UNIVERSITÀ DEGLI STUDI DI TERAMO

Big Data Analytics

Riconoscere strutture in KNIME
Tutorial

Prof.ssa Romina Eramo
Università degli Studi di Teramo
Dipartimento di Scienze della Comunicazione
reramo@unite.it

Tutorial: creare gruppi di consumatori

» Applicare algoritmi di **clustering** per:
  - creare **gruppi omogenei di consumatori**
  - supportare **strategie di comunicazione differenziate**
  - trasformare l'analisi dei dati in **azioni di business**

» Il tutorial copre sia:
  - la **definizione dei cluster**
  - sia la **preparazione e interpretazione del dataset**

Contesto

» **Il caso**
  - negozio online di articoli da regalo e gadget (Londra)
  - forte crescita del business
  - nascita di un primo nucleo di **CRM (Customer Relationship Management)**

» **Esigenza**
  - creare **strategie di comunicazione personalizzate**
  - aumentare **loyalty** e **fatturato**

Problema analitico

» Il negozio ha:
  - oltre **4.000 clienti attivi**
  - storico delle **transazioni**
  - poche informazioni anagrafiche

» Serve:
  - ridurre la complessità
  - passare da transazioni a **profili cliente**

Variabili descrittive del cliente

» Per caratterizzare i clienti scegliamo 4 indicatori:

- **Frequenza**
  - » numero di atti d'acquisto
  - » indicatore di lealtà
- **Dimensione del carrello**
  - » numero medio di unità per acquisto
- **Prezzo medio**
  - » indica preferenza per prodotti economici o premium
- **Numero di articoli diversi**
  - » misura varietà vs focalizzazione degli acquisti

Dataset di partenza

» File **Ecommerce-Clean.csv**
  - versione ripulita delle transazioni *
  - una riga per **articolo/fattura**

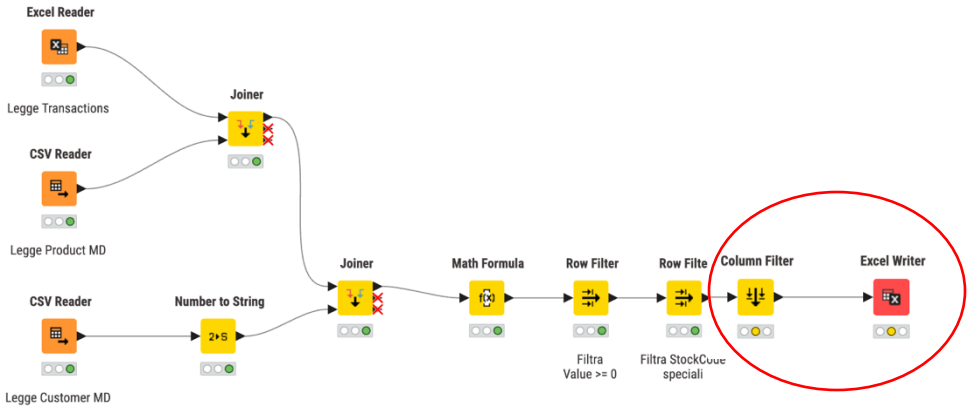
» Colonne principali:
  - InvoiceNo
  - CustomerID
  - StockCode
  - Quantity
  - UnitPrice
  - Value

* Vedi slide successiva per dettagli dataset

Dataset di partenza

» Il dataset si può ottenere in 2 modi diversi:

- Esportare i dati ottenuti (uniti e manipolati) nel tutorial Ecommerce (lezione 5) e filtrare le colonne necessarie



- Scaricare il dataset disponibile qui https://www.kaggle.com/datasets/aliessamali/ecommerce e calcolare/aggiungere il campo Value

Dal dato transazionale al dato cliente

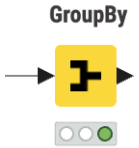
» Problema:
  - il dataset contiene **più di 500.000 righe**
  - ma serve una tabella con **una riga per cliente**

» Soluzione:
  - **aggregare** le transazioni
  - calcolare le variabili descrittive per cliente

Strategia di aggregazione

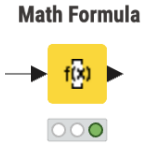
» Per le prime tre variabili:

- aggregazione **per fattura**



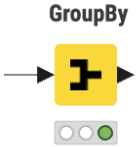
Gruppi: InvoiceNo e CustomerID
Aggregazione: Quantity (per somma) e Value (per somma)

- calcolar eil Prezzo medio per unità



Espressione: $Sum(Value)$/$Sum(Quantity)$
Append column: Price

- poi aggregazione **per cliente**

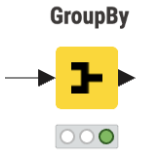


Gruppi: CustomerID
Aggregazione: Sum(Quantity) e Price per media (Mean), mentre il num delle fatture generate sarà ottenuto con il conteggio (Count) di una qualsiasi colonna, ad esempio InvoiceNo

Strategia di aggregazione

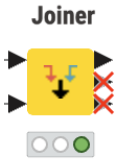
» Per la quarta variabile dobbiamo creare un percorso alternativo nel nostro workplow:

- aggregazione diretta per cliente per contare gli articoli che ogni cliente acquista almeno una volta



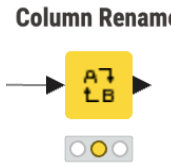
Gruppi: CustomerID
Aggregazione: StockCode (usando Unique Count)

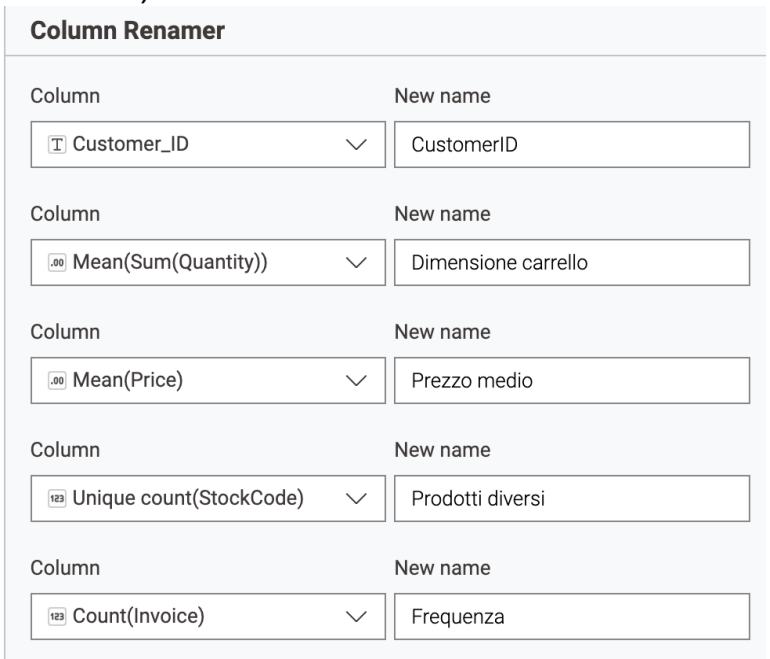
- Combiniamo le tabelle ottenute



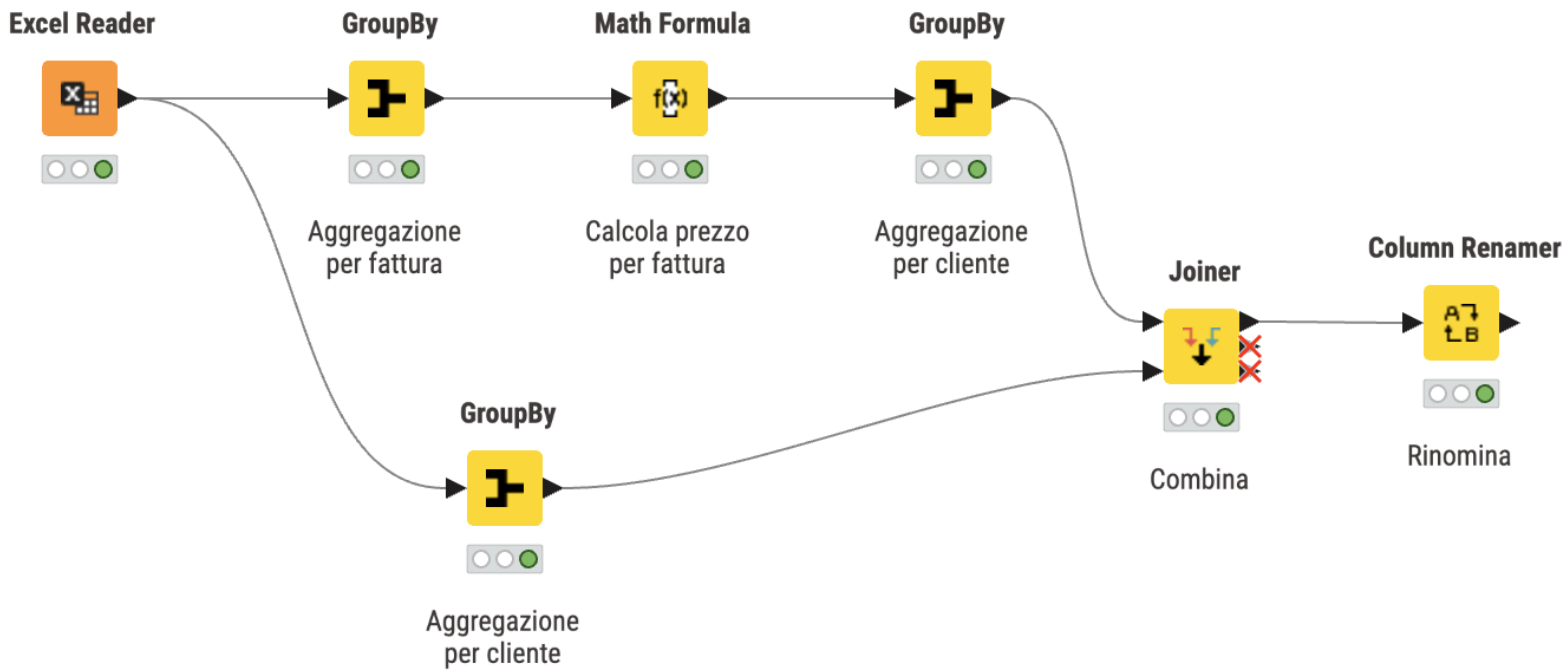
Criteri: CustomerID e CustomerID

- Rinominiamo le colonne

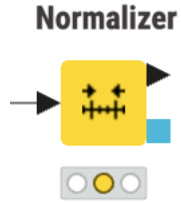




Porzione del workflow che si occupa di ottenere i quattro indicatori di preferenze di acquisto per cliente



Normalizer



» Il nodo **Normalizer** consente di:
- normalizzare i valori numerici di una tabella
- ridurre le differenze di scala tra le colonne
- preparare i dati per algoritmi basati sulla distanza (es. clustering)

» **Cosa fa il Normalizer**
- considera **ogni colonna numerica selezionata come un insieme indipendente**
- applica una trasformazione ai valori
- produce una versione normalizzata della tabella

⚠ Possono essere normalizzate **solo colonne numeriche**.

Configurazione del nodo

» Per configurare il Normalizer è necessario:

- selezionare le **colonne numeriche** da normalizzare
- scegliere il **metodo di normalizzazione**
  1. **Min–Max Normalization**
  2. **Metodo Z-Score**
  3. **Metodo Decimal Scaling**
- (se necessario) impostare i parametri del metodo scelto

1. Min–Max Normalization

- comprime i valori in un **intervallo definito**
- intervallo predefinito: **[0, 1]**
- personalizzabile tramite i campi **Min** e **Max**

👉 È il metodo più usato nel clustering.

Configurazione del nodo

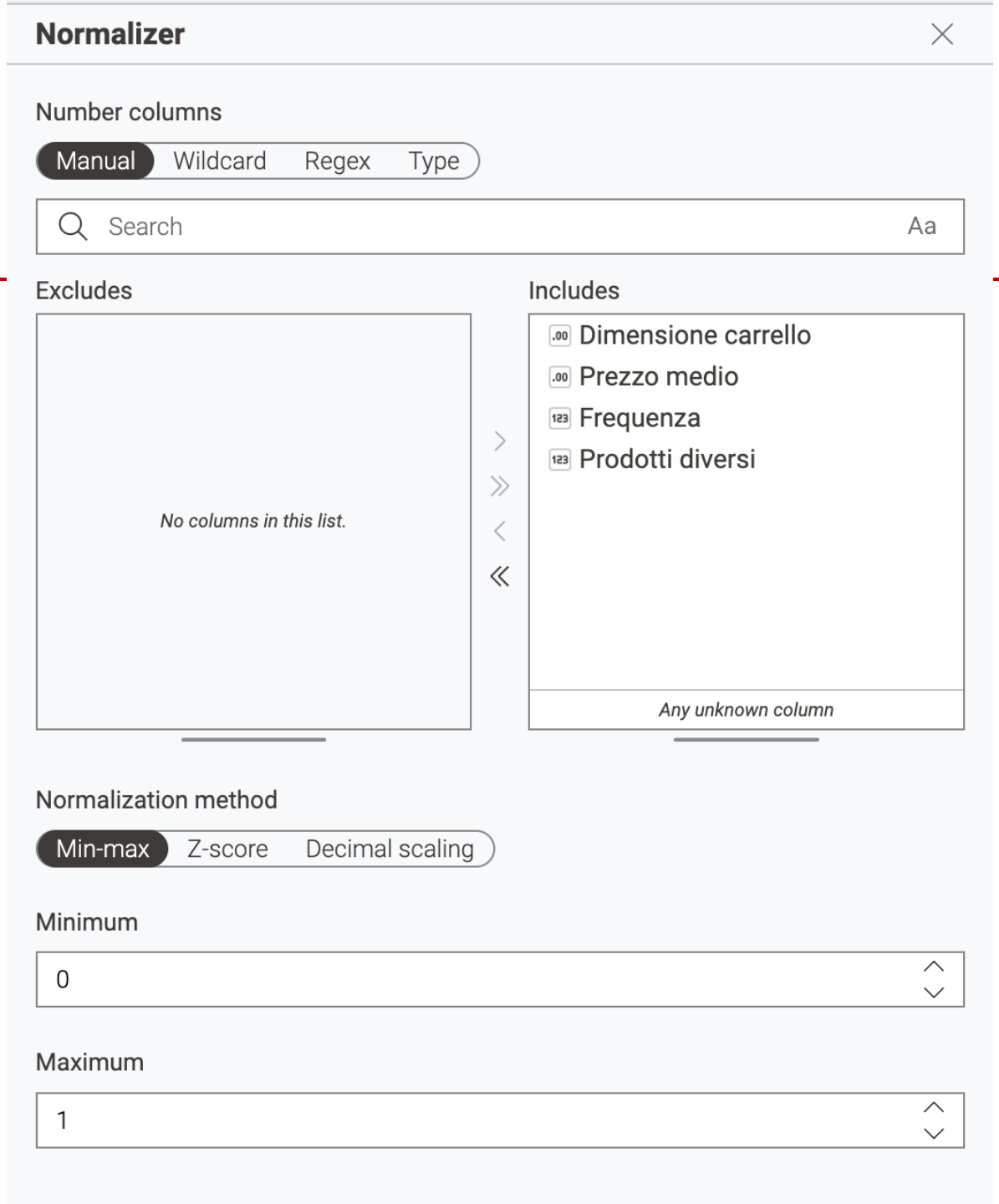
2. Z-Score Normalization
   - standardizzazione statistica
   - assume una distribuzione approssimativamente normale
   - media = 0, varianza = 1

» Il valore normalizzato indica:
   - la distanza dalla media
   - in unità di deviazione standard

» Utile per:
   - individuare **outlier**
   - analisi statistiche

3. Decimal Scaling
   - divide i valori per una potenza di 10 (10, 100, 1000, …)
   - la potenza è scelta automaticamente
   - obiettivo: riportare i valori nel range **[−1, 1]**

» Esempio:
   - valori tra −25 e 170
   - divisione per 1000
   - range normalizzato: [−0.025, 0.170]


Normalizer
Number columns
Manual
Wildcard
Regex
Type
Search
Aa
Excludes
No columns in this list.
Includes
Dimensione carrello
Prezzo medio
Frequenza
Prodotti diversi
Any unknown column
Normalization method
Min-max
Z-score
Decimal scaling
Minimum
0
Maximum
1

Output del Normalizer

» Il nodo produce **due output**:

- **Tabella normalizzata**
  - » le colonne selezionate sono sostituite dai valori normalizzati
- **Modello di normalizzazione**
  - » descrive formalmente le trasformazioni applicate
  - » serve per la **denormalizzazione**

Denormalizer

» È utile per **interpretare correttamente i risultati** del clustering.
  - inverte il processo di normalizzazione
  - riporta i valori nel **range originale**
  - applica in senso inverso i passi del **Normalizer**

» Caratteristiche principali:
  - **non richiede configurazione**
  - richiede in input:
    » la tabella normalizzata
    » il **modello di normalizzazione**
  - restituisce in output:
    » la tabella con valori **denormalizzati**

Scelta dell'algoritmo di clustering

» **Il momento chiave: clustering dei clienti**
  - A questo punto del workflow, i dati sono puliti, le variabili sono aggregate per cliente, le colonne sono normalizzate
  - Siamo pronti a 👉 **creare gruppi omogenei di clienti**

» In KNIME sono disponibili diversi nodi di clustering:
  - **k-means**
  - **Hierarchical Clustering**

» Nel nostro caso scegliamo **k-means** perché:
  - non serve una struttura gerarchica
  - il numero di clienti è elevato
  - vogliamo cluster **operativi e gestibili**

Il parametro k

» Il parametro chiave di k-means è:

k = numero di cluster

- La scelta di k non è solo tecnica, deve essere compatibile con l'uso pratico dei risultati

» **Vincolo di business sulla scelta di k**

- Confrontandoci con il team CRM: il team è ridotto, la gestione delle campagne è impegnativa, non possono essere gestiti troppi gruppi
- Conclusione: 👉 **massimo 5 cluster gestibili**

» **Intervallo per k**

- Dalla discussione emerge che:
  - » k deve essere **almeno 2**
  - » k non deve superare **5**
- Definiamo quindi:
  - » **$k \in [2, 5]$**
- Questo intervallo guida le nostre sperimentazioni

K-means

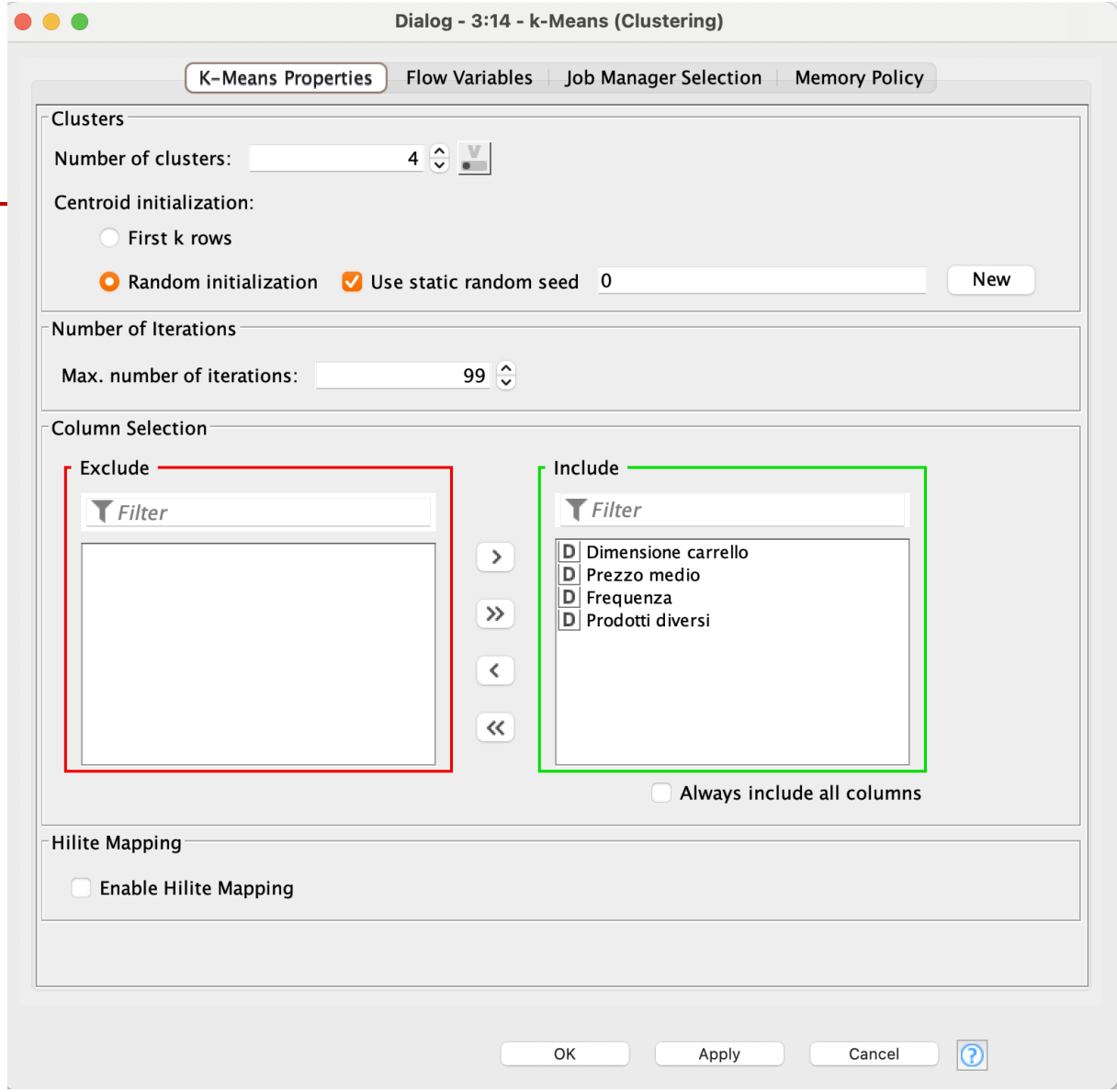
» Il nodo **k-Means** implementa l'algoritmo di clustering k-means
  - raggruppa righe simili
  - assegna ogni riga a un **cluster**
  - lavora su **colonne numeriche**

» **Parametri principali del k-Means**
  - **Number of clusters (k)**
    → numero di gruppi da creare
  - **Max. number of iterations**
    → limite massimo di iterazioni
    (il valore predefinito 99 è in genere sufficiente)
  - **Colonne numeriche da includere**
    → variabili su cui calcolare le distanze

K-means

» Nel clustering:
  - si usano **solo colonne numeriche**
  - colonne identificative (es. *CustomerID*) sono escluse

» Le variabili incluse determinano:
  - la distanza tra i punti
  - la forma dei cluster



Output del nodo k-Means

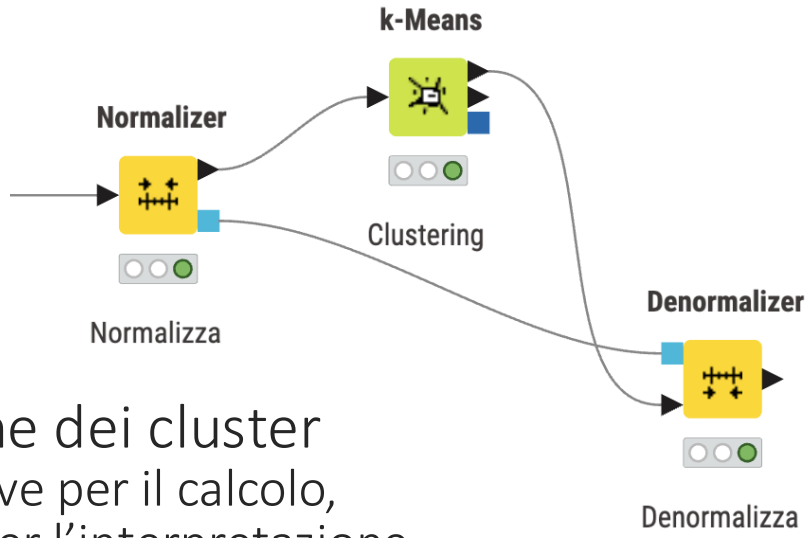
» Il nodo k-Means produce **tre output**:

- **Tabella etichettata**
  - » nuova colonna **Cluster**
  - » valori: cluster_0, cluster_1, …
- **Tabella dei centroidi**
  - » una riga per cluster
  - » valori medi delle colonne
  - » descrizione sintetica dei cluster
- **Modello di clustering**
  - » regole di assegnazione ai cluster
  - » utilizzabile per nuovi dati tramite **Cluster Assigner**

Flusso operativo

» **Normalizzazione e k-Means**
  - ⚠ Il nodo k-Means **non normalizza automaticamente** i dati.
  - Buona pratica: precedere sempre k-Means con il **Normalizer e** garantire che tutte le variabili abbiano lo stesso peso

» Workflow consigliato:



» Analisi e interpretazione dei cluster
  - La normalizzazione serve per il calcolo,
  - la **denormalizzazione** per l'interpretazione.

Denormalizzazione dei risultati

» Dopo il clustering: i valori sono ancora normalizzati e non sono immediatamente interpretabili

» Con il nodo **Denormalizer**:

- si riporta la tabella ai valori originali
- si facilitano le analisi descrittive dei cluster

» Una volta denormalizzati i dati:

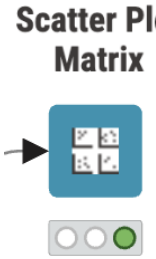
- è possibile confrontare i cluster
- valutare differenze tra centroidi
- decidere il **valore definitivo di k**

» L'interpretazione guida la scelta finale.

Visualizzare i cluster

» Quando il numero di variabili è contenuto (5–6):
  - la **visualizzazione grafica** è il metodo più efficace
  - aiuta a capire cosa distingue i cluster
» Due visualizzazioni molto usate:
  - **Scatter Matrix**
  - **Box Plot**

Scatter Matrix



» Il nodo **Scatter Matrix**:
  - genera una matrice di **scatter plot**
  - ogni grafico rappresenta una coppia di variabili
  - le variabili compaiono sia come righe sia come colonne
» È una visualizzazione combinata e compatta.
» All'interno della matrice:
  - i grafici sulla diagonale mostrano la **distribuzione di una singola variabile**
  - gli altri grafici mostrano la relazione tra **due variabili diverse**
» Serve per individuare:
  - correlazioni
  - separazioni tra cluster
  - pattern strutturali

Scatter Matrix e clustering

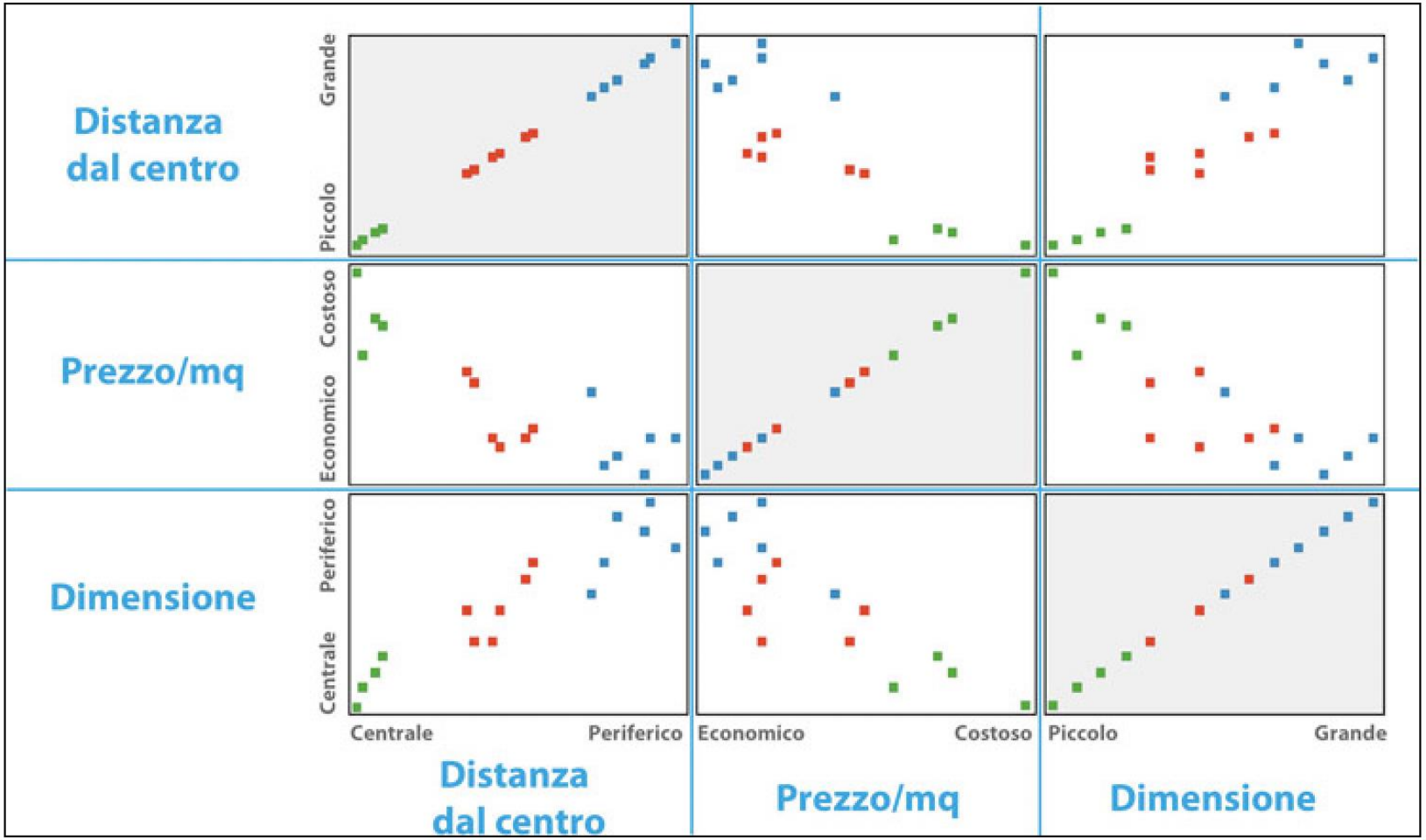
» Colorando i punti in base al cluster:
  - possiamo vedere come i gruppi si distribuiscono
  - capire se i cluster sono realmente distinti
  - interpretare il significato dei gruppi

» La scatter matrix può essere sufficiente, in molti casi, per spiegare il clustering.

» **Esempio di interpretazione** (vedi slide successiva):
  - **cluster verde**: immobili costosi, centrali e piccoli
  - **cluster blu**: immobili periferici, economici e grandi
  - **cluster rosso**: soluzioni intermedie

» I cluster risultano interpretabili grazie alla distribuzione nei grafici.

Scatterix matrix prodotta per visualizzare dimensione, prezzo e distanza al centro di 16 immobili

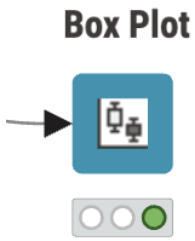


Colorare i punti: Color Manager

» Per colorare i punti nei grafici in base al cluster:
  - si utilizza il nodo **Color Manager**
  - disponibile in **Views → Property**

» Il nodo:
  - associa colori ai valori di una colonna (es. cluster)
  - supporta anche gradienti per colonne numeriche

Box Plot



» Il **Box Plot** è una visualizzazione sintetica che:
  - descrive la distribuzione di una variabile numerica
  - evidenzia valori centrali e dispersione
  - mostra la presenza di **outlier**
» È molto efficace per confronti tra gruppi.
» l nodo **Conditional Box Plot**:
  - genera un box plot per ogni valore di una variabile nominale
  - permette di confrontare la stessa variabile numerica tra cluster diversi
» Se la variabile nominale è il cluster:
  👉 possiamo confrontare i cluster “in parallelo”.

Come leggere un Box Plot

» Elementi principali:
  - **Mediana (Q2)**: valore centrale della distribuzione
  - **Quartile inferiore (Q1)**: 25% dei valori più bassi
  - **Quartile superiore (Q3)**: 25% dei valori più alti
  - **IQR (Q3 – Q1)**: ampiezza della parte centrale dei dati

» La scatola rappresenta il 50% centrale dei valori.

» **Outlier nel Box Plot**
  - I "baffi" indicano il range accettabile
  - I punti oltre i baffi sono **outlier**
  - In KNIME:
    » cerchi → *mild outlier*
    » "X" → *extreme outlier*
  - Gli outlier non sono errori, ma informazioni utili.

Box Plot e clustering

» Confrontando i box plot dei cluster:
- osserviamo se le distribuzioni si sovrappongono
- valutiamo se i cluster sono davvero distinti
- individuiamo gruppi troppo piccoli o anomali

👉 È un controllo qualitativo molto importante.

» **Problema dei cluster sbilanciati**
- Nel caso analizzato:
  - » un cluster contiene ~85% degli elementi
  - » altri cluster contengono pochissimi clienti
- Questo risultato:
  - » è poco utile
  - » non rappresenta gruppi significativi
  - » indica la presenza di **punti eccezionali**

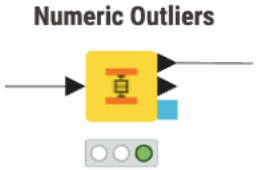
Interpretazione dei cluster sbilanciati

» Quando pochi punti formano cluster a sé:
  - significa che sono **molto distanti** dagli altri
  - rappresentano casi estremi
  - distorcono il clustering

⚠ Questi casi vanno **gestiti separatamente**.

» **Trattamento degli outlier**
  - Strategia adottata:
    » identificare gli outlier con la **distanza interquartile (IQR)**
    » eliminare o gestire i punti troppo estremi
    » ripetere il clustering su dati più omogenei

Numeric Outliers



» Il nodo **Numeric Outliers**:
  - identifica outlier basandosi su Q1, Q3 e IQR
  - usa un moltiplicatore (default 1.5)
  - permette di:
    - » rimuovere outlier
    - » sostituire i valori anomali

» Dopo la rimozione:
  - i cluster risultano più bilanciati
  - diminuisce il numero di cluster "vuoti"
  - la normalizzazione funziona meglio

» Ma attenzione:
  - rimuovere troppi punti può far perdere informazione

Scelta del moltiplicatore IQR

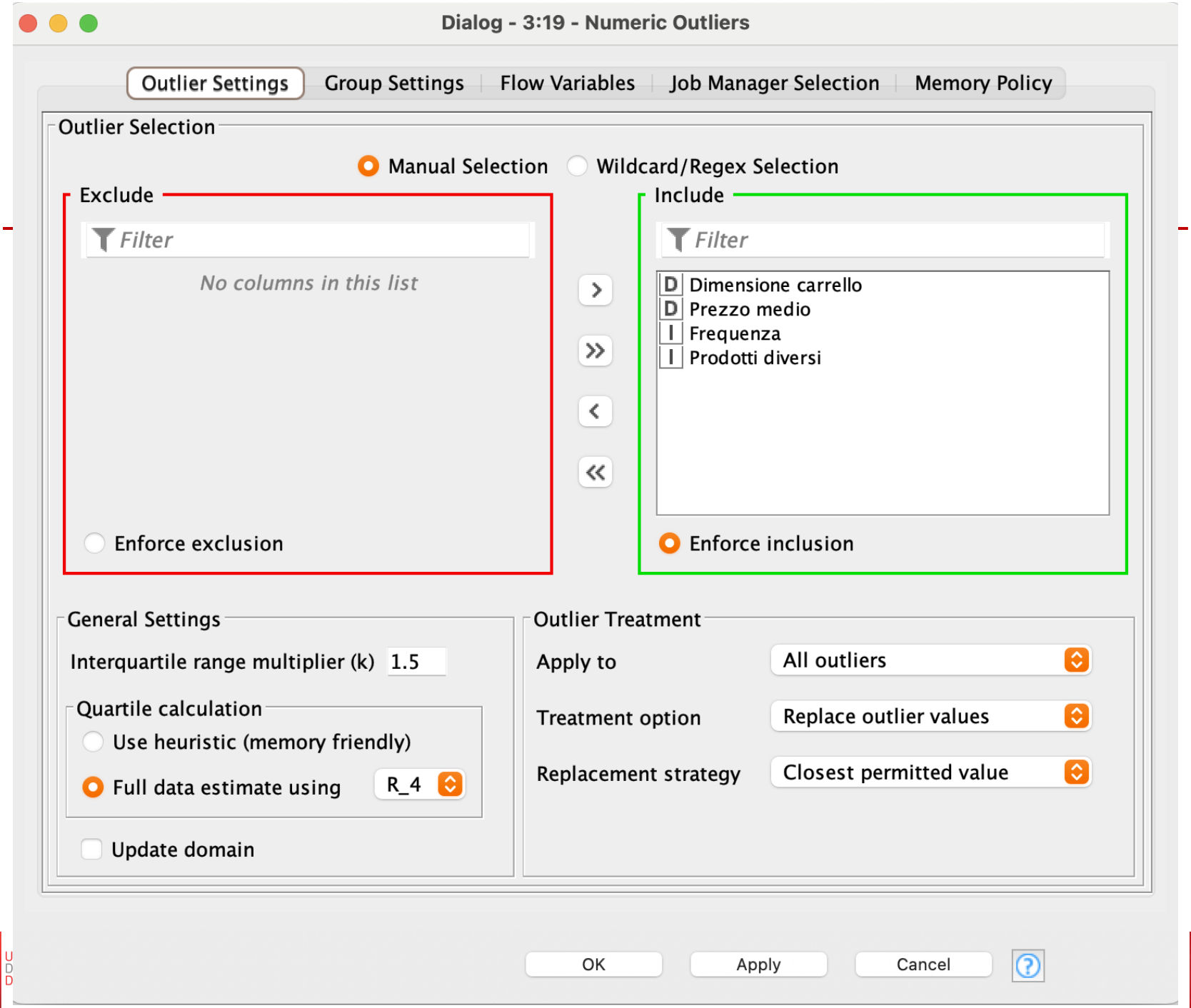
» valore predefinito: **1.5**

» aumentarlo (es. 3.0):

- mantiene i mild outlier
- rimuove solo gli extreme outlier

» Buona pratica:

- iterare
- controllare dimensione dei cluster
- evitare di rimuovere >10% dei dati


Dialog - 3:19 - Numeric Outliers
Outlier Settings
Group Settings
Flow Variables
Job Manager Selection
Memory Policy
Outlier Selection
Manual Selection
Wildcard/Regex Selection
Exclude
Filter
No columns in this list
Enforce exclusion
Include
Filter
D Dimensione carrello
D Prezzo medio
I Frequenza
I Prodotti diversi
Enforce inclusion
General Settings
Interquartile range multiplier (k) 1.5
Quartile calculation
Use heuristic (memory friendly)
Full data estimate using R_4
Update domain
Outlier Treatment
Apply to All outliers
Treatment option Replace outlier values
Replacement strategy Closest permitted value
OK
Apply
Cancel

Ritorno alle visualizzazioni

» Dopo il trattamento degli outlier:
  - Scatter Matrix
  - Conditional Box Plot

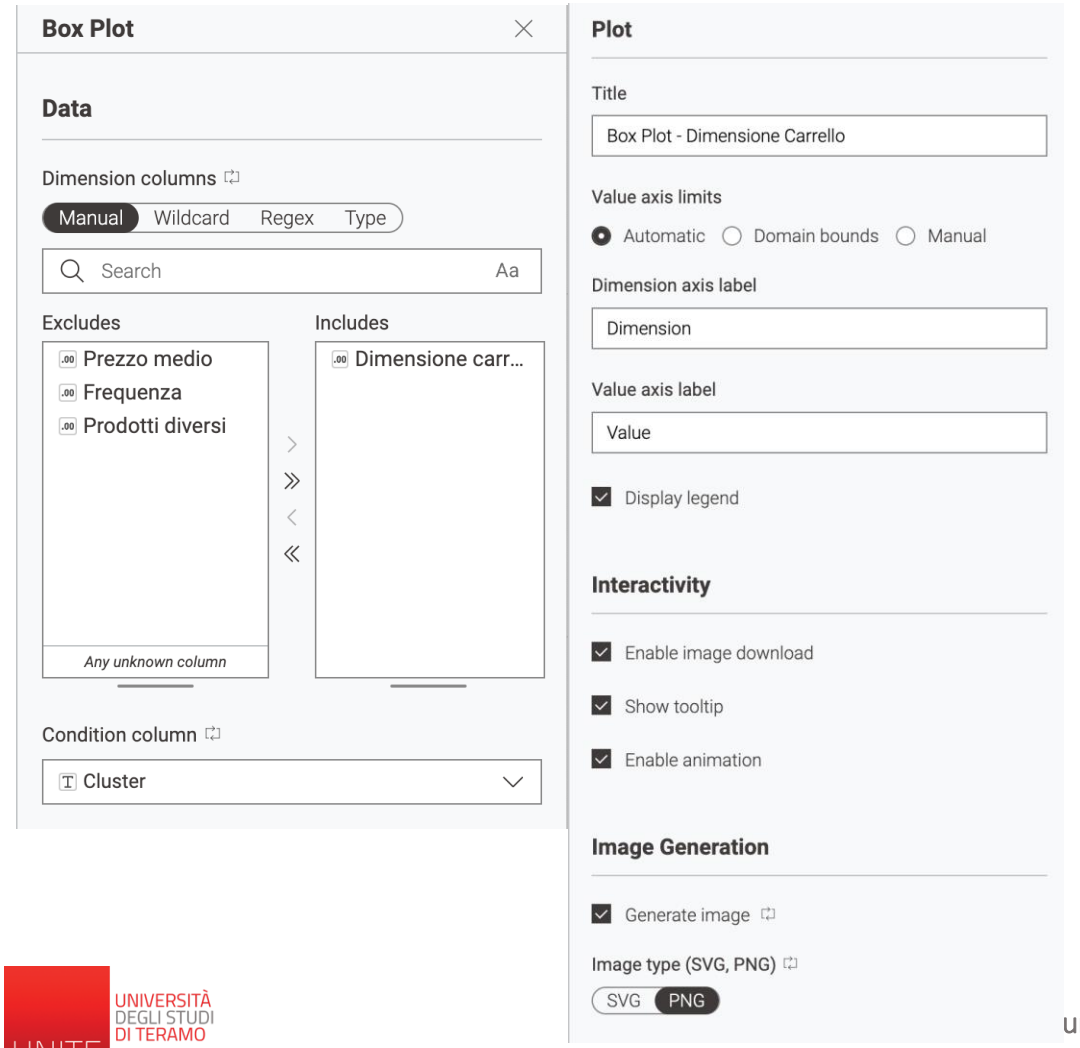
» vengono riutilizzati per:
  - interpretare i cluster finali
  - verificare separazione e coerenza

Confronto tra soluzioni con k diverso

» Testando k = 2, 3, 4, 5:
  - 2–3 cluster → troppo generici
  - 5 cluster → frammentazione inutile
  - **4 cluster** → miglior compromesso

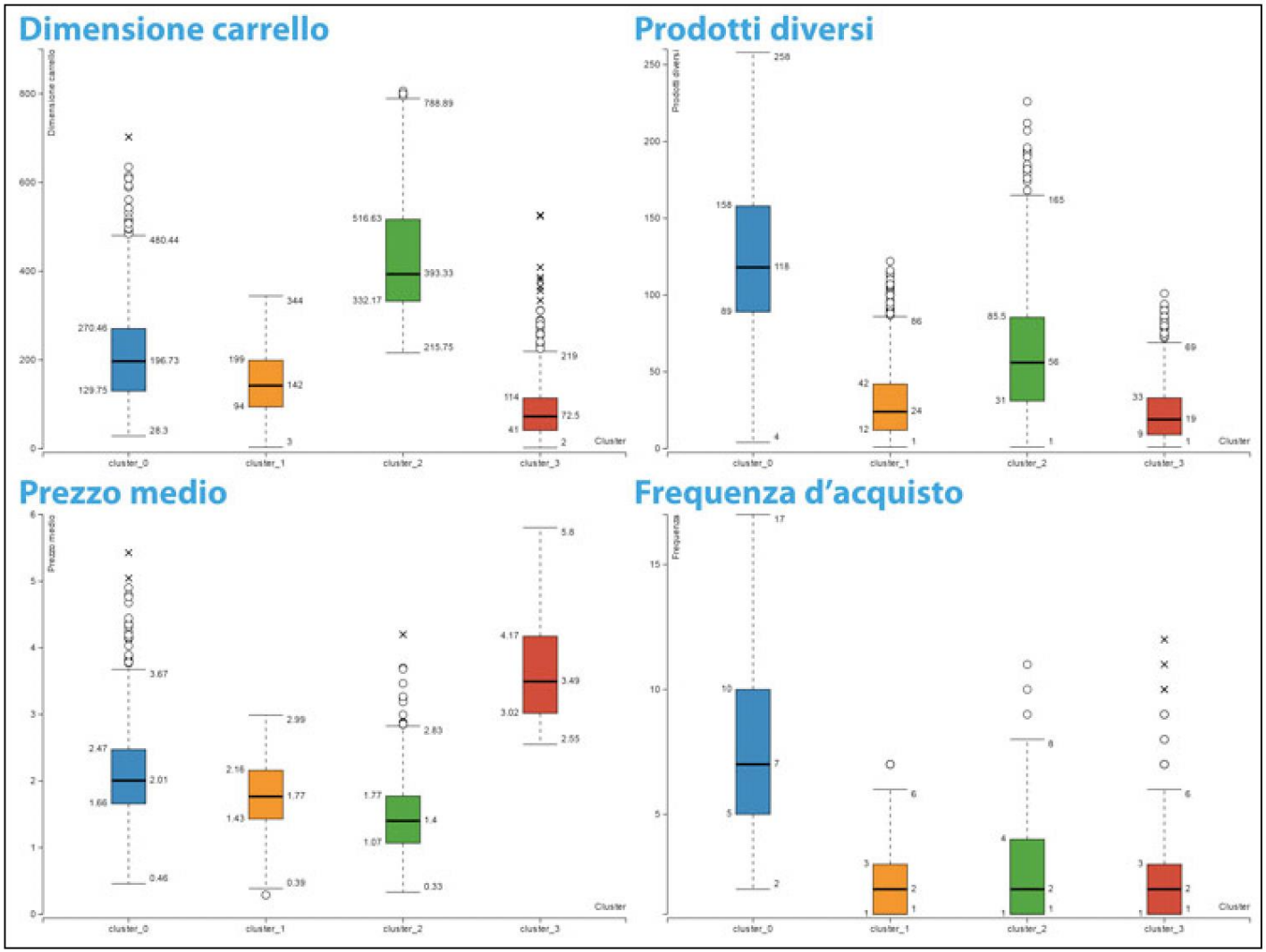
» La scelta di k è guidata dall'interpretabilità.

Configurazione Box Plot

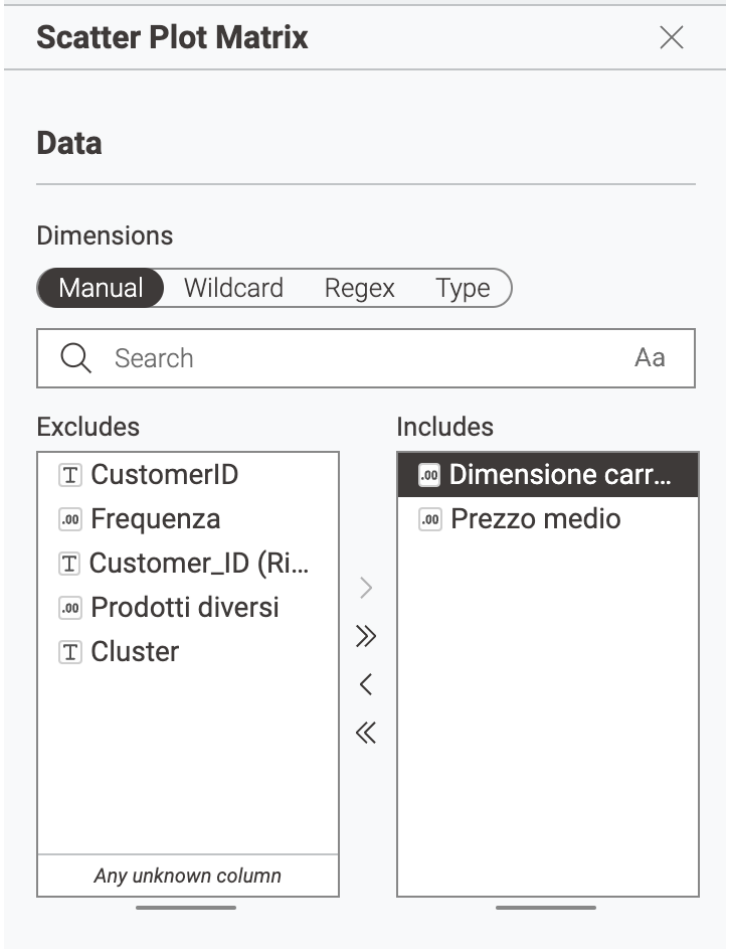


- Creare i box plot condizionali per cluster
- Includere in parallelo le 4 variabili descrittive
  - Dimensione carrello
  - Prezzo medio
  - Frequaneza d'acquisto
  - Prodotti diversi
- Otteniamo 4 diversi diagrammi

Box plot per cluster delle quattro variabili descrittive. Quando le scatole non si sovrappongono in altezza c'è una chiara differenza tra cluster.



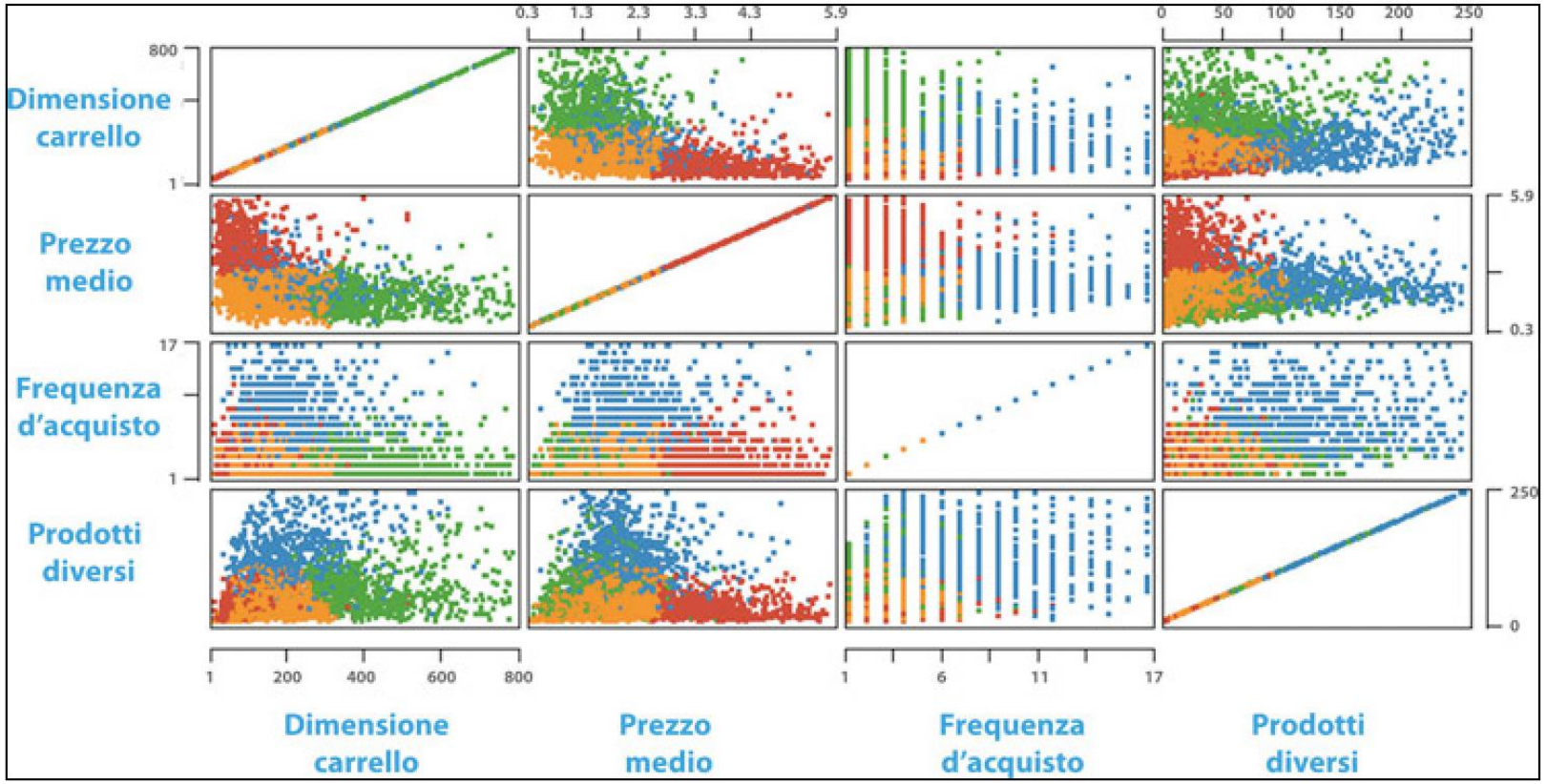
Configurazione Scatter Matrix



» Configurare includendo coppie di variabili descrittive in parallelo

» Otteniamo 16 diversi diagrammi

Scatterix matrix con la distribuzione dei cluster per ogni coppia di variabili desccrittive. I colori dei cluster sono gli stessi della figura con I box plot



Ritorno alle visualizzazioni

» Dopo il trattamento degli outlier:
  - Scatter Matrix
  - Conditional Box Plot

» vengono riutilizzati per:
  - interpretare i cluster finali
  - verificare separazione e coerenza

Confronto tra soluzioni con k diverso

» Testando k = 2, 3, 4, 5:
  - 2–3 cluster → troppo generici
  - 5 cluster → frammentazione inutile
  - **4 cluster** → miglior compromesso

» La scelta di k è guidata dall'interpretabilità.

Descrizione finale dei cluster

» Esempio di profili ottenuti:

- **Cluster 0 (blu)**
  - » Clienti frequenti, molti prodotti diversi → da coinvolgere spesso
- **Cluster 1 (arancione)**
  - » Clienti occasionali → da riattivare con promozioni mirate
- **Cluster 2 (verde)**
  - » Carrelli molto grandi → rivenditori / grossisti
- **Cluster 3 (rosso)**
  - » Pochi prodotti ma costosi → clienti premium

Dal clustering all'azione CRM

» I cluster permettono di:
  - differenziare messaggi
  - personalizzare le offerte
  - semplificare il lavoro del team CRM

» Il valore nasce dall'unione di:
  - analisi automatica
  - interpretazione umana

Workflow finale