

Software evolutions are inevitable, phase your coding (r)evolution to modern Clinical Trial Analytics by using both old/new technique in modern SAS platform.

PHUSE : EU Connect 2021
November 2021



Pritesh Desai

SAS, US Principal Strategic Advisor



Olivier Bouchard

SAS, EMEA Customer Advisory Life Sciences



Customer Perspective | Value Drivers



Business Continuity & Faster Execution

- Foundational Activities Biometrics
- No delays on IA, DSMB, Final Submissions
- Expand Workloads/Faster Execution: Modeling & Simulations; Resampling → no separate environments.



Increased Focus on Data Science

- Data driven Clinical Development
- Model Dev and Ops
- Software as a Medical Device (SaMD)



Openness

- Different backgrounds/skills
- Different Languages: SAS, R, Python, Lua, ...
- Integration capabilities (3rd party apps & other data i.e. EHR, Omics, ...)



Hybrid/Decentralized Trials

- App development (Telehealth, eConsent, Adherence,...)
- Digital Biomarkers / Endpoints (Sensor data)
- Patient (consumer) Centricity



Real World Evidence (RWE)

- RCTs and RWD (RWE) is the future
- Safety, Efficacy & Effectiveness : Submissions
- Use Cases: Trial Feasibility, Synt. Control Arms, Safety Signal Detection, Patient Outcomes, ...



Move to the Cloud

- Scalability & Fault Tolerance
- Innovation – Faster/Easier upgrades
- Compliance with all regulatory requirements incl. Security & Privacy requirements.

VIYA platform

The SAS Platform to embrace Data-Science at scale

9.4 & ™

SAS Viya

SAS® platform

SAS 9

VIYA Platform

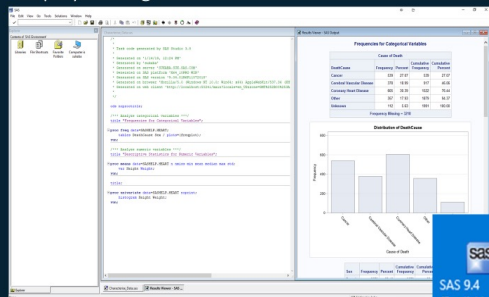
Unifies SAS 9 and SAS “In Memory” – Sustainability & Modernization

SAS VIYA

SAS 9.4

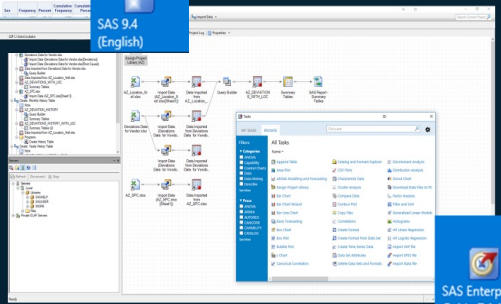
SAS
New In Memory
Capability

SAS Display Manager

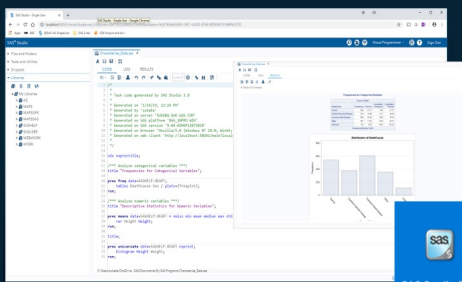


SAS 9.4
(English)

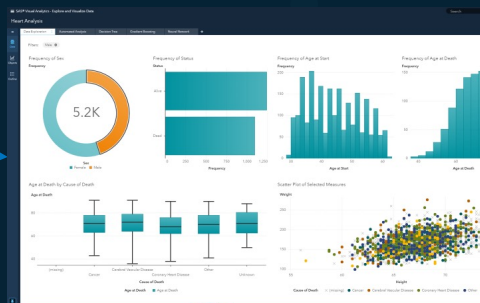
SAS Studio



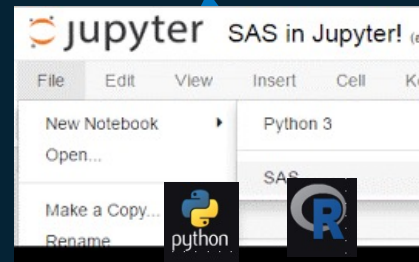
SAS Enterprise
Guide 7.1 (64)



SAS Studio 3.8



New Interfaces



StatusQuo or (R)evolution : why to choose

What's the plan for our presentation

- Can I work the same in a more modern environment?
- How can I benefit from a modern environment with minimum changes
- How a new way of developing can help me to overcome current & future challenges

SAS9 Code in EG vs VIYA

Same Code no Change

* Studio.sas - SAS Enterprise Guide

Fichier Edition Affichage Programme Outils Aide

Page de démarrage... * Studio.sas x

Exécuter Annuler Partager Déboguer Local

Code

```
1 libname stat1 "C:\Users\fraobo\OneDrive - SAS\Documents\Sanofi\VIYA-POC";
2
3 /*st103d01.sas*/ /*Part A*/
4 ods graphics off;
5 proc means data=STAT1.ameshousing3
6     mean var std nway;
7     class Season_Sold Heating_QC;
8     var SalePrice;
9     format Season_Sold Season.;
10    title 'Selected Descriptive Statistics';
11 run;
12
13 /*st103d01.sas*/ /*Part B*/
14 proc sgplot data=STAT1.ameshousing3;
15     vline Season_Sold / group=Heating_QC
16         stat=mean
17         response=SalePrice
18         markers;
19     format Season_Sold season.;
20 run;
21
22 /*st103d01.sas*/ /*Part C*/
23 ods graphics on;
```

Journal Résultats (1)

HTML - Studio.sas

Selected Descriptive Statistics

La procédure MEANS

Variable d'analyse : SalePrice Sale price in dollars					
Season when house sold	Heating quality and condition	N obs	Moyenne	Variance	Ec-type
1	Ex	6	145583.33	1579141667	39738.42
	Fa	3	58100.00	321330000	17925.68
	Gd	10	124330.00	935189000	30580.86
	TA	16	121312.50	1679295833	40979.21
2	Ex	41	153765.24	1129742652	33611.64
	Fa	7	98657.14	452506190	21272.19
	Gd	18	149619.83	1082782633	32905.66
	TA	34	129404.41	767370965	27701.46
3	Ex	45	154279.42	1244833504	35282.20
	Fa	5	128800.00	1332825000	36507.88
	Gd	22	113727.27	1155184935	33988.01
	TA	58	134046.55	1138642444	33743.78
4	Ex	15	163726.93	2436449681	49360.41
	Fa	1	45000.00	.	.
	Gd	8	143812.50	547495536	23398.62
	TA	11	129345.45	462560727	21507.23

SAS9 Code in EG vs VIYA

Same Code no Change

viyawaves.sas.com/SASStudioV/

SAS® Studio - Develop SAS Code

New Options View Open Save All

Start Page Studio.sas x +

Run Cancel Copy to My Snippets Debug

Code

```
1 libname stat1 "/home/fraobo/STUDIO";
2
3 /*st103d01.sas*/ /*Part A*/
4 ods graphics off;
5 proc means data=STAT1.ameshousing3
6     mean var std nway;
7     class Season_Sold Heating_QC;
8     var SalePrice;
9     format Season_Sold Season.;
10    title 'Selected Descriptive Statistics';
11 run;
12
13 /*st103d01.sas*/ /*Part B*/
14 proc sgplot data=STAT1.ameshousing3;
15     vline Season_Sold / group=Heating_QC
16         stat=mean
17         response=SalePrice
18         markers;
19     format Season_Sold season.;
20 run;
21
22 /*st103d01.sas*/ /*Part C*/
23 ods graphics on;
24
25 proc glm data=STAT1.ameshousing3 order=internal;
26     class Season_Sold Heating_QC;
27     model SalePrice = Heating_QC Season_Sold;
```

Log Results

- The Means Procedure
 - Summary statistics
- The SGPlot Procedure
 - The SGPlot Procedure
- The GLM Procedure
 - Data
 - Class Levels
 - Number of ...
 - Analysis of Vari...
 - SalePrice
 - Overall ...
 - Fit Statist...
 - Type I M...
 - Type III ...
 - Interacti...
 - Least Squares ...
 - Season_Sold
 - SalePrice
 - LSMe...

Selected Descriptive Statistics

The MEANS Procedure

Analysis Variable : SalePrice Sale price in dollars

Season when house sold	Heating quality and condition	N Obs	Mean	Variance	Std Dev
1	Ex	6	145583.33	1579141667	39738.42
	Fa	3	58100.00	321330000	17925.68
	Gd	10	124330.00	935189000	30580.86
2	TA	16	121312.50	1679295833	40979.21
	Ex	41	153765.24	1129742652	33611.64
	Fa	7	98657.14	452506190	21272.19
3	Gd	18	149619.83	1082782633	32905.66
	TA	34	129404.41	767370965	27701.46
	Ex	45	154279.42	1244833504	35282.20
4	Fa	5	128800.00	1332825000	36507.88
	Gd	22	113727.27	1155184935	33988.01
	TA	58	134046.55	1138642444	33743.78
4	Ex	15	163726.93	2436449681	49360.41
	Fa	1	45000.00	.	.

SAS9 Code in VIYA

DATA Step

SAS9

```
proc freq data=work.actdata;  
tables age;  
quit;
```

NOTE: There were 204163217 observations read from the data set DM.ACTDATA.

NOTE: The PROCEDURE FREQ printed pages 3-4.

NOTE: PROCEDURE FREQ used (Total process time):

real time	12:19.57
-----------	----------

cpu time	44.37 seconds
----------	---------------

In-Memory

```
proc freqtab data=public.actdata;  
tables age;  
quit;
```

NOTE: The Cloud Analytic Services server processed the request in 15.052263 seconds.

NOTE: The PROCEDURE FREQTAB printed pages 3-4.

NOTE: PROCEDURE FREQTAB used (Total process time):

real time	15.16 seconds
-----------	---------------

cpu time	0.11 seconds
----------	--------------

Proc Mixed code in SAS9 and SAS Viya

DATA Step and Proc(mixed in SAS9 and lmixed in CAS)

SAS9

```
1 data SchoolSample;
2   do SchoolID = 1 to 1000;
3     do nID = 1 to 5;
4       Neighborhood = (SchoolID-1)*15 + nID;
5       bInt = 15*ranuni(1);
6       bTime = 15*ranuni(1);
7       bTime2 = ranuni(1);
8       do sID = 1 to 2;
9         do Time = 1 to 4;
10          Math = bInt + bTime*Time + bTime2*Time*Time + rannor(2);
11          output;
12        end;
13      end;
14    end;
15  end;
16 run;
17
18 proc mixed data=SchoolSample noclprint plots(maxpoints=none);
19   class Neighborhood SchoolID;
20   model Math = Time Time*Time / solution;
21   random int Time Time*Time / sub=Neighborhood(SchoolID) type=un;
22 run;
```

real time	6:57:33
cpu time	6:56:71

In-Memory

```
1 cas mysess;
2 caslib _all_ assign;
3
4 data casuser.SchoolSample;
5   do SchoolID = 1 to 1000;
6     do nID = 1 to 5;
7       Neighborhood = (SchoolID-1)*15 + nID;
8       bInt = 15*ranuni(1);
9       bTime = 15*ranuni(1);
10      bTime2 = ranuni(1);
11      do sID = 1 to 2;
12        do Time = 1 to 4;
13          Math = bInt + bTime*Time + bTime2*Time*Time + rannor(2);
14          output;
15        end;
16      end;
17    end;
18  end;
19 run;
20
21 proc lmixed data=casuser.SchoolSample;
22   class Neighborhood SchoolID;
23   model Math = Time Time*Time / solution;
24   random int Time Time*Time / sub=Neighborhood(SchoolID) type=un;
25 run;
```

real time	1.12 seconds
cpu time	0.13 seconds

Proc Mixed code in SAS9 and SAS Viya

DATA Step and Proc(mixed in SAS9 and Imixed in CAS)

SAS9

```
1 data SchoolSample;
2   do SchoolID = 1 to 1000;
3     do nID = 1 to 50;
4       Neighborhood = (SchoolID-1)*15 + nID;
5       bInt = 15*ranuni(1);
6       bTime = 15*ranuni(1);
7       bTime2 = ranuni(1);
8       do sID = 1 to 2;
9         do Time = 1 to 4;
10          Math = bInt + bTime*Time + bTime2*Time*Time + rannor(2);
11          output;
12        end;
13      end;
14    end;
15  end;
16 run;
17
18 proc mixed data=SchoolSample noclprint plots(maxpoints=none);
19   class Neighborhood SchoolID;
20   model Math = Time Time*Time / solution;
21   random int Time Time*Time / sub=Neighborhood(SchoolID) type=un;
22 run;
```

```
ERROR: Integer overflow on computing amount of memory required. A request to allocate the memory
cannot be honored.
ERROR: The SAS System stopped processing this step because of insufficient memory.
```

In-Memory

```
1 cas mysess;
2 caslib_all_assign;
3
4 data casuser.SchoolSample;
5   do SchoolID = 1 to 1000;
6     do nID = 1 to 50;
7       Neighborhood = (SchoolID-1)*15 + nID;
8       bInt = 15*ranuni(1);
9       bTime = 15*ranuni(1);
10      bTime2 = ranuni(1);
11      do sID = 1 to 2;
12        do Time = 1 to 4;
13          Math = bInt + bTime*Time + bTime2*Time*Time + rannor(2);
14          output;
15        end;
16      end;
17    end;
18  end;
19 run;
20
21 proc imixed data=casuser.SchoolSample;
22   class Neighborhood SchoolID;
23   model Math = Time Time*Time / solution;
24   random int Time Time*Time / sub=Neighborhood(SchoolID) type=un;
25 run;
```

10x

real time	1.12 seconds
cpu time	0.10 seconds

SAS Code In Jupyter Notebook

The screenshot displays a Jupyter Notebook window titled "Untitled1.ipynb" with a menu bar (File, Edit, View, Run, Kernel, Tabs, Settings, Help) and a toolbar. The left sidebar shows a file explorer with a folder named "casuser/readmissions/" containing a file "readmissions_f..." last modified "a month ago".

The notebook contains two code cells. The first cell, labeled "[0]:", contains SAS code that sets up the environment, including loading the "myLib" library, creating a data set "work.readmits", and logging charges. The second cell, labeled "[9]:", contains SAS code that creates a table "work.diagnosis" and prints it. This second cell is highlighted with a green rounded rectangle.

The output of the second cell is displayed below the code, showing the text "The SAS System" and a table with 6 rows and 4 columns: "Obs", "diagnosis", "frequency", and "readmissions".

Obs	diagnosis	frequency	readmissions
1	AMI	8913	2159
2	CABG	5922	1279
3	COPD	10306	2897
4	HF	14005	4239
5	HIP/KNEE	20043	2550
6	PN	15886	3883

The bottom status bar indicates "Mode: Edit", "Ln 1, Col 1", and "Untitled1.ipynb".

Requesting Execution in CAS

SAS 9.4
CAS enabled
Proc's

```
Proc ____;
```

SAS 9.4
CASL

```
Proc CAS;
```

```
result = actionSet.action
```

R

```
Library(swat)  
conn <- swat::cas
```

```
result <- cas.actionSet.action
```

Python

```
import swat  
conn = swat.cas
```

```
result = conn.actionSet.action
```

* Additional APIs for REST, Java, Lua

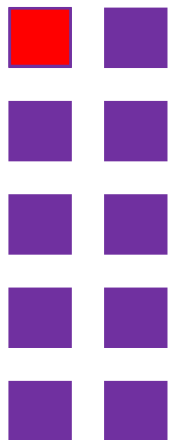
Company Confidential – For Internal Use Only
Copyright © SAS Institute Inc. All rights reserved.

SAS Viya Cloud Analytics Services Execution ?

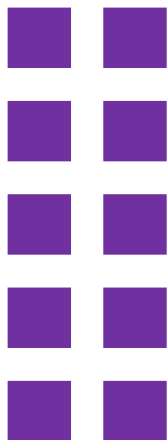
SAS Viya

CAS

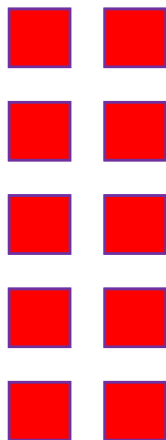
SAS 9.4



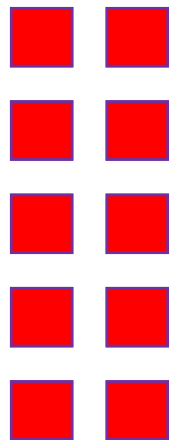
Control



Worker

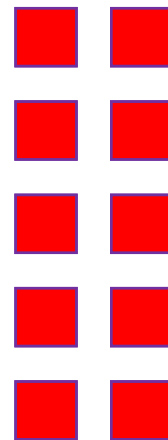


Worker



...

Worker



SAS Viya Cloud Analytics Services Computation

Serial Processing

How This Works for Big Data PROC LOGISTIC and PROC GENMOD

- Your PC, 800,000 observations, 2 servers each with 4 CPUs, compute an $X'X$



Do this many times per optimization step!

How This Works for Big Data PROC LOGISTIC and PROC GENMOD

- Your PC, 800,000 observations, 2 servers each with 4 CPUs, compute an $X'X$



$\Sigma X'X$

Do this many times per optimization step!

How This Works for Big Data PROC LOGISTIC and PROC GENMOD

- Your PC, 800,000 observations, 2 servers each with 4 CPUs, compute an $X'X$



$\Sigma X'X$

Do this many times per optimization step!

Parallel Processing

How This Works for Big Data PROC LOGSELECT and PROC GENSELECT

- Your PC, 800,000 observations, 2 servers each with 4 CPUs, compute an $X'X$

Ready...
Set...



GO!



Company Confidential – For Internal Use Only
Copyright © SAS Institute Inc. All rights reserved.

Notes: <https://www.youtube.com/watch?v=mL4JChKFDDw>



SAS VIYA Architecture

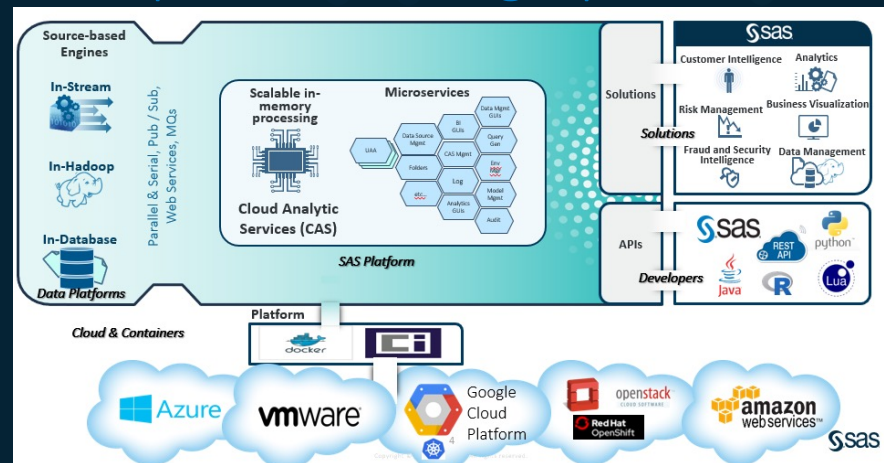
Cloud Native

Result over 10 year of research and development into high-performance in-memory computing

- In-memory
- Massively parallel processing

Cloud Ready

- Compute where you want
- Dynamic, burst capacity
- Containerized deployment through Kubernetes



Clinical Trial Analytics | PROCEDURES THAT USE CAS ACTIONS

PROC APPEND	Adds rows from a CAS table to the end of a SAS data set, and adds rows from a SAS data set to the end of a CAS table.
PROC CONTENTS	Shows the contents of a CAS table and prints the directory of the caslib.
PROC COPY	Copies entire SAS libraries or specific members of the library.
PROC DATASETS	Manages CAS tables.
PROC DELETE	Deletes SAS data sets and CAS tables.
PROC DS2 ¹	Manipulates data with DS2 language statements.
PROC FCMP	Enables you to create, test, and store SAS functions, CALL routines, and subroutines before you use them in other SAS procedures or in DATA steps.
PROC FEDSQL ₂	Manipulates data and performs reporting with FedSQL language statements.
PROC FORMAT	Creates user-defined informats to read data and user-defined formats to display data.
PROC LUA	Enables you to run statements from the Lua programming language within SAS code.
PROC MEANS	Computes descriptive statistics; can produce printed output and output data sets. By default, PROC MEANS produces printed output.
PROC REPORT	Combines features of the PRINT, MEANS, and TABULATE procedures with features of the DATA step in a single report-writing tool that can produce a variety of reports.
PROC SCOREACCEL	Provides an interface to the CAS server for DATA step and DS2 model publishing and scoring.
PROC SORT	Provides the ability to manage duplicate observations in data sets. The SORT procedure uses the CAS deduplicate action to perform the equivalent functionality of the NODUPKEY or NOUNIQUE options of PROC SORT.
PROC SUMMARY	Computes descriptive statistics; can produce a printed report and create an output data set. By default, PROC SUMMARY creates an output data set.
PROC TABULATE	Displays descriptive statistics in tabular format, using some or all of the variables in a data set.
PROC TRANSPOSE	Transforms SAS data sets so that observations become variables and variables become observations.

Clinical Trial Analytics | RECOMMENDATIONS

MIGRATE

Near-Term

- Continue to run 9.4m5 procedures on new environment
- Conduct migration assessment to inventory active SAS workloads and identify potential code challenges
- Pilot effort to migrate initial set of SAS workloads and document best practices
- Update the customer's internal SOPs

ADOPT

Short-Term

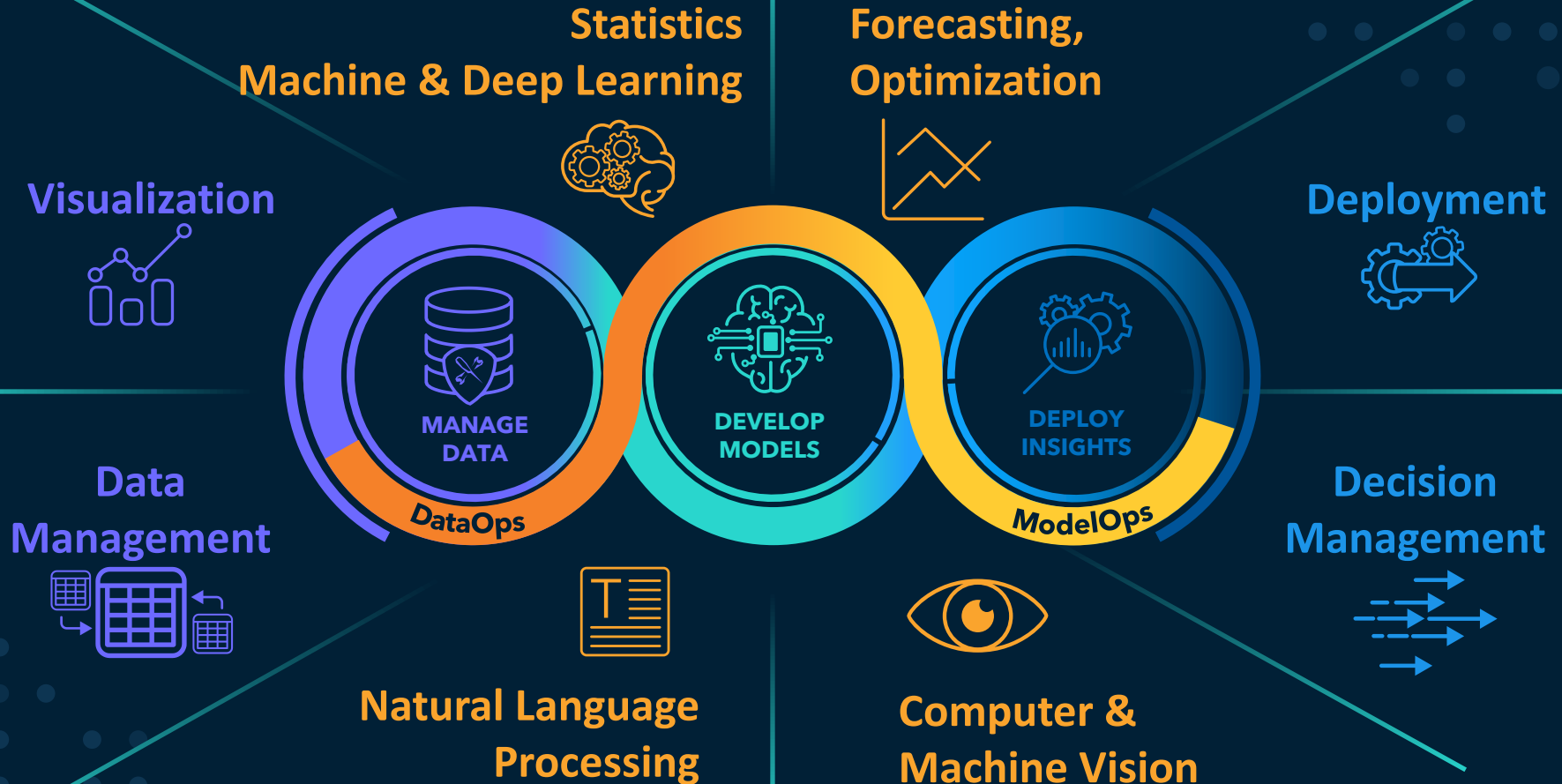
- Identify priority analytic analysis and pilot how to leverage SAS' Cloud Analytics Services and advanced analytics
- Transition batch process to use Viya 3.5 scheduling
- Use new capabilities and performance tracking

EVOLVE

Medium Term

- Use new capabilities to enhance the clinical trial reporting
- Utilize ephemeral containers and/or serverless-functions for running batch SAS jobs
- Provide dedicated analytic containers for model development
- Integrate with External Data

SAS Viya for AI/ML Development



Thank You



Pritesh Desai

Pritesh.desai@sas.com



Olivier Bouchard

Olivier.bouchard@sas.com