

Creating applications with mash-ups

user written functions, stored sql queries, and interfaces

Steve Prust
elderbrook solutions Gmbh

Contents

- What is a mash-up?
- User-written functions
- %GetSQL macro
- ADaM: how to get to “one-step away”
- Your own data warehouse
- Putting it together
- A sample application
- Opportunities and implications
- Conclusion

What is a “mash-up“?

Urban Dictionary

Mash Up · 1. *To take 2 completely different types of music and putting them together.* · 2. *To completely destroy, wreck or mutilate.* · 3. *A cacophony of the senses ...*

mash-up noun

'mash-, əp ▶▶

: something created by combining elements from two or more sources: such as

a : a piece of music created by digitally overlaying an instrumental track with a vocal track from a different recording

b : a movie or video having characters or situations from other sources

c : a Web service or application that integrates data and functionalities from various online sources

Definition: combining things in a new way to get results quicker / better ...

This mash-up: user written fns / ADaM based “data warehouse” / user interface

User written functions in SAS

- Introduced SAS v9 (initially limited but v9.2 made enhancements)
- PROC FCMP to define
- General rule: user-written functions can be used where a SAS function can be used (in data step, in a where statement ...)
- A poor relation to SAS Macro ?
 - It can't encapsulate a procedure ...
 - It only does one thing ...
 - But it allows us to do some clever things ...

%GetSQL macro - 1

- SAS Global Forum 2012 paper “*Use the Full Power of SAS® in Your Function-Style Macros*” Mike Rhoads
 - “*what do you do if you need a macro that can be embedded within a SAS statement, but the underlying task requires the execution of one or more DATA or PROC steps?*”
- Solution of three parts
 - Outer macro – invokes a user written function
 - User written function – a bridge between inner and outer macros
 - Inner macro – does the work – can contain whatever sequence of PROCs and DATA steps needed
- The paper has some examples including %GetSQL ...

%GetSQL macro - 2

- Outer macro

```
%MACRO GetSQL / PARMBUFF;
  %let SYSPBUFF = %superq(SYSPBUFF);
  %let SYSPBUFF = %substr(&SYSPBUFF,2,%LENGTH(&SYSPBUFF) - 2);
  %let SYSPBUFF = %superq(SYSPBUFF);
  %let SYSPBUFF = %sysfunc(quote(&SYSPBUFF));
  %local UNIQUE_INDEX;
  %let UNIQUE_INDEX = &SYSINDEX;
  %sysfunc(GetSQL(&UNIQUE_INDEX,&SYSPBUFF))
%MEND GetSQL;
```

- User written fn

```
proc fcmp outlib=sasuser.mysubs.utility;
  function GetSQL(unique_index_2, query $) $ 32;
  length query query_arg $ 32000 viewname $ 32;
  query_arg = dequote(query);
  rc = run_macro('GetSQL_Inner', unique_index_2, query_arg, viewname);
  if rc = 0 then return(trim(viewname));
  else return('*** ERROR ***');
  endsub;
run;
```

- Inner macro

```
%MACRO GetSQL_Inner;
  %local query;
  %let query = %superq(query_arg);
  %let query = %sysfunc(dequote(&query));
  %let viewname = GetSQL_tmpview_&UNIQUE_INDEX_2;
  proc sql;
  create view &viewname as &query;
  quit;
%MEND GetSQL_Inner;
```

%GetSQL macro – 3 – simple example

```
options cmplib=sasuser.mysubs;  
data example;  
  set  
    %GetSQL (select name, age, weight from sashelp.class where sex = 'M' order by age)  
    %GetSQL (select name, age, height from sashelp.class where sex = 'F' order by name)  
  ;  
run;  
proc print data=example;  
run;
```

The SAS System				
Obs	NAME	AGE	WEIGHT	HEIGHT
1	Thomas	11	85.0	.
2	James	12	83.0	.
3	Robert	12	128.0	.
4	John	12	99.5	.
5	Jeffrey	13	84.0	.
6	Alfred	14	112.5	.
7	Henry	14	102.5	.
8	William	15	112.0	.
9	Ronald	15	133.0	.
10	Philip	16	150.0	.
11	Alice	13	.	56.5
12	Barbara	13	.	65.3
13	Carol	14	.	62.8
14	Jane	12	.	59.8
15	Janet	15	.	62.5
16	Joyce	11	.	51.3
17	Judy	14	.	64.3
18	Louise	12	.	56.3
19	Mary	15	.	66.5

ADaM – “One step away” - 1

- Why did CDISC happen?
 - Easier / standardised data structures
 - “One step away” from analysis
- Good principles to inform what we do?
- Combine ADaM structures and SQL functionality to get to “one step away”

- Complex query: a dataset suitable for AE analysis (freqs, rate ratios, rate differences)

```

proc sql noprint;
create VIEWTABLE: C4_A1 as
select STUDYID, SITEID, TRTA, TRTAN, AGE
1 CDISPILOT01, 701, 0 Placebo, 0, 6
2 CDISPILOT01, 701, 0 Placebo, 0, 6
3 CDISPILOT01, 701, 0 Placebo, 0, 6
(100*nae*pctae / (b.tottrtdur - a.offset)) as rate,
(100*nae/npt) as pctae
from
VIEWTABLE: C4_A1 as
select STUDYID, USUBJID, SUBJID, SITEID, SITEGR1, ARM
1 CDISPILOT01, 01-701-1015, 1015, 701, 701, Placebo
2 CDISPILOT01, 01-701-1023, 1023, 701, 701, Placebo
3 CDISPILOT01, 01-701-1028, 1028, 701, 701, Xanomeline High Dose
VIEWTABLE: C4_A1 as
select distinct usubjid, astdy, (trtedt-trtsdt+1) as trtdurd, (trtedt-trtsdt+1-astdy) as offset,
trtan, trta, aesoc, aebodsys, aeecod, min(sub a.astdy) as min astdy

```

	TRTAN	TRTA	AEBODSYS	AEDECOD	OFFSET	NAE	TOTTRTDUR	TOTY	NPT	TAR	RATE	PCTAE
1	0	Placebo	CARDIAC DISORDERS	ATRIAL FIBRILLATION	0	1	12820	35.099247091	86	35.099247091	2.8490639626	1.1627906977
2	0	Placebo	CARDIAC DISORDERS	ATRIAL HYPERTROPHY	168	1	12820	35.099247091	86	34.639288159	2.8868953525	1.1627906977
3	0	Placebo	CARDIAC DISORDERS	ATRIOVENTRICULAR BLOCK FIRST DEGREE	68	1	12820	35.099247091	86	34.913073238	2.8642565872	1.1627906977
4	0	Placebo	CARDIAC DISORDERS	ATRIOVENTRICULAR BLOCK SECOND DEGREE	6	1	12820	35.099247091	86	35.082819986	2.8503980022	1.1627906977
5	0	Placebo	CARDIAC DISORDERS	BRADYCARDIA	163	1	12820	35.099247091	86	34.652977413	2.8857549182	1.1627906977
6	0	Placebo	CARDIAC DISORDERS	BUNDLE BRANCH BLOCK LEFT	-1	1	12820	35.099247091	86	35.101984942	2.848841744	1.1627906977
7	0	Placebo	CARDIAC DISORDERS	BUNDLE BRANCH BLOCK RIGHT	143	1	12820	35.099247091	86	34.707734428	2.8812021772	1.1627906977
8	0	Placebo	CARDIAC DISORDERS	CARDIAC FAILURE CONGESTIVE	-1	1	12820	35.099247091	86	35.101984942	2.848841744	1.1627906977
9	0	Placebo	CARDIAC DISORDERS	MYOCARDIAL INFARCTION	165	4	12820	35.099247091	86	34.647501711	11.544843935	4.6511627907
10	0	Placebo	CARDIAC DISORDERS	SINUS ARRHYTHMIA	152	1	12820	35.099247091	86	34.683093771	2.8832491317	1.1627906977
11	0	Placebo	CARDIAC DISORDERS	SINUS BRADYCARDIA	66	2	12820	35.099247091	86	34.918548939	5.7276148659	2.3255813953
12	0	Placebo	CARDIAC DISORDERS	SUPRAVENTRICULAR EXTRASYSTOLES	134	1	12820	35.099247091	86	34.732375086	2.8791581271	1.1627906977
13	0	Placebo	CARDIAC DISORDERS	TACHYCARDIA	12	1	12820	35.099247091	86	35.066392882	2.8517332917	1.1627906977
14	0	Placebo	CARDIAC DISORDERS	VENTRICULAR HYPERTROPHY	14	1	12820	35.099247091	86	35.06091718	2.8521786663	1.1627906977

ADaM – “One step away” – 3 – combine with %GetSQL

- and this SQL query can be used in a %GetSQL call ...

```
data example2;
set %GetSQL
( select a.*, b.tottrtdur, b.toty, b.npt,
  (b.tottrtdur - a.offset) / 365.25 as tar,
  (100*nae*365.25/(b.tottrtdur - a.offset)) as rate,
  (100*nae/npt) as pctae
  from
    (
      select trtan, trta, aebodsys, aeDecod, sum(offset) as offset, n(offset) as nae
      from
        (
          select distinct usubjid, astdy, (trtedt-trtsdt+1) as trtdur, (trtedt-trtsdt+1-astdy) as offset, trtan, trta,
            aesoc, aebodsys, aeDecod, min(sub_a.astdy) as min_astdy
          from cdp.adae (where=(TRTEMFL='Y')) as sub_a
          group by usubjid, aesoc, aebodsys, aeDecod
          having astdy = min_astdy
        )
      group by trtan, trta, aebodsys, aeDecod
    ) as a
  inner join
    (
      select sum(trtdur) as tottrtdur, sum(trtdur)/365.25 as toty, n(trtdur) as npt, trt01an, trt01a
      from cdp.adsl
      group by trt01an, trt01a
    ) as b
  on a.trtan = b.trt01an
  order by trtan, trta, aebodsys, aeDecod);
run;
```

- ```
%let _temp_query = select a.*, b.tottrtdur, b.toty, b.npt,
(b.tottrtdur - a.offset) / 365.25 as tar,
(100*nae*365.25/(b.tottrtdur - a.offset)) as rate,
(100*nae/npt) as pctae from (select trtan, trta ...
```

# The SAS System

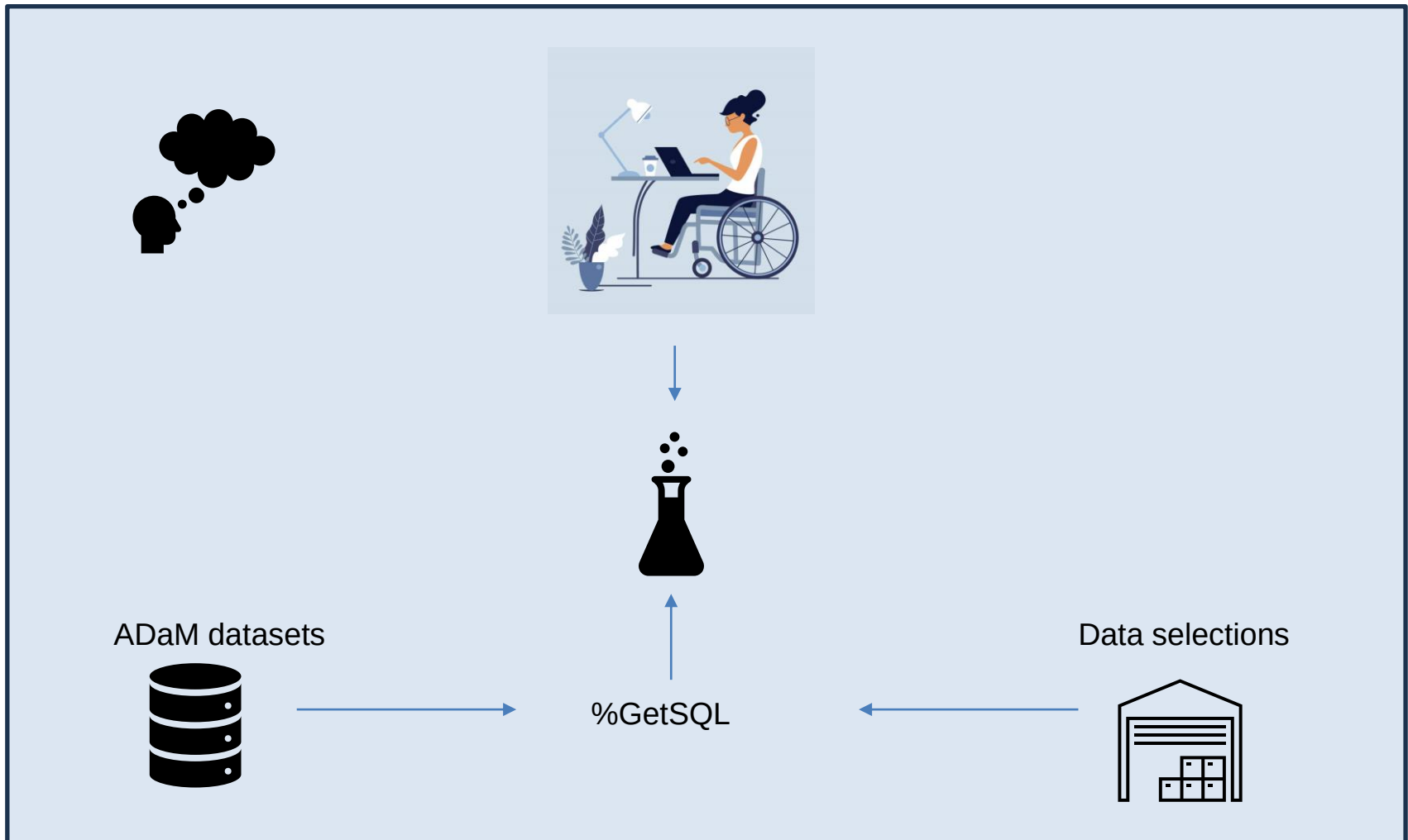
| O<br>b<br>s | T<br>R<br>T<br>A<br>N | T<br>R<br>T<br>A | A<br>D<br>D<br>I<br>T<br>I<br>O<br>N<br>A<br>L |      |     |              |        |           | A<br>D<br>D<br>I<br>T<br>I<br>O<br>N<br>A<br>L | O<br>F<br>F<br>S<br>E<br>T |     |
|-------------|-----------------------|------------------|------------------------------------------------|------|-----|--------------|--------|-----------|------------------------------------------------|----------------------------|-----|
|             |                       |                  |                                                |      |     |              |        |           |                                                |                            |     |
| 326         | 81                    | Xanomeline       | High Dose                                      | SKIN | AND | SUBCUTANEOUS | TISSUE | DISORDERS | PRURITUS                                       | GENERALISED                | 146 |
| 327         | 54                    | Xanomeline       | Low Dose                                       | SKIN | AND | SUBCUTANEOUS | TISSUE | DISORDERS | RASH                                           |                            | 800 |
| 328         | 0                     | Placebo          |                                                | SKIN | AND | SUBCUTANEOUS | TISSUE | DISORDERS | RASH                                           |                            | 212 |
| 329         | 81                    | Xanomeline       | High Dose                                      | SKIN | AND | SUBCUTANEOUS | TISSUE | DISORDERS | RASH                                           |                            | 466 |
| 330         | 54                    | Xanomeline       | Low Dose                                       | SKIN | AND | SUBCUTANEOUS | TISSUE | DISORDERS | RASH                                           | ERYTHEMATOUS               | 4   |
| 331         | 81                    | Xanomeline       | High Dose                                      | SKIN | AND | SUBCUTANEOUS | TISSUE | DISORDERS | RASH                                           | MACULO-PAPULAR             | 182 |

## Data Warehouse

---

- Definitions of data selections (statistical analysis plans and so on)
- Each data selection can be defined as an SQL query
- Store these data selections with relevant attributes (name, ordering, key ...)
- Result ... a data warehouse
- Can access data warehouse and combine with tools such as %GetSQL to have analysis ready data
- ... where does this lead ?

## Putting it together - 1



## Putting it together - 2

---

- Storing queries ...
  - SQL code could be stored in any suitable container
  - JSON is one example (and is likely to become more common) ...
  - This example has a two level structure – table of queries and the queries themselves

```
{
 "SASJSONExport": "1.0 PRETTY",
 "SASTableData+QUERYIDS": [
 {
 "QNAME": "Example 1",
 "QDESC": "Further example of a query of eGFR endpoint",
 "QAVAR": "chg",
 "QBVAR": "trtn, visit"
```

```
"SASJSONExport": "1.0 PRETTY",
"SASTableData+QUERYDET": [
 {
 "QNAME": "Example 1",
 "CODELINE": "select a.*, b.avisitn as visit label=\"Visit\" format=_avisitn., b.aval, b.base, b.chg, b.param, b.paramcd from",
 "LINEN": 1
```

## Putting it together - 3

---

- Application requirements
  - Access data via stored queries in data warehouse
  - Additional functionality (for example subsets / subgroups)
  - Process data and produce results
    - Local solutions (for example macros and/or standard methods)
- What tools are there ?
  - SAS – SAS A/F (well-integrated, flexible ....)
  - Other platforms

## Putting it together - 4

---

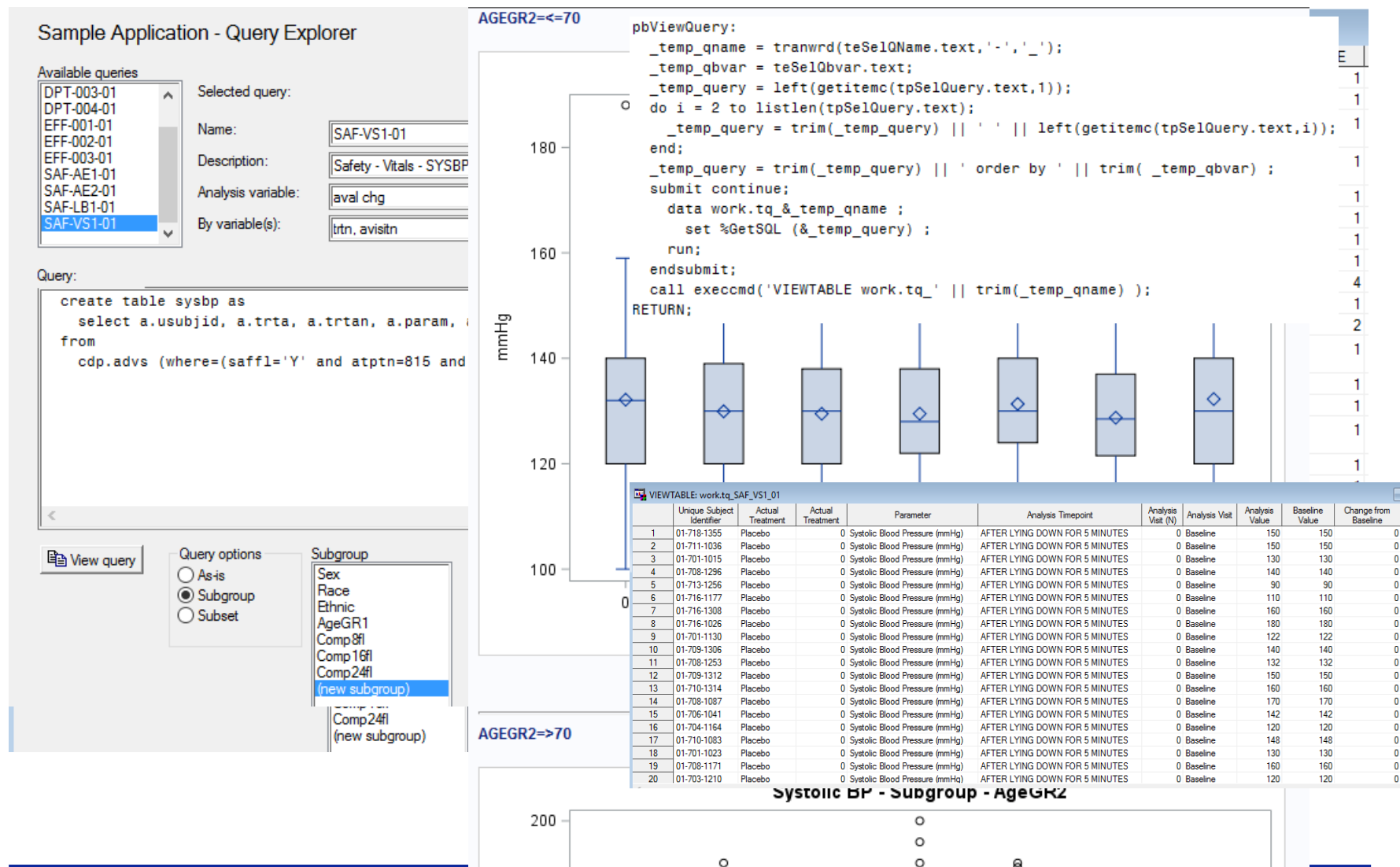
- Data warehouse – JSON dataset
- GetSQL macro
- SAS A/F application
- Local analysis macros and SAS PROCs

## A Sample Application - 1

---

- SAS A/F
- Table of queries based on SAP requirements
- Table of potential analyses (both SAS PROCs and user written macros)
  - each query has attributes that determine which PROCs / macros are appropriate and what variables are to be used in the analysis
- Selecting a query shows what analyses are possible
- User may specify subsets or subgroups (either based on existing variables of defined interactively)
- Analysis results are shown interactively and may be saved
- Code may be saved

## A Sample Application - 2



## Opportunities and Implications

---

- Opportunities:
  - Speedy responses to requests for new analyses
  - Applications can be distributed beyond programming department
  - ... Investigations are facilitated
- Implications:
  - Need well-defined queries (starts with well-defined analysis plans)
    - Can be validated
  - Need careful definition of the analyses, variables, and so on that are applicable to each specific query
  - Consider the control of results (including validation requirements)

## Conclusion

---

- Improving efficiency
  - Clinical trial reporting has fewer applications than might be expected
  - Find places where processes can be streamlined
  - Combining techniques can provide opportunities
- This mash-up is just one example

## Questions and comments

---

Your comments and questions are encouraged and valued

[stephen.prust@elderbrooksolutions.com](mailto:stephen.prust@elderbrooksolutions.com)