de cambio…).

- **No es solo "importar":**
  - Definir claves de relación (ID cliente, producto, fecha…).
  - Resolver conflictos de formato (fechas, unidades, categorías).
  - Documentar origen y reglas de integración (trazabilidad).

Ejemplo muy habitual

Ventas (SQL) + Clientes (Excel) + Tipos de cambio (API).

Ejemplo en Python: uniendo SQL + Excel + API

1. **Conectar a fuentes**
   - pandas.read_sql(),
   - read_excel()
   - requests.get() para API JSON.

   ➢ Traer la tabla de VENTAS de una BDD SQL.
   ➢ Leer el Excel de CLIENTES.
   ➢ Llamar a una API de CAMBIO (JSON)
2. **Unificar formato**
   - Fechas → pd.to_datetime()
   - Moneda → aplicar tipos de cambio.

   ➢ Convertir las fechas para estandarizar.
   ➢ Calcular importe en € (cambio/día).
3. **Unir tablas**
   - pd.merge(ventas, clientes, on="id_cliente")
   - pd.merge(..., tipos_cambio, on="fecha").

   ➢ Unir VENTAS c/ CLIENTES (por id_cliente).
   ➢ Unir con TIPOS de CAMBIO por FECHA.
4. **Guardar dataset integrado**
   - A CSV / base de datos / parquet.

   ➢ DataFrame único integrado y limpio.

Ejemplo en R: integración con tidyverse

Dos ideas importantes:

- El papel de las **claves de relación**.
- La **homogeneización de formatos** (fechas, monedas) antes de integrar.

```r

library(DBI); library(readxl)
library(dplyr); library(httr); library(jsonlite)

ventas   <- dbGetQuery(con, "SELECT * FROM ventas")
clientes <- read_excel("clientes.xlsx")

resp  <- GET("https://api.tipos.com/hoy")
tc_df <- fromJSON(content(resp, "text")) |>
  as_tibble()

datos <- ventas %>%
  left_join(clientes, by = "id_cliente") %>%
  left_join(tc_df,    by = "fecha") %>%
  mutate(importe_eur = importe * tipo_cambio)
```

DBI conecta a la base SQL; readxl trae los clientes de Excel,
httr y jsonlite consumen una API de tipos de cambio y la convierten en tibble.

Luego, con dos left_join() unimos tablas por
id_cliente y fecha ➔ generamos importe_eur.

Homogeneización y resolución de conflictos

No es buena práctica mezclar categorías distintas sin documentar; es clave ***homogeneizar y documentar*** *criterios de integración.*

Integración = Limpieza + Reglas claras

Problemas típicos al integrar fuentes heterogéneas:

- **Fechas**: distintos formatos (DD/MM/AAAA, YYYY-MM-DD).
- **Categorías**: “Online”, “on line”, “ON-LINE”.
- **Duplicados**: mismo cliente / transacción desde dos sistemas.

Soluciones programáticas:

- **R**:
  - lubridate estandariza fechas,
  - stringr limpia cadenas
  - distinct() ayuda a desduplicar.
- **Python**:
  - con pandas usamos pd.to_datetime,
  - métodos .str para normalizar textos,
  - drop_duplicates().

3.6. Preparación de datos para dashboards: pensar en el modelo

- Objetivo: adaptar datos integrados a las necesidades del dashboard.
- Diseño del **modelo de datos**:
  - Tablas de **hechos** (ventas, operaciones…).
  - Tablas de **dimensiones** (fecha, cliente, producto, canal…).
- Esquema estrella / copo de nieve ➔ relaciones claras (1–N).
- R/Python generan DataFrames ya con esa estructura.

Ejemplo

- Hechos: hechos_ventas
- Dimensiones: dim_clientes, dim_productos, dim_tiempo.

Agregación y rendimiento con R y Python: no siempre necesitamos el detalle

Buenas prácticas para dashboards

- Reducir **granularidad** si no hace falta detalle (diario ➔ mensual).
- Eliminar columnas **innecesarias** para el análisis.
- Precalcular métricas complejas en el script (no solo en el BI).

Herramientas:

- R: group_by() %>% summarise(), select().
- Python: df.groupby(...).agg(...), df[cols].

*Una técnica clave para mejorar rendimiento es **reducir la granularidad innecesaria**, no cargar todos los detalles si no los vas a visualizar.*

Ejemplo

- R: sumar ventas por mes y producto antes de exportar.
- Python: crear tabla ventas_mensuales para que el dashboard vuele.

Validar y automatizar: que el dashboard no “mienta”

Validación programática:

- Reglas sobre nulos, rangos, unicidad de claves.
- Librerías:
  - R: assertr, validate.
  - Python: pandera.

Trazabilidad y almacenamiento intermedio (APIs):

- Guardar históricos en SQL / ficheros ➔ **trazabilidad**.

Automatizar:

- Programar scripts de R/Python (cron, scheduler, orquestadores).
- Que el dashboard lea siempre datos ya validados.

3.7. Resumen y conclusiones

Checklist para integrar con cabeza

✓El núcleo de la visualización avanzada es el **preprocesamiento**.

✓Integrar múltiples fuentes = definir claves, homogeneizar formatos, resolver duplicidades.

✓R y Python permiten:

- Conectar a archivos, bases de datos y APIs.
- Transformar y validar con flexibilidad.
- Generar modelos “lista de hechos + dimensiones” para dashboards.

✓Preparar datos pensando en rendimiento: agregación y selección de columnas.

✓Automatizar y documentar ➔ dashboards fiables y sostenibles en el tiempo

Práctica: Carga, Limpieza y Transformación

en Tableau Desktop Public

unir LA UNIVERSIDAD EN INTERNET

1. Cargar el Excel en Tableau

1. Abrimos **Tableau**.
2. En la pantalla inicial, en **Conectar** → **Archivos** → hacemos clic en **Microsoft Excel**.
3. Seleccionamos AdventureWorks Sales.xlsx ➔ **Abrir**.
4. Tableau nos lleva a la pestaña **Fuente de datos**:
   - Arriba vemos la conexión.
   - Debajo, arrastramos la hoja principal (por ejemplo Sales, Hoja1, etc.) al lienzo si no se ha puesto sola.
   - Abajo vemos una **tabla tipo Excel** con los datos.

En el panel izquierdo verás ahora las **hojas/tablas** del Excel (por ejemplo: Sales, Customers, etc.). Para esta práctica usaremos la hoja principal de ventas (suele llamarse algo tipo **Sales**, **Hoja1** o similar).

Tableau Public – Book1

File Data Window Help

Tableau Desktop Public Edition

Buy Tableau

Connections

Add

AdventureWorks Sales
Microsoft Excel

Sheets

Use Data Interpreter

Data Interpreter might be able to clean your Microsoft Excel workbook.

- Customer_data
- Date_data
- Product_data
- Reseller_data
- Sales Order_data
- Sales Territory_data
- Sales_data

New Union

New Table Extension

Go to Worksheet

Sales_data (AdventureWorks Sales)

Filters
0 | Add

Sales_data

Sales_data | 15 fields 121253 rows | 100 rows

Name
Sales_data

Fields

Type	Field Name	Physical Table	Remot...

Sales!data Sales Order Line Key	Sales!data Reseller Key	Sales!data Customer Key
43659001	676	-1
43659002	676	-1
43659003	676	-1

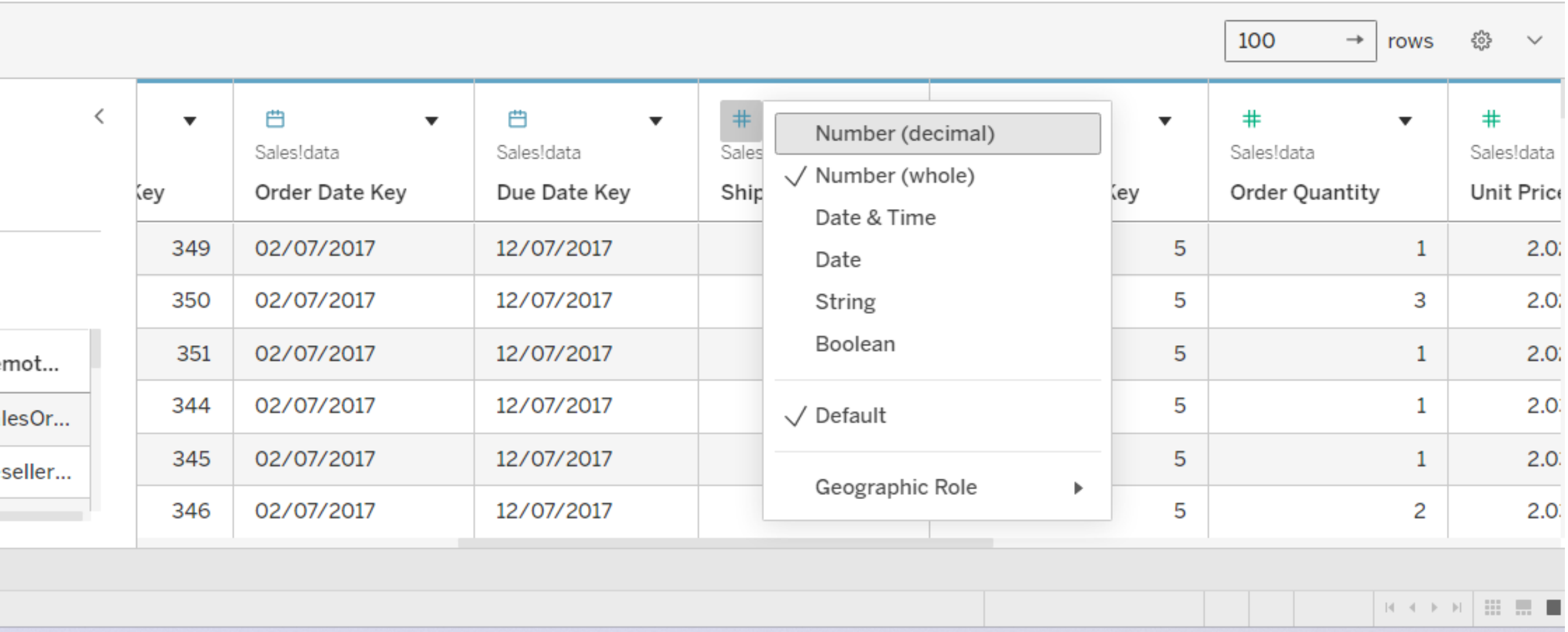
Data Source | Sheet 1

2. Revisar campos y tipos de datos

En la pestaña **Fuente de datos**:

1. Vemos todas las columnas en la cuadrícula.
2. Encima de cada nombre de campo hay un pequeño icono (Abc, #, calendario, etc.), que indica **tipo de dato**:
   - Localizamos la columna de fecha (OrderDate / FechaPedido):
     - Hacemos clic en el icono → selecciona **Fecha**.
   - Localiza la columna de importe (SalesAmount / ImporteVenta):
     - Icono → selecciona **Número (decimal)**.
   - Localiza la columna de cantidad (OrderQty / Cantidad):
     - Icono → selecciona **Número (entero)**.

Esto es el equivalente a ajustar tipos en la “vista de perfil” de Tableau Prep, pero aquí lo hacemos en la **Fuente de datos**.


100
rows
Sales!data
Order Date Key
Sales!data
Due Date Key
Sales!data
Order Quantity
Sales!data
Unit Price
349 02/07/2017 12/07/2017 5 1
350 02/07/2017 12/07/2017 5 3
351 02/07/2017 12/07/2017 5 1
344 02/07/2017 12/07/2017 5 1
345 02/07/2017 12/07/2017 5 1
346 02/07/2017 12/07/2017 5 2
Number (decimal)
Number (whole)
Date & Time
Date
String
Boolean
Default
Geographic Role

3. Limpieza (filtros básicos)

Tableau Desktop Public no tiene los pasos de flujo de Prep, pero podemos hacer limpieza lógica a través de **filtros** y **campos calculados**, por ejemplo:

3.1 Crear un campo calculado para Precio Unitario

1. Vamos a una hoja nueva (**Hoja 1**, botón abajo).
2. En el panel izquierdo, en **Datos**, hacemos clic derecho en el espacio en blanco → **Crear** → **Campo calculado…**
3. Nombre: PrecioUnitario.
4. Fórmula (ajusta nombres de campos reales):

   [SalesAmount] / [OrderQty] (código)
5. Pulsamos **Aceptar**.

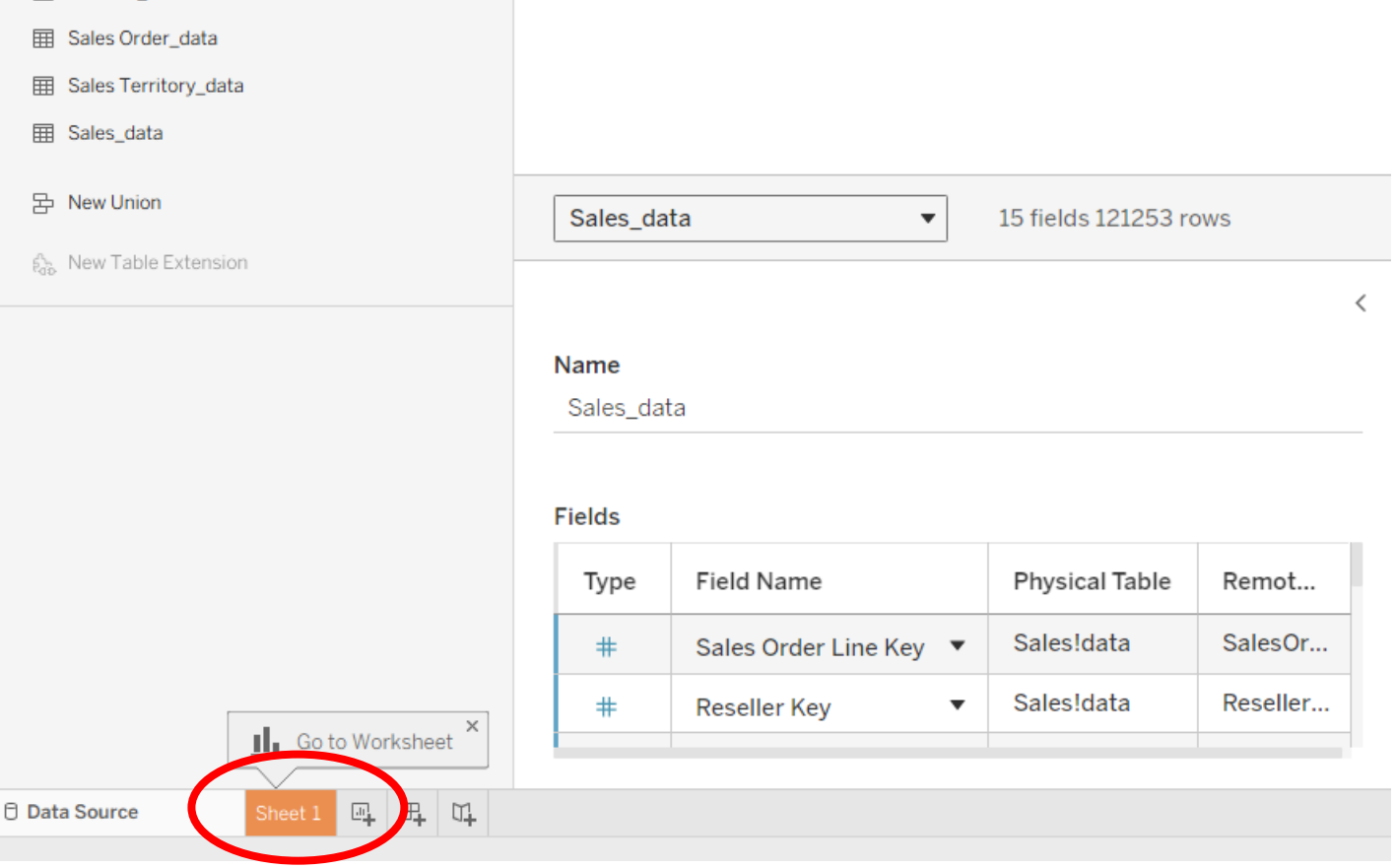
Ya tenemos una métrica derivada.

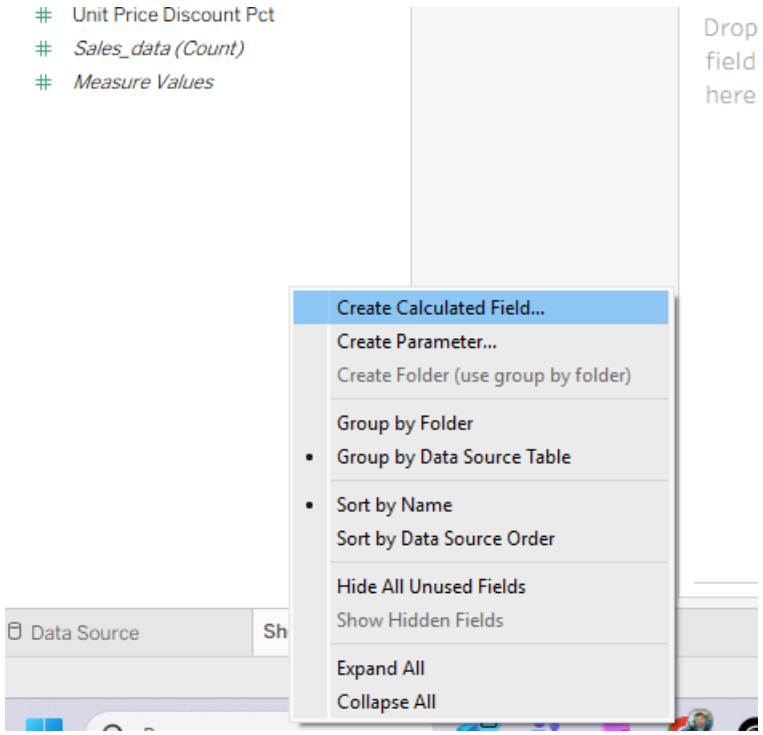
3.2 Filtrar datos "raros"

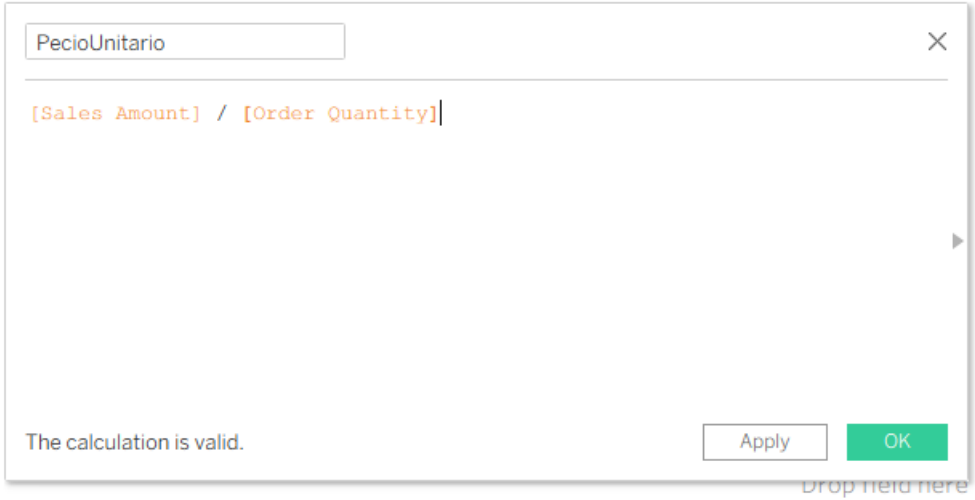
En la misma Hoja 1:

1. Arrastramos el campo Cantidad (OrderQty) al estante de Filtros.
2. En el cuadro de diálogo:
   - Seleccionamos **Rango de valores**.
   - Definimos **mínimo** 1 (para quitar 0 y negativos).
3. Opcional: si sospechamos que hay nulos en importe, podemos:
   - Arrastrar ImporteVenta al estante Filtros → Especial → No nulos.

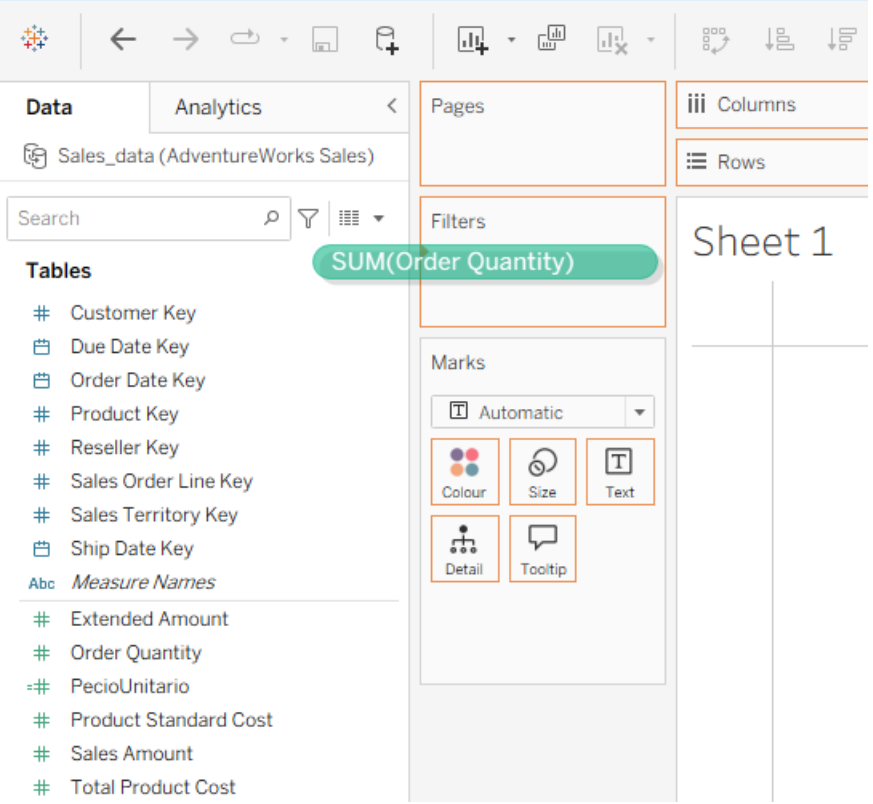
Esto no "borra" filas, pero **las excluye del análisis**, que es lo que necesitamos para los gráficos.

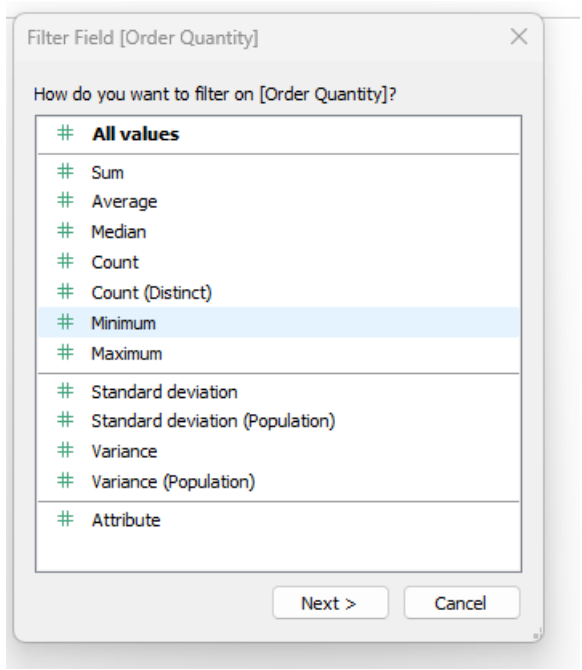


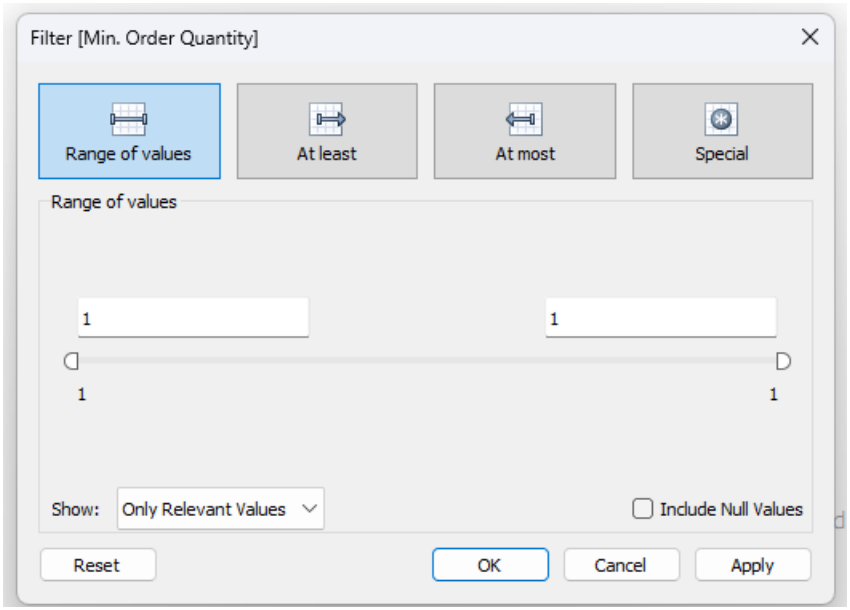




unir LA UNIVERSIDAD EN INTERNET


Data
Analytics
Sales_data (AdventureWorks Sales)
Search
Tables
Customer Key
Due Date Key
Order Date Key
Product Key
Reseller Key
Sales Order Line Key
Sales Territory Key
Ship Date Key
Measure Names
Extended Amount
Order Quantity
PecioUnitario
Product Standard Cost
Sales Amount
Total Product Cost
Pages
Filters
SUM(Order Quantity)
Marks
Automatic
Colour
Size
Text
Detail
Tooltip
Columns
Rows
Sheet 1


Filter Field [Order Quantity]
How do you want to filter on [Order Quantity]?
All values
Sum
Average
Median
Count
Count (Distinct)
Minimum
Maximum
Standard deviation
Standard deviation (Population)
Variance
Variance (Population)
Attribute
Next >
Cancel


Filter [Min. Order Quantity]
Range of values
At least
At most
Special
Range of values
1
1
1
1
Show:
Only Relevant Values
Include Null Values
Reset
OK
Cancel
Apply

4. Comprobar en un gráfico

Para cerrar la mini–práctica:

1. En **Hoja 1**, creamos un gráfico de prueba:
    - Arrastramos Producto o Categoría a **Filas**.
    - Arrastramos PrecioUnitario a **Columnas**.
2. Cambiamos el tipo de gráfico a **Barras** si no lo está ya (en “Mostrarme”).

5. Recap

Hemos visto:

- ✓ Cómo se **carga** el Excel.
- ✓ Cómo se **ajustan tipos de datos**.
- ✓ Cómo se crea un **campo calculado**.
- ✓ Cómo se **limpia por filtro** registros no válidos (cantidades ≤ 0, nulos).

Muchas gracias por vuestra atención

www.unir.net