

Programmazione in PTC Mathcad

Come scovare trend nascosti in un data-set

<https://www.ptc.com/en/mathcad-software-blog/uncover-hidden-trends-programming-in-mathcad>

Versione italiana a cura di Luca Biasibetti.

Uno dei principali scopi di utilizzo di PTC Mathcad a fini ingegneristici è l'analisi dei dati. Per questo è molto utile utilizzare la programmazione per identificare i trend nascosti nei data-set.

In questo articolo approfondiremo ciò che rende la programmazione uno strumento così potente per l'analisi dei dati.

CICLI PER L'ELABORAZIONE DATI IN UN FOGLIO DI CALCOLO

Spesso quando ci si trova ad analizzare dei dati, si inizia con l'importazione degli stessi da un foglio di lavoro CSV o Excel. Utilizzando i cicli *for* e *while* di PTC Mathcad, si possono valutare rapidamente enormi set di dati. Il potere dei computer consiste nelle operazioni di iterazione, quindi perché non lasciare che PTC Mathcad si occupi dei compiti pesanti e noiosi?

Mathcad possiede anche un ciclo *try-on-error*, quindi i programmi possono continuare ad essere eseguiti quando si verificano errori. Di seguito un esempio di programma che utilizza sia un ciclo *for* che un *try-on-error* per valutare quante volte ogni seed (squadra qualificata) è apparso nelle Final Four negli ultimi 33 anni.

In Figura 1 e Figura 2 troviamo rispettivamente il codice del programma e la rappresentazione grafica dei risultati.

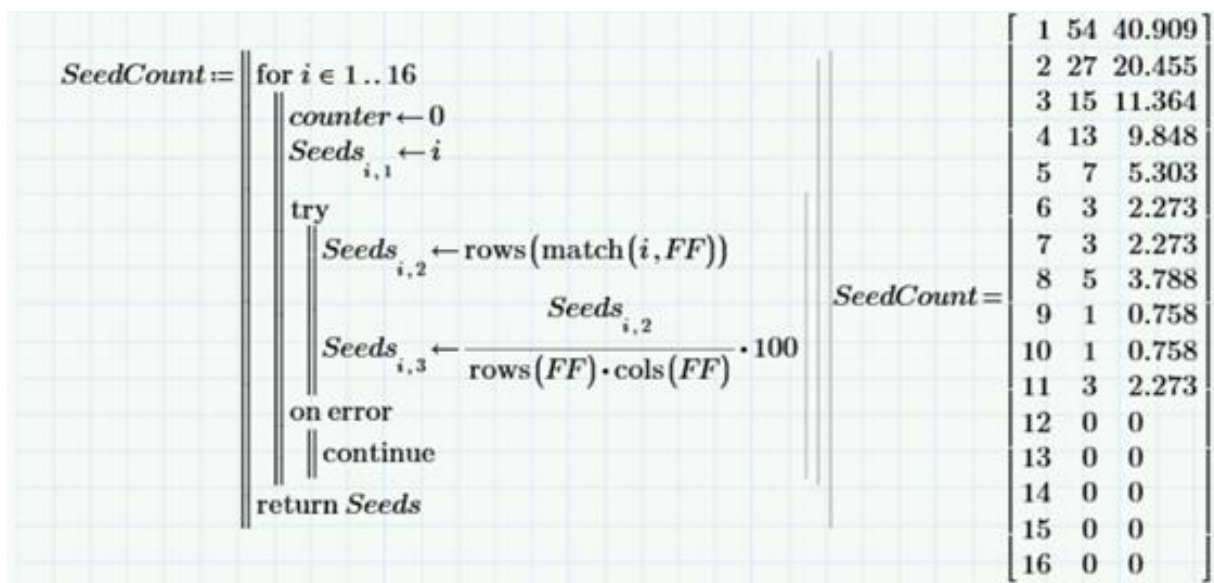


Figura 1: algoritmo del programma per la conta dei seed in PTC Mathcad.

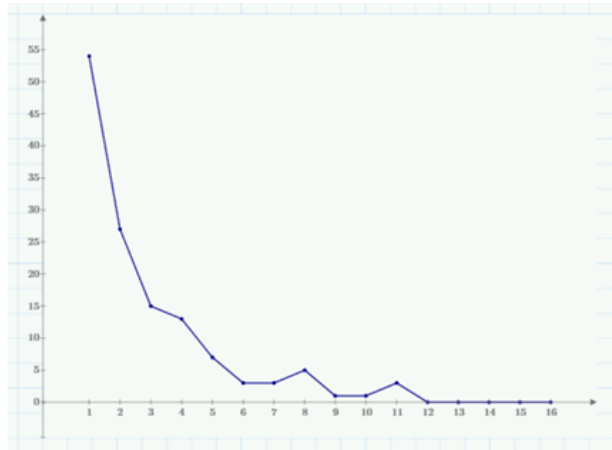


Figura 2: rappresentazione grafica dei risultati.

Si può notare rapidamente che i seed da 12 a 16 non hanno mai fatto le Final Four mentre il numero 11 le ha fatte più volte rispetto a 9 o 10. Un altro semplice programma permette di capire che la media svolta sulla conta dei seed (*SeedAverage*) rispetto all'apparizione nelle Final Four è di circa 2.8 (Figura 3).

```

SeedAverage := || seedtotal ← 0
                || for i ∈ 1..16
                || || seedtotal ← seedtotal + (SeedCounti,1 • SeedCounti,2)
                || Ave ←  $\frac{\text{seedtotal}}{\text{rows}(FF) \cdot \text{cols}(FF)}$ 
                || return Ave
SeedAverage = 2.811

```

Figura 3: calcolo di *SeedAverage* in PTC Mathcad.

OPERATORI CONDIZIONALI

PTC Mathcad ha più di 400 funzioni incorporate ma è anche possibile definirne delle proprie. Spesso però potrebbe essere necessario valutare una situazione composta o che si presenta con diverse condizioni e diramazioni. Con i costrutti condizionali *if-then-else*, *else-if* e *if-if*, i programmi di Mathcad sono anche in grado di prendere delle decisioni articolate.

Analizzando le Final Four NCAA, si può proseguire nell'indagine valutando per quante volte tutti i seed #1 sono arrivati alle Final Four, oppure quante volte ci sono arrivati i seed #1 o #2, #3 e superiori o #4 e superiori e così via. Di seguito il programma (Figura 4) e i risultati in forma grafica e tabulare (Figura 5).

```

TopX(x) := for a ∈ 1..x
  TopTotala,1 ← a
  tempcounter ← 0
  for i ∈ 1..cols(FF)
    if FF1,i ≤ a
      if FF2,i ≤ a
        if FF3,i ≤ a
          if FF4,i ≤ a
            tempcounter ← tempcounter + 1
  TopTotala,2 ← tempcounter
  TopTotala,3 ←  $\frac{\text{tempcounter}}{\text{cols}(FF)}$ 
return TopTotal
  
```

Figura 4: codice per la conta delle volte in cui i seed sono arrivati alle Final Four in PTC Mathcad.

$TopX(1) = [1 \ 1 \ 0.03]$	$TopX(10) =$	$\begin{bmatrix} 1 & 1 & 0.03 \\ 2 & 3 & 0.091 \\ 3 & 11 & 0.333 \\ 4 & 15 & 0.455 \\ 5 & 20 & 0.606 \\ 6 & 23 & 0.697 \\ 7 & 25 & 0.758 \\ 8 & 28 & 0.848 \\ 9 & 29 & 0.879 \\ 10 & 30 & 0.909 \end{bmatrix}$	Legend: Column 1 - Seed Column 2 - # of times Final Four teams have been all that seed or higher. Column 3 - fraction of Final Fours that all teams have been that seed or higher.
$TopX(3) =$			
$TopX(4) =$			

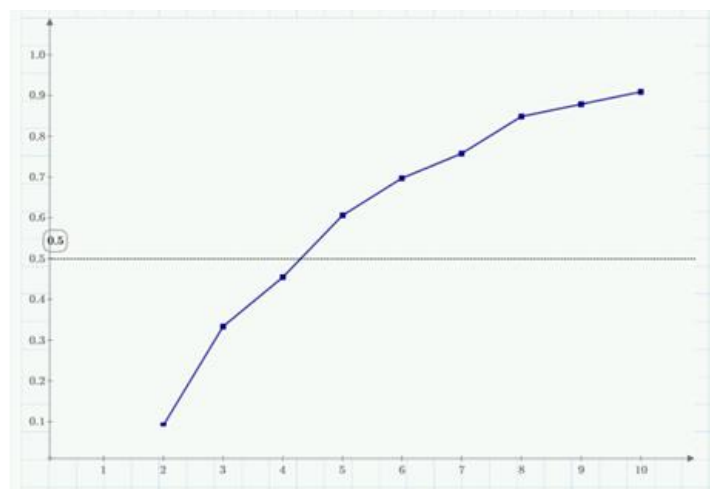


Figura 5: risultati in forma tabulare e grafica relativi all'analisi di Figura 4.

Tutti i seed #1 sono arrivati alle Final Four solo una volta. Per quasi la metà delle volte prese in esame, tutti i seed giunti alle Final Four sono il #4 e successivi, mentre i tre quarti delle volte il #7 o superiori.

DEBUGGING

PTC Mathcad possiede un motore di calcolo che indica gli errori nel programma, in tempo reale, durante la scrittura. Valutando il programma in tempo reale, si possono identificare immediatamente gli errori e quindi correggerli.

Ad esempio, se durante l'analisi dei dati delle Final Four si volesse generare una matrice che riporta quante volte ogni seed appare nelle Final Four per ogni regione, si possono valutare i risultati durante la scrittura del programma.

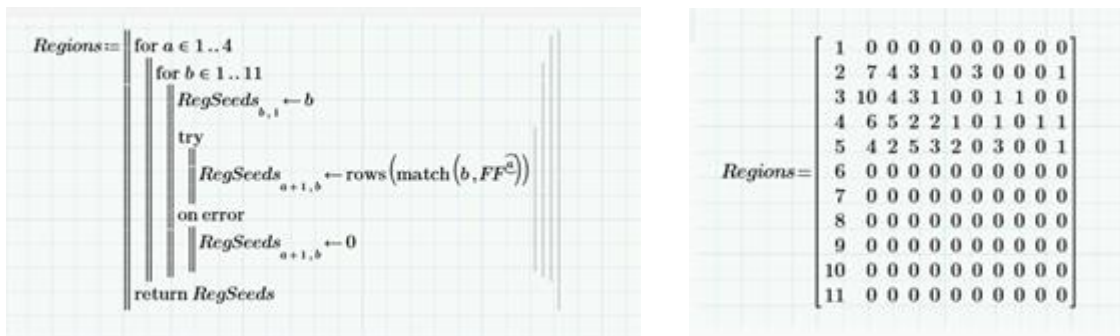


Figura 6: programma e matrice dei risultati.

Nella matrice si può subito notare un numero errato di colonne e un numero troppo elevato di zeri. Da ciò è facile comprendere che i pedici della matrice sono stati invertiti in fase di stesura del codice. Dopo aver applicato le opportune modifiche il programma è stato sottoposto nuovamente a debug.

GENERAZIONE DI NUOVI DATI

Piuttosto che riportare dei singoli valori, quando vengono analizzati dei dati, può essere utile generare vettori e matrici che possono essere riprodotti graficamente. I trend presenti nei dati sono infatti più facilmente riconoscibili quando gli stessi sono rappresentati graficamente. Con la programmazione in PTC Mathcad è possibile generare matrici in modo quasi immediato. Non c'è bisogno di "dichiarare" la matrice o specificare il numero di righe o colonne. La definizione di qualsiasi elemento all'interno di un vettore (o matrice) crea automaticamente un vettore o una matrice della dimensione appropriata.

Perfezionando il programma precedente per calcolare il numero e la percentuale di ciascun seed per ogni regione e generare una matrice dei risultati, otteniamo il codice e il risultato in forma matriciale mostrato in Figura 7.

```

Regions:=
for a ∈ 1..4
  for b ∈ 1..11
    RegSeedsb,1 ← b
    try
      RegSeedsb,a,2 ← rows(match(b, FFa))
      RegSeedsb,a,2+1 ←  $\frac{RegSeeds_{b,a,2}}{cols(FF)}$ 
    on error
      RegSeedsb,a,2 ← 0
      RegSeedsb,a,2+1 ← 0
  return RegSeeds

```

1	14	0.424	13	0.394	14	0.424	13	0.394
2	7	0.212	10	0.303	6	0.182	4	0.121
3	4	0.121	4	0.121	5	0.152	2	0.061
4	3	0.091	3	0.091	2	0.061	5	0.152
5	1	0.03	1	0.03	2	0.061	3	0.091
6	0	0	0	0	1	0.03	2	0.061
7	3	0.091	0	0	0	0	0	0
8	0	0	1	0.03	1	0.03	3	0.091
9	0	0	1	0.03	0	0	0	0
10	0	0	0	0	1	0.03	0	0
11	1	0.03	0	0	1	0.03	1	0.03

Legend:
Col 1: Seed
Col 2-3: East - seeds # and %
Col 4-5: West
Col 6-7: Midwest
Col 8-9: South

Figura 7: programma e matrice dei risultati.

Ovviamente il risultato in forma matriciale non è di semplice interpretazione. Però, se dalla stessa estrapolo un risultato in forma grafica (Figura 8), tutto appare più semplice.

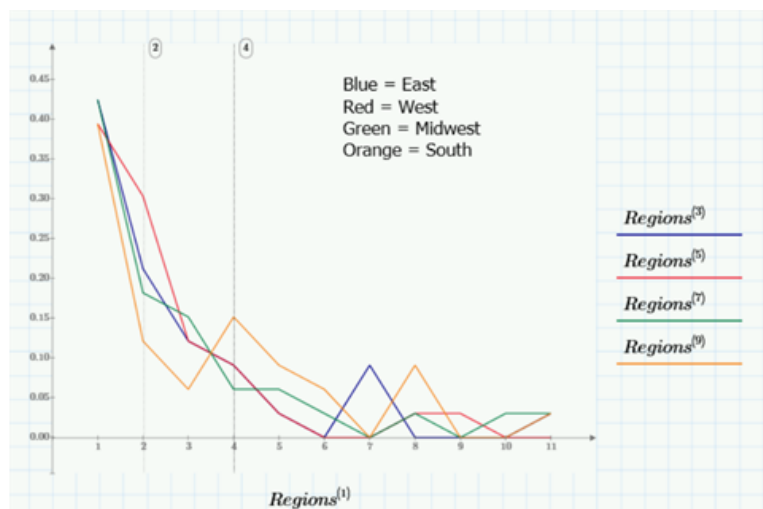


Figura 8: output del programma in forma grafica.

Da qui è possibile dedurre che il seed #1 ha le migliori possibilità di arrivare alla Final Four in tutte le regioni.

Nelle regioni Ovest, se non dovesse avere le migliori possibilità il seed #1, allora le avrà il #2; al Sud invece, è più probabile che vinca il #4 rispetto al #2 o al #3. Ad est, il seed #7 domina come un “jolly”, come a Sud il #8.

CONCLUSIONI

Imparare a programmare in Mathcad è sorprendentemente facile e amplia notevolmente la capacità di raccolta dati per analisi approfondite e studio di trend. La potenza dei cicli, degli operatori condizionali, del debugging e la possibilità di generare matrici può trasformare ed aumentare la produttività nell'analisi dei dati.