

Where Are My Agents? Automating Cohort Building with a Dynamic Cohort Builder

Sherrine Eid, MPH, SAS Institute, Inc., Cary, NC, USA
Mary Dolegowski, MSQM, SAS Institute, Inc., Cary, NC, USA
Sundaresh Sankaran, MS, SAS Institute, Inc., Cary, NC, USA

ABSTRACT

Defining and building patient cohorts involves a mix of automated and manual methods that leverage structured data, unstructured data, and advanced analytics. Producing reliable real-world evidence that informs clinical practice, supports regulatory decisions, and ultimately improves patient outcomes is imperative for regulatory acceptance of integrated evidence packages.

Precisely defining patient cohorts affects study validity and reliability by reducing bias and confounding, ensuring that the evidence generated is robust and valid. This enables meaningful comparisons between treatments and helps in drawing reliable conclusions about effectiveness and safety.

Regulators and payers increasingly rely on RWE to complement randomized clinical trials (RCTs) in decision-making. There is a plethora of examples illustrating how well-defined cohorts support regulatory submissions, label expansions, and post-marketing surveillance.

Identifying distinct patient subgroups can lead to targeted therapies and more personalized treatment strategies. This enables the exploration of treatment effects in real-world subpopulations that might be underrepresented in RCTs.

Standardized and reproducible cohort definitions streamline the research process, making studies more efficient and scalable and facilitates data sharing and collaboration across institutions, which is crucial for large-scale observational studies.

This session will explore the many options researchers have at their disposal to efficiently and accurately define cohorts and accelerate evidence generation in support of drug development and integrated evidence packages.

INTRODUCTION

The process of defining and building patient cohorts in pharma today involves a mix of automated and manual methods that leverage structured data, unstructured data, and advanced analytics. Each approach carries trade-offs between scalability, accuracy, interpretability, and resource requirements. The importance of these methods lies in their ability to produce reliable real-world evidence that informs clinical practice, supports regulatory decisions, and ultimately improves patient outcomes.

Properly defining patient cohorts affects study validity and reliability as accurate definitions reduce bias and confounding, ensuring that the evidence generated is robust and valid. They enable meaningful comparisons between treatments and help in drawing reliable conclusions about effectiveness and safety.

Regulators and payers increasingly rely on RWE to complement randomized clinical trials (RCTs) in decision-making. There is a plethora of examples illustrating how well-defined cohorts support regulatory submissions, label expansions, and post-marketing surveillance.

CUSTOM STEPS – A TOOL FOR BUILDING COHORTS THROUGH A REPEATABLE PROCESS

Cohort identification makes use of multiple techniques, ranging from simple rule-based matches to unsupervised techniques such as clustering. These techniques are codified in the form of SAS programs, and in recent times, also through other open-source interfaces such as Python or R. This implies that the code processed could turn out to be complex and unwieldy for user personas of a non-programming background.

Every technique requires appropriate parameters to output high-quality cohorts. SAS Viya Custom Steps allow users to create repeatable code chunks that are integrated with a graphical user interface (GUI) that capture parameters, process code and make output (cohorts) available for downstream application. Custom steps provide user-friendly interfaces, enhance accessibility for downstream users who may not be familiar with the underlying code, allow for control and defensive programming techniques to capture and handle errors, and also reduce the likelihood of errors related to data selection and manipulation.

In short, use of tools such as SAS Studio Custom Steps improves “first-pass yield”. A metric that originated in the manufacturing domain but increasingly recognized and applied in all processes, first-pass yield measures the efficiency of a process in providing the correct output with minimal rework. This translates into higher productivity for users across the organization.

BUILDING A CUSTOM STEP

We continue with the factory mindset by outlining approaches that help users rapidly build custom steps. Firstly, let's look at the basic architecture of a custom step file. A custom step is quite a simple, serialised object which can be surfaced and viewed as a JSON file. The configuration of a step is defined through two areas:

- Code: executed through a SAS program (which may in turn call other applications and languages as appropriate)
- Components: User interface components that capture input parameters and output specification

Based on their awareness of programming, users can choose from low-code or programmatic interfaces to build custom steps.

Graphical User Interface (GUI)

The low-code interface is accessed in SAS Studio using a Custom Step designer. Upon creating a new custom step, the user notices three tabs: Designer, Prompt UI, and Program. The Designer tab provides a GUI for building the controls that will be presented to the end user. Controls are added using drag-and-drop functionality, and their specific properties can be configured in the Properties panel, as shown in [Figure 1](#), [Figure 2](#), [Figure 3](#), [Figure 4](#), [Figure 5](#), [Figure 6](#), [Figure 7](#), [Figure 8](#), [Figure 9](#), [Figure 10](#), [Figure 11](#), [Figure 12](#), [Figure 13](#), [Figure 14](#), [Figure 15](#), [Figure 16](#), [Figure 17](#), [Figure 18](#), [Figure 19](#), [Figure 20](#), [Figure 21](#), [Figure 22](#), [Figure 23](#), [Figure 24](#), [Figure 25](#), [Figure 26](#), [Figure 27](#), [Figure 28](#), [Figure 29](#), [Figure 30](#), [Figure 31](#), [Figure 32](#), [Figure 33](#), [Figure 34](#), [Figure 35](#), [Figure 36](#), [Figure 37](#), [Figure 38](#), [Figure 39](#), [Figure 40](#), [Figure 41](#), [Figure 42](#), [Figure 43](#), [Figure 44](#), [Figure 45](#), [Figure 46](#), [Figure 47](#), [Figure 48](#), [Figure 49](#), [Figure 50](#), [Figure 51](#), [Figure 52](#), [Figure 53](#), [Figure 54](#), [Figure 55](#), [Figure 56](#), [Figure 57](#), [Figure 58](#), [Figure 59](#), [Figure 60](#), [Figure 61](#), [Figure 62](#), [Figure 63](#), [Figure 64](#), [Figure 65](#), [Figure 66](#), [Figure 67](#), [Figure 68](#), [Figure 69](#), [Figure 70](#), [Figure 71](#), [Figure 72](#), [Figure 73](#), [Figure 74](#), [Figure 75](#), [Figure 76](#), [Figure 77](#), [Figure 78](#), [Figure 79](#), [Figure 80](#), [Figure 81](#), [Figure 82](#), [Figure 83](#), [Figure 84](#), [Figure 85](#), [Figure 86](#), [Figure 87](#), [Figure 88](#), [Figure 89](#), [Figure 90](#), [Figure 91](#), [Figure 92](#), [Figure 93](#), [Figure 94](#), [Figure 95](#), [Figure 96](#), [Figure 97](#), [Figure 98](#), [Figure 99](#), [Figure 100](#), [Figure 101](#), [Figure 102](#), [Figure 103](#), [Figure 104](#), [Figure 105](#), [Figure 106](#), [Figure 107](#), [Figure 108](#), [Figure 109](#), [Figure 110](#), [Figure 111](#), [Figure 112](#), [Figure 113](#), [Figure 114](#), [Figure 115](#), [Figure 116](#), [Figure 117](#), [Figure 118](#), [Figure 119](#), [Figure 120](#), [Figure 121](#), [Figure 122](#), [Figure 123](#), [Figure 124](#), [Figure 125](#), [Figure 126](#), [Figure 127](#), [Figure 128](#), [Figure 129](#), [Figure 130](#), [Figure 131](#), [Figure 132](#), [Figure 133](#), [Figure 134](#), [Figure 135](#), [Figure 136](#), [Figure 137](#), [Figure 138](#), [Figure 139](#), [Figure 140](#), [Figure 141](#), [Figure 142](#), [Figure 143](#), [Figure 144](#), [Figure 145](#), [Figure 146](#), [Figure 147](#), [Figure 148](#), [Figure 149](#), [Figure 150](#), [Figure 151](#), [Figure 152](#), [Figure 153](#), [Figure 154](#), [Figure 155](#), [Figure 156](#), [Figure 157](#), [Figure 158](#), [Figure 159](#), [Figure 160](#), [Figure 161](#), [Figure 162](#), [Figure 163](#), [Figure 164](#), [Figure 165](#), [Figure 166](#), [Figure 167](#), [Figure 168](#), [Figure 169](#), [Figure 170](#), [Figure 171](#), [Figure 172](#), [Figure 173](#), [Figure 174](#), [Figure 175](#), [Figure 176](#), [Figure 177](#), [Figure 178](#), [Figure 179](#), [Figure 180](#), [Figure 181](#), [Figure 182](#), [Figure 183](#), [Figure 184](#), [Figure 185](#), [Figure 186](#), [Figure 187](#), [Figure 188](#), [Figure 189](#), [Figure 190](#), [Figure 191](#), [Figure 192](#), [Figure 193](#), [Figure 194](#), [Figure 195](#), [Figure 196](#), [Figure 197](#), [Figure 198](#), [Figure 199](#), [Figure 200](#), [Figure 201](#), [Figure 202](#), [Figure 203](#), [Figure 204](#), [Figure 205](#), [Figure 206](#), [Figure 207](#), [Figure 208](#), [Figure 209](#), [Figure 210](#), [Figure 211](#), [Figure 212](#), [Figure 213](#), [Figure 214](#), [Figure 215](#), [Figure 216](#), [Figure 217](#), [Figure 218](#), [Figure 219](#), [Figure 220](#), [Figure 221](#), [Figure 222](#), [Figure 223](#), [Figure 224](#), [Figure 225](#), [Figure 226](#), [Figure 227](#), [Figure 228](#), [Figure 229](#), [Figure 230](#), [Figure 231](#), [Figure 232](#), [Figure 233](#), [Figure 234](#), [Figure 235](#), [Figure 236](#), [Figure 237](#), [Figure 238](#), [Figure 239](#), [Figure 240](#), [Figure 241](#), [Figure 242](#), [Figure 243](#), [Figure 244](#), [Figure 245](#), [Figure 246](#), [Figure 247](#), [Figure 248](#), [Figure 249](#), [Figure 250](#), [Figure 251](#), [Figure 252](#), [Figure 253](#), [Figure 254](#), [Figure 255](#), [Figure 256](#), [Figure 257](#), [Figure 258](#), [Figure 259](#), [Figure 260](#), [Figure 261](#), [Figure 262](#), [Figure 263](#), [Figure 264](#), [Figure 265](#), [Figure 266](#), [Figure 267](#), [Figure 268](#), [Figure 269](#), [Figure 270](#), [Figure 271](#), [Figure 272](#), [Figure 273](#), [Figure 274](#), [Figure 275](#), [Figure 276](#), [Figure 277](#), [Figure 278](#), [Figure 279](#), [Figure 280](#), [Figure 281](#), [Figure 282](#), [Figure 283](#), [Figure 284](#), [Figure 285](#), [Figure 286](#), [Figure 287](#), [Figure 288](#), [Figure 289](#), [Figure 290](#), [Figure 291](#), [Figure 292](#), [Figure 293](#), [Figure 294](#), [Figure 295](#), [Figure 296](#), [Figure 297](#), [Figure 298](#), [Figure 299](#), [Figure 300](#), [Figure 301](#), [Figure 302](#), [Figure 303](#), [Figure 304](#), [Figure 305](#), [Figure 306](#), [Figure 307](#), [Figure 308](#), [Figure 309](#), [Figure 310](#), [Figure 311](#), [Figure 312](#), [Figure 313](#), [Figure 314](#), [Figure 315](#), [Figure 316](#), [Figure 317](#), [Figure 318](#), [Figure 319](#), [Figure 320](#), [Figure 321](#), [Figure 322](#), [Figure 323](#), [Figure 324](#), [Figure 325](#), [Figure 326](#), [Figure 327](#), [Figure 328](#), [Figure 329](#), [Figure 330](#), [Figure 331](#), [Figure 332](#), [Figure 333](#), [Figure 334](#), [Figure 335](#), [Figure 336](#), [Figure 337](#), [Figure 338](#), [Figure 339](#), [Figure 340](#), [Figure 341](#), [Figure 342](#), [Figure 343](#), [Figure 344](#), [Figure 345](#), [Figure 346](#), [Figure 347](#), [Figure 348](#), [Figure 349](#), [Figure 350](#), [Figure 351](#), [Figure 352](#), [Figure 353](#), [Figure 354](#), [Figure 355](#), [Figure 356](#), [Figure 357](#), [Figure 358](#), [Figure 359](#), [Figure 360](#), [Figure 361](#), [Figure 362](#), [Figure 363](#), [Figure 364](#), [Figure 365](#), [Figure 366](#), [Figure 367](#), [Figure 368](#), [Figure 369](#), [Figure 370](#), [Figure 371](#), [Figure 372](#), [Figure 373](#), [Figure 374](#), [Figure 375](#), [Figure 376](#), [Figure 377](#), [Figure 378](#), [Figure 379](#), [Figure 380](#), [Figure 381](#), [Figure 382](#), [Figure 383](#), [Figure 384](#), [Figure 385](#), [Figure 386](#), [Figure 387](#), [Figure 388](#), [Figure 389](#), [Figure 390](#), [Figure 391](#), [Figure 392](#), [Figure 393](#), [Figure 394](#), [Figure 395](#), [Figure 396](#), [Figure 397](#), [Figure 398](#), [Figure 399](#), [Figure 400](#), [Figure 401](#), [Figure 402](#), [Figure 403](#), [Figure 404](#), [Figure 405](#), [Figure 406](#), [Figure 407](#), [Figure 408](#), [Figure 409](#), [Figure 410](#), [Figure 411](#), [Figure 412](#), [Figure 413](#), [Figure 414](#), [Figure 415](#), [Figure 416](#), [Figure 417](#), [Figure 418](#), [Figure 419](#), [Figure 420](#), [Figure 421](#), [Figure 422](#), [Figure 423](#), [Figure 424](#), [Figure 425](#), [Figure 426](#), [Figure 427](#), [Figure 428](#), [Figure 429](#), [Figure 430](#), [Figure 431](#), [Figure 432](#), [Figure 433](#), [Figure 434](#), [Figure 435](#), [Figure 436](#), [Figure 437](#), [Figure 438](#), [Figure 439](#), [Figure 440](#), [Figure 441](#), [Figure 442](#), [Figure 443](#), [Figure 444](#), [Figure 445](#), [Figure 446](#), [Figure 447](#), [Figure 448](#), [Figure 449](#), [Figure 450](#), [Figure 451](#), [Figure 452](#), [Figure 453](#), [Figure 454](#), [Figure 455](#), [Figure 456](#), [Figure 457](#), [Figure 458](#), [Figure 459](#), [Figure 460](#), [Figure 461](#), [Figure 462](#), [Figure 463](#), [Figure 464](#), [Figure 465](#), [Figure 466](#), [Figure 467](#), [Figure 468](#), [Figure 469](#), [Figure 470](#), [Figure 471](#), [Figure 472](#), [Figure 473](#), [Figure 474](#), [Figure 475](#), [Figure 476](#), [Figure 477](#), [Figure 478](#), [Figure 479](#), [Figure 480](#), [Figure 481](#), [Figure 482](#), [Figure 483](#), [Figure 484](#), [Figure 485](#), [Figure 486](#), [Figure 487](#), [Figure 488](#), [Figure 489](#), [Figure 490](#), [Figure 491](#), [Figure 492](#), [Figure 493](#), [Figure 494](#), [Figure 495](#), [Figure 496](#), [Figure 497](#), [Figure 498](#), [Figure 499](#), [Figure 500](#), [Figure 501](#), [Figure 502](#), [Figure 503](#), [Figure 504](#), [Figure 505](#), [Figure 506](#), [Figure 507](#), [Figure 508](#), [Figure 509](#), [Figure 510](#), [Figure 511](#), [Figure 512](#), [Figure 513](#), [Figure 514](#), [Figure 515](#), [Figure 516](#), [Figure 517](#), [Figure 518](#), [Figure 519](#), [Figure 520](#), [Figure 521](#), [Figure 522](#), [Figure 523](#), [Figure 524](#), [Figure 525](#), [Figure 526](#), [Figure 527](#), [Figure 528](#), [Figure 529](#), [Figure 530](#), [Figure 531](#), [Figure 532](#), [Figure 533](#), [Figure 534](#), [Figure 535](#), [Figure 536](#), [Figure 537](#), [Figure 538](#), [Figure 539](#), [Figure 540](#), [Figure 541](#), [Figure 542](#), [Figure 543](#), [Figure 544](#), [Figure 545](#), [Figure 546](#), [Figure 547](#), [Figure 548](#), [Figure 549](#), [Figure 550](#), [Figure 551](#), [Figure 552](#), [Figure 553](#), [Figure 554](#), [Figure 555](#), [Figure 556](#), [Figure 557](#), [Figure 558](#), [Figure 559](#), [Figure 560](#), [Figure 561](#), [Figure 562](#), [Figure 563](#), [Figure 564](#), [Figure 565](#), [Figure 566](#), [Figure 567](#), [Figure 568](#), [Figure 569](#), [Figure 570](#), [Figure 571](#), [Figure 572](#), [Figure 573](#), [Figure 574](#), [Figure 575](#), [Figure 576](#), [Figure 577](#), [Figure 578](#), [Figure 579](#), [Figure 580](#), [Figure 581](#), [Figure 582](#), [Figure 583](#), [Figure 584](#), [Figure 585](#), [Figure 586](#), [Figure 587](#), [Figure 588](#), [Figure 589](#), [Figure 590](#), [Figure 591](#), [Figure 592](#), [Figure 593](#), [Figure 594](#), [Figure 595](#), [Figure 596](#), [Figure 597](#), [Figure 598](#), [Figure 599](#), [Figure 600](#), [Figure 601](#), [Figure 602](#), [Figure 603](#), [Figure 604](#), [Figure 605](#), [Figure 606](#), [Figure 607](#), [Figure 608](#), [Figure 609](#), [Figure 610](#), [Figure 611](#), [Figure 612](#), [Figure 613](#), [Figure 614](#), [Figure 615](#), [Figure 616](#), [Figure 617](#), [Figure 618](#), [Figure 619](#), [Figure 620](#), [Figure 621](#), [Figure 622](#), [Figure 623](#), [Figure 624](#), [Figure 625](#), [Figure 626](#), [Figure 627](#), [Figure 628](#), [Figure 629](#), [Figure 630](#), [Figure 631](#), [Figure 632](#), [Figure 633](#), [Figure 634](#), [Figure 635](#), [Figure 636](#), [Figure 637](#), [Figure 638](#), [Figure 639](#), [Figure 640](#), [Figure 641](#), [Figure 642](#), [Figure 643](#), [Figure 644](#), [Figure 645](#), [Figure 646](#), [Figure 647](#), [Figure 648](#), [Figure 649](#), [Figure 650](#), [Figure 651](#), [Figure 652](#), [Figure 653](#), [Figure 654](#), [Figure 655](#), [Figure 656](#), [Figure 657](#), [Figure 658](#), [Figure 659](#), [Figure 660](#), [Figure 661](#), [Figure 662](#), [Figure 663](#), [Figure 664](#), [Figure 665](#), [Figure 666](#), [Figure 667](#), [Figure 668](#), [Figure 669](#), [Figure 670](#), [Figure 671](#), [Figure 672](#), [Figure 673](#), [Figure 674](#), [Figure 675](#), [Figure 676](#), [Figure 677](#), [Figure 678](#), [Figure 679](#), [Figure 680](#), [Figure 681](#), [Figure 682](#), [Figure 683](#), [Figure 684](#), [Figure 685](#), [Figure 686](#), [Figure 687](#), [Figure 688](#), [Figure 689](#), [Figure 690](#), [Figure 691](#), [Figure 692](#), [Figure 693](#), [Figure 694](#), [Figure 695](#), [Figure 696](#), [Figure 697](#), [Figure 698](#), [Figure 699](#), [Figure 700](#), [Figure 701](#), [Figure 702](#), [Figure 703](#), [Figure 704](#), [Figure 705](#), [Figure 706](#), [Figure 707](#), [Figure 708](#), [Figure 709](#), [Figure 710](#), [Figure 711](#), [Figure 712](#), [Figure 713](#), [Figure 714](#), [Figure 715](#), [Figure 716](#), [Figure 717](#), [Figure 718](#), [Figure 719](#), [Figure 720](#), [Figure 721](#), [Figure 722](#), [Figure 723](#), [Figure 724](#), [Figure 725](#), [Figure 726](#), [Figure 727](#), [Figure 728](#), [Figure 729](#), [Figure 730](#), [Figure 731](#), [Figure 732](#), [Figure 733](#), [Figure 734](#), [Figure 735](#), [Figure 736](#), [Figure 737](#), [Figure 738](#), [Figure 739](#), [Figure 740](#), [Figure 741](#), [Figure 742](#), [Figure 743](#), [Figure 744](#), [Figure 745](#), [Figure 746](#), [Figure 747](#), [Figure 748](#), [Figure 749](#), [Figure 750](#), [Figure 751](#), [Figure 752](#), [Figure 753](#), [Figure 754](#), [Figure 755](#), [Figure 756](#), [Figure 757](#), [Figure 758](#), [Figure 759](#), [Figure 760](#), [Figure 761](#), [Figure 762](#), [Figure 763](#), [Figure 764](#), [Figure 765](#), [Figure 766](#), [Figure 767](#), [Figure 768](#), [Figure 769](#), [Figure 770](#), [Figure 771](#), [Figure 772](#), [Figure 773](#), [Figure 774](#), [Figure 775](#), [Figure 776](#), [Figure 777](#), [Figure 778](#), [Figure 779](#), [Figure 780](#), [Figure 781](#), [Figure 782](#), [Figure 783](#), [Figure 784](#), [Figure 785](#), [Figure 786](#), [Figure 787](#), [Figure 788](#), [Figure 789](#), [Figure 790](#), [Figure 791](#), [Figure 792](#), [Figure 793](#), [Figure 794](#), [Figure 795](#), [Figure 796](#), [Figure 797](#), [Figure 798](#), [Figure 799](#), [Figure 800](#), [Figure 801](#), [Figure 802](#), [Figure 803](#), [Figure 804](#), [Figure 805](#), [Figure 806](#), [Figure 807](#), [Figure 808](#), [Figure 809](#), [Figure 810](#), [Figure 811](#), [Figure 812](#), [Figure 813](#), [Figure 814](#), [Figure 815](#), [Figure 816](#), [Figure 817](#), [Figure 818](#), [Figure 819](#), [Figure 820](#), [Figure 821](#), [Figure 822](#), [Figure 823](#), [Figure 824](#), [Figure 825](#), [Figure 826](#), [Figure 827](#), [Figure 828](#), [Figure 829](#), [Figure 830](#), [Figure 831](#), [Figure 832](#), [Figure 833](#), [Figure 834](#), [Figure 835](#), [Figure 836](#), [Figure 837](#), [Figure 838](#), [Figure 839](#), [Figure 840](#), [Figure 841](#), [Figure 842](#), [Figure 843](#), [Figure 844](#), [Figure 845](#), [Figure 846](#), [Figure 847](#), [Figure 848](#), [Figure 849](#), [Figure 850](#), [Figure 851](#), [Figure 852](#), [Figure 853](#), [Figure 854](#), [Figure 855](#), [Figure 856](#), [Figure 857](#), [Figure 858](#), [Figure 859](#), [Figure 860](#), [Figure 861](#), [Figure 862](#), [Figure 863](#), [Figure 864](#), [Figure 865](#), [Figure 866](#), [Figure 867](#), [Figure 868](#), [Figure 869](#), [Figure 870](#), [Figure 871](#), [Figure 872](#), [Figure 873](#), [Figure 874](#), [Figure 875](#), [Figure 876](#), [Figure 877](#), [Figure 878](#), [Figure 879](#), [Figure 880](#), [Figure 881](#), [Figure 882](#), [Figure 883](#), [Figure 884](#), [Figure 885](#), [Figure 886](#), [Figure 887](#), [Figure 888](#), [Figure 889](#), [Figure 890](#), [Figure 891](#), [Figure 892](#), [Figure 893](#), [Figure 894](#), [Figure 895](#), [Figure 896](#), [Figure 897](#), [Figure 898](#), [Figure 899](#), [Figure 900](#), [Figure 901](#), [Figure 902](#), [Figure 903](#), [Figure 904](#), [Figure 905](#), [Figure 906](#), [Figure 907](#), [Figure 908](#), [Figure 909](#), [Figure 910](#), [Figure 911](#), [Figure 912](#), [Figure 913](#), [Figure 914](#), [Figure 915](#), [Figure 916](#), [Figure 917](#), [Figure 918](#), [Figure 919](#), [Figure 920](#), [Figure 921](#), [Figure 922](#), [Figure 923](#), [Figure 924](#), [Figure 925](#), [Figure 926](#), [Figure 927](#), [Figure 928](#), [Figure 929](#), [Figure 930](#), [Figure 931](#), [Figure 932](#), [Figure 933](#), [Figure 934](#), [Figure 935](#), [Figure 936](#), [Figure 937](#), [Figure 938](#), [Figure 939](#), [Figure 940](#), [Figure 941](#), [Figure 942](#), [Figure 943](#), [Figure 944](#), [Figure 945](#), [Figure 946](#), [Figure 947](#), [Figure 948](#), [Figure 949](#), [Figure 950](#), [Figure 951](#), [Figure 952](#), [Figure 953](#), [Figure 954](#), [Figure 955](#), [Figure 956](#), [Figure 957](#), [Figure 958](#), [Figure 959](#), [Figure 960](#), [Figure 961](#), [Figure 962](#), [Figure 963](#), [Figure 964](#), [Figure 965](#), [Figure 966](#), [Figure 967](#), [Figure 968](#), [Figure 969](#), [Figure 970](#), [Figure 971](#), [Figure 972](#), [Figure 973](#), [Figure 974](#), [Figure 975](#), [Figure 976](#), [Figure 977](#), [Figure 978](#), [Figure 979](#), [Figure 980](#), [Figure 981](#), [Figure 982](#), [Figure 983](#), [Figure 984](#), [Figure 985](#), [Figure 986](#), [Figure 9](#)

Display 1: Custom Step – Designer

In Program 1, we see the code associated with the design shown in Display 1. This code uses a collection of macro variables that correspond to various controls defined in the Designer tab. For example, 'select CONSISTENT_MEMBER_ID from &rx_data. where CODE_VALUE in &RX_filter.' references the macro variables &rx_data and &rx_filter. These variables are directly linked to the controls labeled RX Data and RX Filter in Display 1. This relationship between the Designer and Program tabs makes it easy to adapt existing code for use within a Custom Step. By simply converting user-configurable elements into macro variables, developers can integrate a GUI interface without rewriting the entire program.

```

/* SAS templated code goes here */

proc fedsql sessref = mysession;

create table and_Filtered_Member_IDs as
select DISTINCT _RX_Members.CONSISTENT_MEMBER_ID
from
    (select CONSISTENT_MEMBER_ID from &rx_data. where CODE_VALUE in &RX_filter.)
    as _RX_Members
INNER JOIN
    (select CONSISTENT_MEMBER_ID from &code_data. where CODE_VALUE in &code_filter.)
    as _Code_Members
ON _RX_Members.CONSISTENT_MEMBER_ID = _Code_Members.CONSISTENT_MEMBER_ID
;

create table Filtered_RX as
select RX.*
from &rx_data as RX
INNER JOIN
(
    select DISTINCT _RX_Members.CONSISTENT_MEMBER_ID
    from
        (select CONSISTENT_MEMBER_ID from &rx_data. where CODE_VALUE in &RX_filter.)
        as _RX_Members
        INNER JOIN
            (select CONSISTENT_MEMBER_ID from &code_data. where CODE_VALUE in &code_filter.)
            as _Code_Members
            ON _RX_Members.CONSISTENT_MEMBER_ID = _Code_Members.CONSISTENT_MEMBER_ID
    )
as F
ON RX.CONSISTENT_MEMBER_ID = F.CONSISTENT_MEMBER_ID
;

create table Filtered_Service_Code as
select code.*
from &code_data as code
INNER JOIN
(
    select DISTINCT _RX_Members.CONSISTENT_MEMBER_ID
    from
        (select CONSISTENT_MEMBER_ID from &rx_data. where CODE_VALUE in &RX_filter.)
        as _RX_Members
        INNER JOIN
            (select CONSISTENT_MEMBER_ID from &code_data. where CODE_VALUE in &code_filter.)
            as _Code_Members
            ON _RX_Members.CONSISTENT_MEMBER_ID = _Code_Members.CONSISTENT_MEMBER_ID
    )
as F
ON code.CONSISTENT_MEMBER_ID = F.CONSISTENT_MEMBER_ID
;

create table &Step1_output. as
select CONSISTENT_MEMBER_ID
from
    (select CONSISTENT_MEMBER_ID,
        count(CODE_VALUE) as count
    from FILTERED_SERVICE_CODE
    where
        SERVICE_BEGIN_DT >= date%str('%')&start_date.%str('%') AND
        service_end_dt < date%str('%')&end_date.%str('%') AND
        CODE_VALUE in &code_filter.
    group by CONSISTENT_MEMBER_ID) as subset1
where count >= &diag_freq.
;

run;
quit;

```

Program 1: Custom Step - Program

Programmatic Interface

Custom step designers appreciate programmatic interfaces to automate the entire process of building a custom step for cohort build. Advantages to using a programmatic interface over a drag-and-drop interface include:

1. Large-scale automation: The dynamicity of the number of query filters and their variety may require the designer to create multiple custom steps of similar nature, but with different specifications. A programmatic interface helps handle this demanding workload.
2. "Baked-in" parameters: In some cases, the designer may wish to specify hard limits and constraints which are not editable by the end-users and make some other parameters editable. A programmatic interface provides designers with a mechanism to enforce these non-editable values in a governed manner.
3. Standardisation: Programmatic interfaces allow designers convenient ways to enforce standards in terms of wording, choice of control, parameter values, visibility and enablement etc. all of which render the custom step more consumable to downstream users.

An open-source project – `py-sas-studio-custom-steps` (<https://pypi.org/project/py-sas-studio-custom-steps/>) - is an example of a programmatic interface to help create SAS Studio Custom Steps at scale. It is a Python package and can be installed through standard pip package installers as follows:

```
pip install py-sas-studio-custom-steps
```

Reminding ourselves again that a custom step is a simple, serialized object which has a JSON structure, we view this Python package as a set of methods which make it easy to create a custom step file through programmatic methods. The step designer can construct a custom step, attach a SAS program and UI component json to the same, and finally test the same before making it available for consumption.

Here's a simple snippet which shows how easy it is to create a custom step, assuming that the SAS code and the components UI has been finalized.

```
from py_sas_studio_custom_steps import CustomStep

custom_step = CustomStep()

# attach SAS program to custom step
custom_step.attach_sas_program(sas_file="/path/to/a/SAS/program.sas")

# attach UI components to custom step
custom_step.attach_ui(ui_json_file= "/path/to/a/UI/components.json")

# Generate a README (optional)
custom_step.generate_readme(description= "This custom step builds cohorts..")

# save custom step on storage
custom_step.create_custom_step(custom_step_path="/path/to/where/you/save/this_file.step")
```

A crucial final step – **always** test your code and run once before releasing to the "downstream user", even if the downstream user happens to be yourself. Right click on the step where you uploaded it, edit it and only after testing, go ahead and save it in SAS Content. SAS Content is the preferred home for most custom steps (even though it is possible to access and execute custom steps in storage) because it holds information about the custom step metadata within its registry and can form part of future transfer objects, which enables portability and use across environments. Custom steps are basically another form of SAS program at the end of the day and it's important that they go through the full cycle of testing and validation. Many programs have failed in the past due to simple issues like mismatched environmental variables or faulty assumptions and testing your step, irrespective of how they were built, is an ideal way to weed them out.

INTERACTING WITH CUSTOM STEPS

There are two primary ways to interact with Custom Steps once they have been built: as independent steps or as components within a SAS Flow. Running a Custom Step independently allows the specific process defined within it to be executed as a one-off action. However, this mode does not support scheduling. When a Custom Step is included within a Flow (even if the Flow consists of only a single step) it gains additional flexibility. The entire Flow can be executed either as a one-time run or scheduled as a recurring job.

INDEPENDENT

To interact with a Custom Step as a standalone component, right-click the step in question and select Open in a Tab. This action opens the Custom Step in a new tab, as shown in Display 2. In this view, you can see the output of the same program referenced in Display 1, with all associated controls rendered as interactive elements.

For instance, the RX Data control includes a link that lets users browse available libraries and datasets to select the desired input. Similarly, the calendar icon next to the Start Date field opens an interactive calendar for easy date selection.

It's important to note that in Display 2, the Run option is disabled because no output dataset has been selected. This is due to the output field being configured as a required variable within the Designer tab.

▷ Run ■ Cancel | ↺ ↻ Code panel

Patients [from Service Codes data and RX data] with at least 1 diagnosis provided in the Service Code filter or 1 medication provided in the RX Filter between Start Date and End Date.

RX data*

RWE800K.RWE_RX

RX Filter

('00187045302', '00187045301', '27505011130', '27505011

Service Codes data*

RWE800K.RWE_SERVICE_CODES

Service Code Filter

('G9017', 'G9033', 'J0190', 'J0364', 'J0515', 'J1265', 'J2360', 'J7

Start Date

Dec 30, 2007

End Date

Dec 30, 2010

Frequency of Diagnosis

1

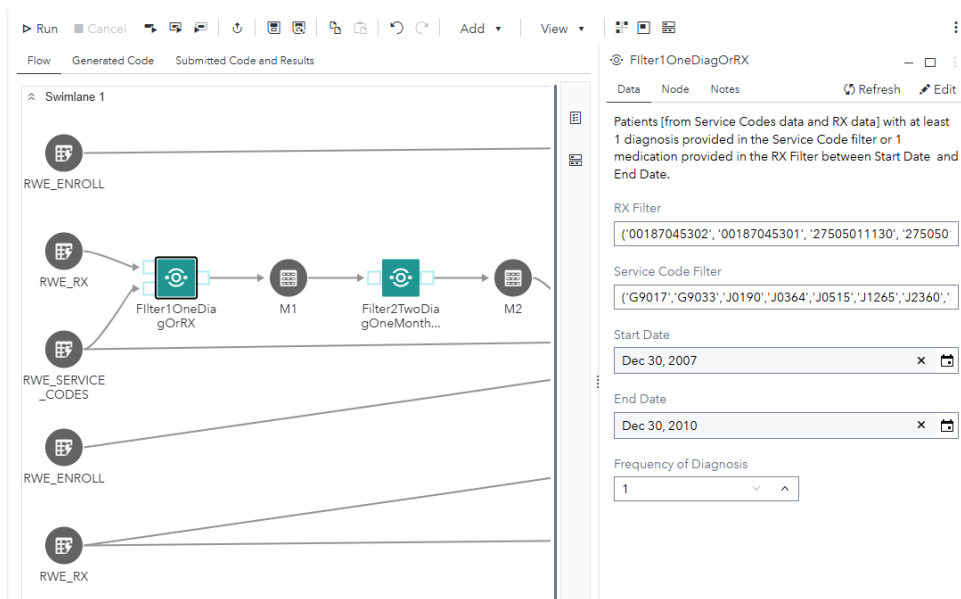
Output*

Display 2: Custom Step - Single Step

PROCESS FLOW

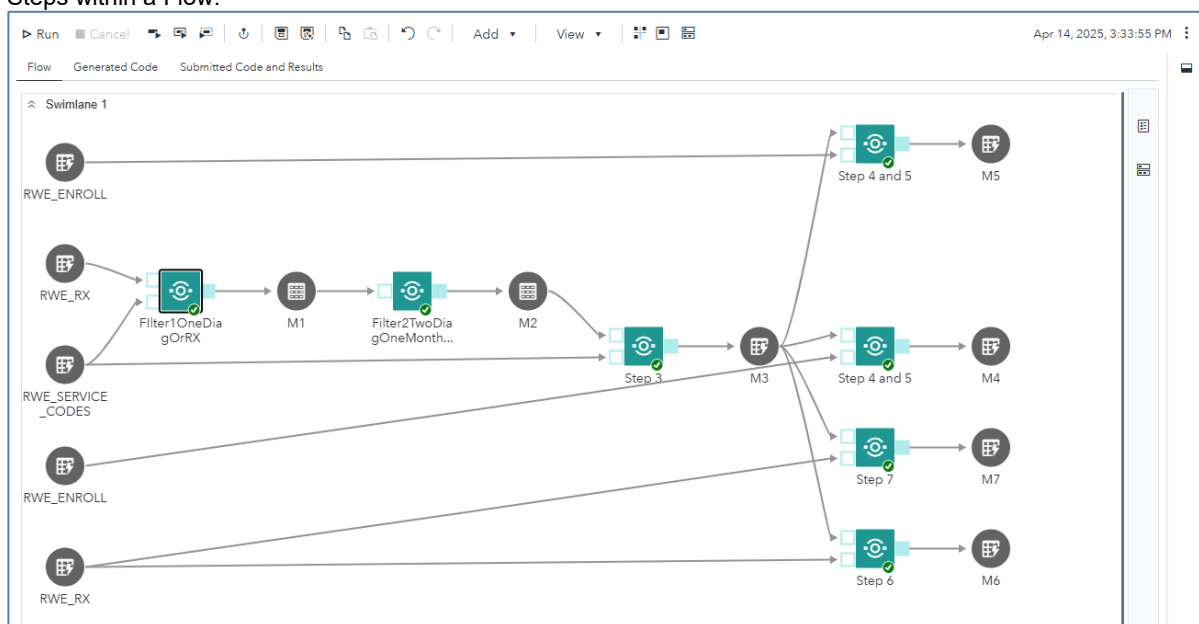
To use a Custom Step within a Flow, the user can either drag and drop the step into the Flow or right-click the step and select Add to Flow. In both cases, the Flow must already be open. The Custom Step can be selected from the user's program list, where it was originally saved, or, if it's stored in a shared location, accessed through the Step domain in SAS Studio under the Shared tab. This second option provides easy access to a centralized catalog of available Custom Steps.

The Custom Step has all the same interactive controls in a Flow as it does when used as a standalone component, as shown in Display 3.



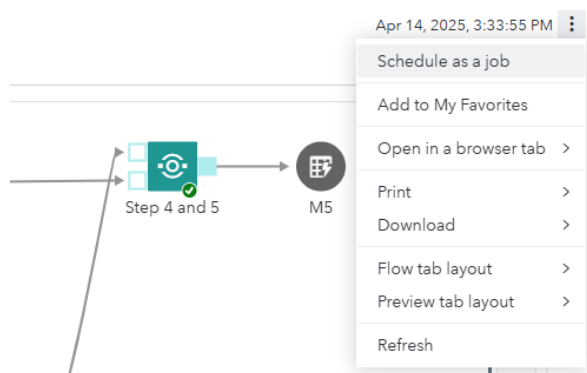
Display 3: Flow – Custom Step Interface

Display 4 shows an example of a completed Flow containing eight Custom Steps. Each step represents a different cohort filtering process. Notably, one of the steps is labeled Step 4 and 5, demonstrating how a single Custom Step can be reused in multiple contexts based on its configuration. This highlights the repeatability and flexibility of Custom Steps within a Flow.



Display 4: Flow

Finally, while the Flow and all of its internal Custom Steps can be run on an ad hoc basis by clicking the Run button in the top-right corner (as shown in Display 4), users also have the option to schedule the Flow as a background job. Display 5 illustrates this process: by selecting the ellipsis menu in the upper-right corner and choosing Schedule as a Job, users can open the New Trigger window. This interface guides them through the options for configuring when and how the Flow should be executed automatically.



Display 5: Flow - Schedule as a Job

WHY NOT HAVE AN 'AGENT' DO THIS FOR YOU?

Due to the popularity (and some weary folks may say almost criminal overuse) of the word “agent” in automated workflows, let us pause to acknowledge that analytics practitioners vary in their definition and understanding of what an agent is. For purpose of simplicity, we adopt a standard definition that establishes an agent as follows:

“An Agent is as Agent does.”

Were we to adopt this definition, the role of the SAS Studio Flow in the above section becomes clearer. The entire flow can be considered as an agent because it has a defined purpose (to build cohorts given some specification), a set of specifications governing the range to which it operates, and a defined output it should strive to provide. This “SAS Studio Flow” agent is also characterized by the human triggers which intervene at certain endpoints. In this case, the human intervenes prior to the flow to provide parameters and after the flow execution to validate output.

The other characterization of the SAS Studio Flow agent is its determinism – the execution path is already well defined, and the flow of execution is modified only on hard conditions that have been prescribed beforehand. We do not suggest that these conditions change – in fact it is desirable for purpose of uniform execution. But we would like to also build some flexibility into the system to be able to understand when to run a series of steps in sequence and determine the sequence at which to run them in real time.

This is where we introduce the concept of tools. Tools are functions made available to an agent for use in a workflow. The choice whether to use a specific tool or not depends on the question at hand. A Large Language Model (LLM), a Generative AI component makes this decision and acts as the orchestrator of tool decisions.

We demonstrate this through SAS Retrieval Agent Manager (RAM), a new offering from SAS that enables agent build, LLM interactions and retrieval against unstructured data sources. Using RAM, the authors identified and created a data dictionary encapsulating the objective, input and output contracts and description of the process of each custom step in a collection. Every custom step can be considered a “tool” that is made available to the agent.

These tools need to be made available from a central place or “service” from where the LLM can access them. Here, we introduce the concept of a Model Context Protocol (MCP) server. An MCP server exposes tools that the agent chooses from, based on guidance from the LLM. MCP servers makes it possible to access local resources and functions within an agentic flow, components that a Large Language Model may not have direct access or ability to execute.

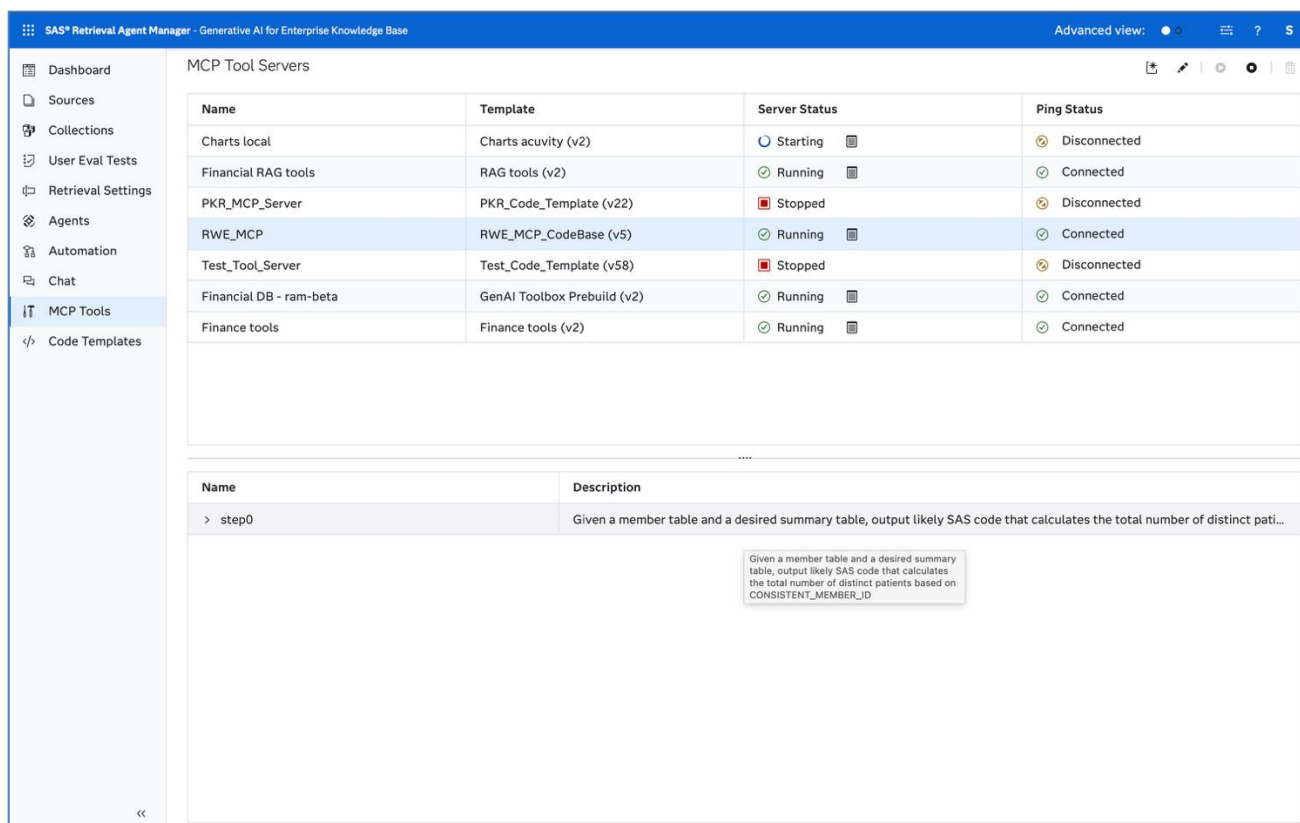


Figure 1: "Starting Block" example of an MCP tool server with a single tool for Step 0 (counting distinct patients)

In this case, the individual cohort builder programs are exposed as tools on a dedicated MCP server which is then available to an Agent within a RAM workflow. The Agent, upon receiving a query, makes use of Retrieval Augmented Generation (RAG) to identify the tools best suited for its purpose and then calls the MCP server to request execution of those identified tools in a decided sequence. Note the immediate benefit over a SAS Studio Flow in terms of not having to specify the execution path in a rigid manner. Further, arguments to provide to each code to populate input parameters are automatically input by the agent.

PROS AND CONS

Every approach has its pros and cons. When considering a decision whether to build cohorts through a SAS Studio Flow or a conversational agent that makes use of an LLM, designers should keep the following in mind.

SAS Studio Flow Agent – Pros and Cons:

Pros:

- Deterministic execution paths
- Programs do not deviate from planned flow
- Lineage and governance due to status as a SAS content "object"

Cons:

- Rigid execution paths
- Changes need to be hard-coded
- SAS / SQL based paradigms for filter – requires conformation to syntax

Conversational Agent (through Retrieval Agent Manager / other platforms) – Pros and Cons:

Pros:

- Flexible execution path
- Agent decides flow on case-by-case basis
- Conversational interface accommodating wide range of filters

Cons:

- Hallucination risks
- Need for stronger, explicit guardrails
- Need to track each execution path separately

ACCESSING YOUR AGENT

Both SAS Studio Flow agents and agents designed using Retrieval Agent Manager can be accessed and triggered via Application Programming Interfaces (APIs). SAS Studio Flows follow the jobs paradigm and are deployed as a job once designed. Designers follow the same method for deploying a SAS Studio Flow as a job as they would for any other job.

Agents designed in Retrieval Agent Manager exist as API endpoints and run on ephemeral resources – i.e. it's possible to deactivate an agent, make it sleep, and then make it operational at various points in time. This optimizes operational costs since resources to run an agent can be accessed only upon need. A typical practice is to design end user web applications (which may differ from client to client) which call RAM, establish credentials and make requests for various operations such as chatting with an agent, or querying tools from the MCP server etc.

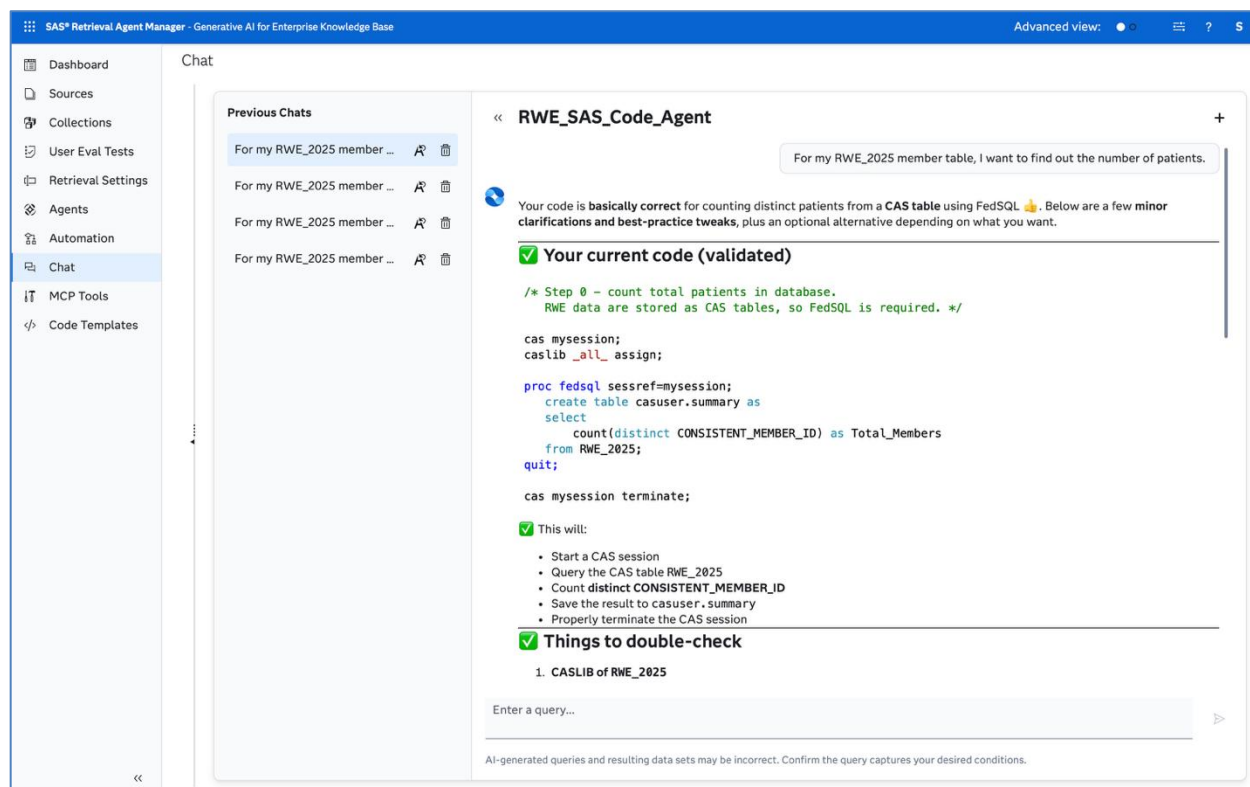


Figure 2: An example of an RWE Agent serving appropriate code

CONCLUSION

There are many options that researchers have at their disposal to efficiently and accurately define cohorts and accelerate evidence generation in support of drug development and integrated evidence packages. We demonstrated the ease of standardizing and reproducing cohort definitions to streamline the research process and make studies more efficient and scalable and facilitate data sharing and collaboration across institutions, which is crucial for large-scale observational studies.

The question is not whether we can build an agent to orchestrate cohort construction and analytical workflows. The

question is how we do so **responsibly, reproducibly, and in alignment with enterprise governance standards.**

The introduction of a conversational, LLM-orchestrated agent—enabled through Retrieval Agent Manager (RAM), Retrieval-Augmented Generation (RAG), and Model Context Protocol (MCP) servers—represents a meaningful architectural evolution. Here, tools (encapsulated analytical programs) are exposed as governed services. The LLM does not replace code; rather, it acts as a reasoning layer that selects, sequences, and parameterizes validated tools at runtime. The intelligence lies not in rewriting analytics logic, but in orchestrating it dynamically.

Agentic flexibility does not eliminate the need for governance. On the contrary, it amplifies it. Hallucination risk, tool misuse, and execution variability require explicit guardrails, contractual input/output definitions, tool validation, execution logging, and audit trails. An enterprise-ready agent is not a free-form chatbot; it is a controlled orchestration engine operating within defined constraints.

For life sciences, regulatory analytics, and other high-stakes domains, this hybrid architecture offers a path forward: intelligent automation without sacrificing compliance; flexibility without abandoning lineage; conversational access without relinquishing control.

Ultimately, an “agent” should not be defined by novelty or trend, but by capability and accountability.

An agent is as an agent does.

And when designed with clear contracts, governed tools, explicit guardrails, and enterprise controls, an agent becomes not a buzzword—but a disciplined extension of analytic expertise.

The future of cohort construction and governed analytics is not purely deterministic, nor purely generative. It is intelligently orchestrated, rigorously governed, and architected for trust.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Sherrine Eid, MPH

Company: SAS Institute, Inc

Address: 100 SAS Campus Drive, Cary, NC

Email: Sherrine.Eid@sas.com

Website: www.sas.com

Author Name: Mary Dolegowski, MQSM

Company: SAS Institute, Inc

Address: 100 SAS Campus Drive, Cary, NC

Email: Mary.Dolegowski@sas.com

Website: www.sas.com

Author Name: Sundaresk Sankaran

Company: SAS Institute, Inc

Address: 100 SAS Campus Drive, Cary, NC

Email: Sundaresk.Sankaran@sas.com

Website: www.sas.com

Brand and product names are trademarks of their respective companies.