Diving deep in SAS® Macro World

Adapting and developing SAS Macro tools
(for reporting and exploratory analysis of MedDRA coded adverse events in clinical studies as MS Excel multi-tab reports)

Alex Ginev
CollegeBoard *(aginev@Collegeboard.org)*,
Delaware County Community College *(aginev@dccc.edu)*

Background: The context

Modeled after reporting tool known as MAED – MedDRA-based Adverse Event Diagnostics

- ***Initially developed and launched in 2009***
- ***Based on existing SAS code (shared by the author)***
- ***Designed for Regulatory authorities to display MedDRA coded AEs for Safety Signals using coding levels and Standardized MedDRA Queries (SMQs)***

Attempt to create and test SAS macro code for similar reporting output, structure and format that reproduces and expands the capabilities of the available MAED tool by

- **Producing multi-sheet Excel reports with common risk estimators for all treatment comparisons of interest**
- **Showing degree of disproportionality between treatments**

Content

- **Purpose: Present and discuss examples of three powerful SAS Macro techniques used in this project, namely:**

- Various ways to create, validate and use of *macro lists*
- Capturing and validating user input with *%WINDOW - %DISPLAY*
- Examples of use of iterative *(%do-%end*) and conditional *(%if-%then-%else*) macro function logic: within DATA steps and PROC statements / in combination with other (regular) SAS keywords and statements

The Model Product (*Reverse Engineering* needed?)

re·verse en·gi·neer·ing
noun
the reproduction of another manufacturer's product following detailed examination of its construction or composition.

Table 2: AE MedDRA SMQs summary at narrow search

Sorted by P-value

SMQ (Narrow Search)	STUDY DRUG (N = 166) Events	Number subject	Proporti (%)	PLACEBO (N = 166) Events	Number subject	Proporti (%)	STUDY DRUG vs. PLACEBO RD (per hundr	RD C.I. (lower bou	RD C.I. (upper bou	RR	(lower bound)	(upper bound)	OR	(lower bound)	(upper bound)	P-value
(1)Gastrointestinal nonspecific inflammation and dysfunctional conditions	120	57	34.34	75	41	24.7	9.64	-0.12	19.4	139	0.991	1.951	1.584	0.964	2.646	0.071
(2)Gastrointestinal nonspecific symptoms and therapeutic procedures	116	54	32.53	73	41	24.7	7.83	-1.86	17.52	1.317	0.933	1.858	1.47	0.886	2.447	0.145
(1)Glaucoma	5	5	3.01	1	1	0.6	2.41	-0.44	5.26	5	0.59	42.337	5.124	0.562	243.801	0.214
(1)Dyslipidaemia *	12	12	7.23	6	6	3.61	3.61	-1.24	8.47	2	0.769	5.203	2.078	0.699	6.908	0.225
(1)Peripheral neuropathy	5	4	2.41	1	1	0.6	1.81	-0.81	4.42	4	0.452	35.411	4.074	0.396	201.829	0.371
(1)Haemodynamic oedema, effusions and fluid overload *	8	7	4.22	12	11	6.63	-2.41	-7.27	2.46	0.636	0.253	1.601	0.62	0.199	1.308	0.468
(1)Accidents and injuries	42	31	18.67	56	37	22.29	-3.61	-12.29	5.06	0.838	0.547	1.283	0.801	0.451	1.414	0.497
(1)Ischaemic heart disease	1	1	0.6	6	3	1.81	-12	-3.55	1.14	0.333	0.035	3.172	0.329	0.006	4.162	0.623
inflammation	4	4	2.41	2	2	1.2	1.2	-1.66	4.07	2	0.371	10.771	2.025	0.285	22.627	0.685
(2)Gastrointestinal nonspecific dysfunction	4	4	2.41	2	2	1.2	1.2	-1.66	4.07	2	0.371	10.771	2.025	0.285	22.627	0.685
(1)Malignancies *	5	5	3.01	3	3	1.81	1.2	-2.09	4.5	1.667	0.405	6.861	1.687	0.322	11027	0.723
(2)Malignant or unspecified tumours *	5	5	3.01	3	3	1.81	1.2	-2.09	4.5	1.667	0.405	6.861	1.687	0.322	11027	0.723
(1)Depression and suicide/self-injury	11	11	6.63	12	9	5.42	1.2	-3.91	6.32	1.222	0.52	2.872	1.238	0.452	3.48	0.818
(1)Cerebrovascular disorders	1	1	0.6	0	0	0	0.6	-0.57	1.78	3	0.123	73.114	3.018	0.122	74.624	1
(2)Central nervous system haemorrhages and cerebrovascular accidents	1	1	0.6	0	0	0	0.6	-0.57	1.78	3	0.123	73.114	3.018	0.122	74.624	1
(3)Ischaemic cerebrovascular conditions *	1	1	0.6	0	0	0	0.6	-0.57	1.78	3	0.123	73.114	3.018	0.122	74.624	1
(1)Acute central respiratory depression	1	1	0.6	0	0	0	0.6	-0.57	1.78	3	0.123	73.114	3.018	0.122	74.624	1
(1)Noninfectious meningitis	1	1	0.6	0	0	0	0.6	-0.57	1.78	3	0.123	73.114	3.018	0.122	74.624	1
(1)Ischaemic colitis	1	1	0.6	0	0	0	0.6	-0.57	1.78	3	0.123	73.114	3.018	0.122	74.624	1
(1)Hyperglycaemia/new onset diabetes mellitus	2	2	1.2	2	2	1.2	0	-2.35	2.35	1	0.143	7.015	1	0.072	13.948	1
(1)Angioedema	11	5	3.01	6	6	3.61	-0.6	-4.45	3.25	0.833	0.259	2.677	0.828	0.196	3.332	1
(1)Asthma/bronchospasm	2	2	1.2	5	3	1.81	-0.6	-3.22	2.02	0.667	0.113	3.938	0.663	0.055	5.87	1
(2)Suicide/self-injury *	0	0	0	1	1	0.6	-0.6	-1.78	0.57	0.333	0.014	8.124	0.331	0.013	8.192	1
(1)Extrapyramidal syndrome	0	0	0	1	1	0.6	-0.6	-1.78	0.57	0.333	0.014	8.124	0.331	0.013	8.192	1
(2)Parkinson-like events	0	0	0	1	1	0.6	-0.6	-1.78	0.57	0.333	0.014	8.124	0.331	0.013	8.192	1
(1)Biliary disorders *	0	0	0	1	1	0.6	-0.6	-1.78	0.57	0.333	0.014	8.124	0.331	0.013	8.192	1
(2)Functional, inflammatory and gallstone related biliary disorders *	0	0	0	1	1	0.6	-0.6	-1.78	0.57	0.333	0.014	8.124	0.331	0.013	8.192	1

Showing Narrow SMQ

Introduction | SOC | HLGT | HLT | PT | Broad | Algorithmic | Narrow

ct_121_28aug12_1529.xls - Microsoft Excel

Sorted by Risk Difference (in this example)

WARNING: THIS IS ONLY AN EXPLORATORY ANALYSIS!
Secondary preferred terms excluded.
P-values should be used for ranking purposes only, not for determining statistical significance.

SOC	STUDY DRUG XR 400 MG QD (N = 295) Events	Number of subjects	Proportion (%)	COMPARATOR DRUG 200 MG BID (N = 148) Events	Number of subjects	Proportion (%)	STUDY DRUG Vs, COMP RD (per hundre	RD C.I. (lower boun	RD C.I. (upper boun	RR	RR C.I. (lower bound	RI (uppe
Gastrointestinal disorders	98	70	23.73	11	11	7.43	16.3	9.86	22.73	3.193	1.745	5
Injury, poisoning and procedural complications	34	31	10.51	5	5	3.38	7.13	2.58	11.68	3.111	1.235	7
General disorders and administration site conditions	41	35	11.86	9	9	6.08	5.78	0.45	11.12	1.951	0.964	
Nervous system disorders	40	33	11.19	10	9	6.08	5.11	-0.16	10.37	1.84	0.904	3
Psychiatric disorders	34	31	10.51	8	8	5.41	5.1	0.05	10.15	1.944	0.917	4
Skin and subcutaneous tissue disorders	41	26	8.81	7	7	4.73	4.08	-0.62	8.79	1.863	0.828	4
Infections and infestations	162	108	36.61	76	49	33.11	3.5	-5.86	12.87	1.106	0.841	1

Output from the available SAS MAED macros)

By *SOC* AE reports

Study X
All MedDRA Levels By Descending Incidence Rates
(MedDRA vXX-0)
SOC = General disorders and administration site conditions

System/Organ/Class High Level Group Term High Level Term Preferred Term	Treatment Group Tx 1 (N=xx)	Tx 2 (N=xx)	Tx 3 (N=xx)	Total (N=xx)	p-Value*
General disorders and administration site conditions (SOC)	72 (20.2%)	62 (14.9%)	62 (15.0%)	168 (14.7%)	0.5982
General system disorders NEC (HLGT)	60 (13.5%)	46 (11.1%)	46 (11.1%)	152 (12.2%)	0.2334
Pain and discomfort NEC (HLT)	38 (6.2%)	28 (6.7%)	30 (7.3%)	96 (7.7%)	0.3880
Chest pain (PT)	23 (5.2%)	16 (3.8%)	15 (3.6%)	54 (4.3%)	0.3353
Non-cardiac chest pain (PT)	16 (3.9%)	12 (2.2%)	14 (3.7%)	42 (3.9%)	0.7408
Chest discomfort (PT)	3 (0.7%)	3 (0.7%)	13(1.7%)	9 (0.7%)	1.0000
Axillary pain (PT)	1 (0.2%)	0 (0.0%)	0 (0.0%)	1 (0.1%)	0.3283

By *p-value* AE reports

Study X
Preferred Terms by p-Value

p-Value*	Preferred Term (MedDRA vXX-0)	Treatment Group Tx 1 (N=xx)	Tx 2 (N=xx)	Tx 3 (N=xx)
0.0196	PARAESTHESIA	7 (2.7%)	3 (0.8%)	2 (0.5%)
0.0290	VASCULAR STENT RESTENOSIS	1 (0.4%)	0 (0.0%)	7 (1.1%)
0.0435	HEADACHE	9 (2.0%)	16 (3.8%)	5 (1.2%)

p-Value*	Standardized MedDRA Query (MedDRA Version YY.0)	Treatment Group Tx 1 (N=xx)	Tx 2 (N=xx)	Tx 3 (N=xx)
0.0176	Shock (SMQ)	8 (2.9%)	0 (0.0%)	4 (1.5%)
0.0451	Haemorrhage terms (excl laboratory terms) (SMQ)	33 (8.0%)	35 (8.0%)	52 (15.6%)
0.0497	Respiratory failure (SMQ)	3 (0.7%)	0 (0.0%)	0 (0.0%)

Prerequisites, Features and Requirements

Original MAED programs create two types of AE reports:

- PDF/RTF reports *grouped by MedDRA term / SMQ and sorted by a single test statistic (Chi-squared test p-value, "By p-value" report)*
- *MedDRA AE terms hierarchically grouped within their SOC ("all in one" report)*

- Generates the pre-processing SAS code from HTML user interface

Both original and updated MAED tools have assumptions and requirements pertaining to

- *Location for input and output sets (report)*
- *Uniform locations (subfolders) and naming conventions for MedDRA versions*
- *Global macro variables allow input for some flexibility*
- *Every AE or SMQ is processed as 2 x 2 table (comparison) – proc FREQ / TABLES with several statistical test options is used*
- *Standard formats and names of the core time and Tx variables (ADaM data)*
- *Analysis data sets from the study (main inputs):*

ADAE data set (Analysis Data Set for Adverse Events) and

ADSL data set (Analysis Data Set for Subject Level)

Idea and Strategy: Steps to solve the *multi-sheet reporting task*

- **STEP 1: Prepare the data:** Declare global macro variables, subset the ADSL as 2 x 2 comparison sets of interest for the study, clean up blanks, missing data, subset on SAE or other conditions - if needed, define the order of *Active vs Comparator* arm
- **STEP 2: Receive and manage user input:** Type of AE report to be created (format, full report versus a single or a subset of all Tx comparisons of interest, define custom rules, etc.)
- **STEP 3: Run iteratively the original MAED macros** for each of the selected 2 x 2 Tx subsets
- **STEP 4: Rename / Index all columns** of the output sets with statistics so they can be **merged by MedDRA (AE) level term or by SMQ** (Narrow and Broad scope)
- **STEP 5: Check for and clean up duplicate columns** for event and subject count data in the combined sets
- **STEP 6: Produce the whole multi-tab report** (proc REPORT and ODS EXCEL join forces here)

STEP 1: Get your data ready for input

Subset in series of data steps like these – depending on needs and interest

```
/* Define the input AE data, as well as the desired TX subsets of interest.                    */
/* Those should represent 2-arm sets - they will have to be processed as 2 x 2 tables by MAED macros; */

data ae;
  set aedata.adae;
  where SAFFL='Y' and TRTEMFL='Y'; .
data dm_nsf;

  set dmdata.adsl;
    if left(put(&trt,8.)) in('.','') then
      do;
        put 'NOTE: Missing treatment code-- deleting: ' &usubjid= &trt= ;
        delete;
      end;
    if (&trt ne 'SCRNFAIL') then do;
        output dm_nsf;
    if (&trt eq 'Tx_1') or (&trt eq 'Tx_2') then do;
        &trt = 'Tx_1_2';
        output dm_nsf;
        end;
    end;
    where SAFFL='Y';
run;
```

```
%let poplist=;
%let poplist=dm_Tx1_p dm_Tx2_p dm_Tx2_Tx1 dm_active_p;
```

```
data &poplist;
       holdtrt = &trt;
       if holdtrt in ('Tx_1', 'PLACEBO') then
         do;
           ** Append "A_" for Test or "P_", "B_", etc. for Ref based on TRT value;
           if (holdtrt eq 'Tx_1') then &trt = "A_"||holdtrt;
             else if (holdtrt eq 'PLACEBO')  then &trt = "P_"||holdtrt;
             output dm_Tx1_p;
         end; /* ... and so on! */
```

```
data dm_active_p;
     set dm_active_p;
     if &trtdesc in ("Tx dosage 1", "Tx dosage 2") then &trtdesc="Tx dos1+dos2";
run;
```

STEP 1: Get your data ready for input (next)

Declare global macro variables and paths, set MedDRA /SMQ versions, load (compile) the macros

```
*------- Set the path for the output files -------*;
%let outputpath=H:\My Documents\MedDRA\data_studyx\output;
*------- Specify the MedDRA version in X.X format  -------*;
%let meddraver=19.0;
%let smqversion=22.0;
*------- Specify the main variable names  -------*;
%let usubjid=USUBJID; * the patient ID variable name;
%let trt=ACTARMCD; * the coded treatment group variable name;
%let trtdesc=ACTARM; * the de-coded treatment group variable name (contains the descriptive name);
%let aeterm=AEDECOD; * the preferred term variable name;
%let repterm=AETERM; * the reported adverse event term;
%let bodysys=AEBODSYS; * the bodysystem or system/organ/class variable name;

options notes nofmterr nodate mprint mlogic symbolgen;
ods escapechar= '^' ;
%global meddraroot;
%let meddraroot=H:\My Documents\MedDRA\;
%let mac_root=H:\My Documents\MedDRA\programs\;
```

… Followed by *%include* and *libname* statements for input and output sets

STEP 2: Manage User Input, as

- Use of ***%window*** *and* ***%display*** features (works in SAS for Windows only)
- Specify the desired number of Tx comparison sets
- Validates the input – checks for misspelled / invalid Tx filenames
- Lists *&badnames* (if any) and gives warning if needed

STEP 2: Manage the User Input

- *%WINDOW* can offer special formatting based on selection rule(s) for rows or cells

```
%window Highlight
    #5 @5   'Highlight AEs/SMQs with low Fisher right-sided p-values'
    #7 @10  '(<0.050, blue on light pink, large font) or RR>=2.5 (blue on light yellow, large font)
    #9 @15  'This output is effective with PDF and MS Excel Workbook formats only'
    #11 @20 'Please enter Yes to accept; No, blank, or any string to reject/move on:'
    #13 @25 input 7 attr=underline;
%display Highlight;
```

Later in the sequence this can trigger a COMPUTE / CALL DEFINE block in proc REPORT with *%if-%then-%do-%end* macro function logic …

```
%if &input=%sysfunc(trim(Yes)) or &input=%upcase(Yes) %then %do;

    compute %sysfunc(trim(&cols&_y_.p_e));
        if . ne %sysfunc(trim(&cols&_y_.p_e)) < 0.050 then
        call define(_col_, "style/replace", "style=[backgroundcolor=mistyrose color=blue fontstyle=italic fontsize=6]");
    endcomp;
    compute %sysfunc(trim(&cols&_y_.p_rr));
        if %sysfunc(trim(&cols&_y_.p_rr)) ge 2.50 then
        call define(_col_, "style/replace", "style=[backgroundcolor=lightyellow color=blue fontstyle=italic fontsize=6]");
    endcomp;
                                                    %end;
```

STEP 3: Run iteratively the *%aefreq* and some other macros

- ***%AEFREQ*** **creates Tx comparison sets with aggregated subject and event counts, plus exploratory statistics (*Fisher's exact test p-value, RD and RR with 95 % confidence bounds*) by MedDRA level term (PT, HLT, HLGT and SOC) and by SMQ (Narrow and Broad scope)**

```
data _null_;
    %do j=1 %to %eval(&items);
    %let popset&j=%scan(&goodlist, &j); %put &&popset&j;

        %AEFREQ(aeset=ae, popset=&&popset&j, aeterm=&aeterm, trt=&trt, trtdesc=&trtdesc, usubjid=&usubjid, outset=pt&j, stratvar=&stratvar,
        pflag=1, stat=&stat, pname_e=&pname_e, pname_rd=&pname_rd, pname1=&pname1, pname2=&pname2, pname_rr=&pname_rr, pname3=&pname3, pname4=&pname4);
        /* .... More code here ..... */
    %end;
run;
```

STEP 4: Rename and index all sets and columns

4.1 Reads the sets produced by *%aefreq* macro → Renames all variables (appends the set name to each variable name) except for the AE level term name (variable)

```
%let popset&j=%scan(&goodlist, &j);
%let dsid=%sysfunc(open(&&set&j, i));
%let varlist=;
data copy&&set&j;
    length sheetlabel $30.;
    set &&set&j;
        sheetlabel=trim(upcase("&lib"||" "||"Name"));
        if find(sheetlabel, '1', 'i') ge 1 then sheetlabel='SMQ - Broad Scope';
        else if find(sheetlabel, '2', 'i') ge 1 then sheetlabel='SMQ - Narrow Scope';

        %do i=1 %to %sysfunc(attrn(&dsid, nvars));
          %let varlist=&varlist %sysfunc(varname(&dsid, &i));
          %let varname&i=%sysfunc(varname(&dsid, &i));
              %if &&varname&i=variable %then %do;
                  rename &&varname&i = &&varname&i; %end;
              %else %do;
                  rename &&varname&i = &&set&j&&varname&i;
              %end;
              %put &&varname&i;
        %end;
        %put varlist=&varlist;
run;
```

Builds macro lists of the column names iteratively (log check)

STEP 4: … Then merge (Continued)

4.2 Creates and assigns event and subject count sum labels for each Tx arm in the "copy" sets →

```
%do _i=1 %to %eval(&n_labs);
    %global events&_i&lib&j;
        proc sql;
            select sum(&&set&j..e&_i) into :events&_i&lib&j
            from copy&&set&j;
            alter table copy&&set&j modify &&set&j..e&_i
            label="&&tx&j._c&_i, |Event |Counts |=%left(&&events&_i&lib&j)";
            ;
```

Number of unique Tx labels in the subset

→ 4.3 Builds a macro list of the renamed ("copy") sets to be passed to a MERGE statement in subsequent DATA step

```
/* ... preceeded by other code here ... */
        %let name&j=copy&&set&j;
        %let mergelist=&&mergelist %trim(&&name&j);
        %let dsid=%sysfunc(close(&items));
        %put &mergelist;
    %end;

data big&lib;
    merge &mergelist;
    by sheetlabel variable;
run;
```

To become a sheet for each MedDRA level term or SMQ

STEP 4: Rename and index all sets and columns, then merge (Last)

→ 4.4 Replace missing values of subject and event counts in the combined dataset (MedDRA level term / SMQ) with zeroes:

```
data big&lib;
    set big&lib;
    by sheetlabel variable;
    %do y=1 %to %eval(&items);
        array events_&y{*} %sysfunc(trim(&lib&y.el))-%sysfunc(trim(&lib&y.e&n_arm)) _nu
        array counts_&y{*} %sysfunc(trim(&lib&y.nl))-%sysfunc(trim(&lib&y.n&n_arm)) _nu
        /* ... some other code here ... */
            do i=1 to &n_arm;
                if events_&y[i] =. then events_&y[i]=0;
                if counts_&y[i] =. then counts_&y[i]=0;
                /* ... other statements here ... */
            end;
        drop i;
    %end;
run;
```

STEP 5: Check for and remove duplicate count columns from the combined sets

- After merging the Tx comparison sets there will almost always be *duplication of some subject and event count columns in the combined (MedDRA level term / SMQ sets)*
- Duplicates should be removed

5.1 Collect info about all columns in the combined datasets

```
proc contents data=big&lib out=varinfo; run;
```

→ 5.2 Extract subject / event count variables section (where duplicates may appear)

```
proc sql; ** all_counts lists variables and labels of the count columns (Active and Comparator);
    create table all_counts as
    select distinct name, label
    from varinfo
    where (name like '%c1' or name like '%e1'
    or name like '%c2' or name like '%e2');
```

Holds unique column label information from the combined count data

1 indicates *Active* (left), 2 - *Comparator* (right) for subject (c) and event (e) counts respectively

STEP 5: Check for and remove duplicate count columns in the merged sets (Next)

5.3 Creates Tx identifiers in a data step (not shown) and removes duplicates

```
proc sort data=all_counts out=nodupcnts nodupkey; by label; run;
proc sort data=nodupcnts; by type id tx_id; run;
```

nodupcnts will list the unique subject and event counts for each Tx in the final, cleaned set (Excel sheet).

	Treatment 1			Treatment 2		
Preferred Term (MedDRA vXX-0)	Tx 1, Subjects w/Event (N=xx)	Tx 1 Proportion, Subj. w/Event(%)	Tx 1, Event Counts =YY	Tx 2, Subjects w/Event (N=xx)	Tx 2, Proportion, Subj. w/Event(%)	Tx 2, Event Counts =YY
PULPITIS DENTAL	0	0.00	0	1	0.24	1
PUNCTURE SITE PAIN	0	0.00	0	0	0.00	0
PYREXIA	5	1.20	5	1	0.24	1
PYURIA	0	0.00	0	0	0.00	0
RADICULOPATHY	0	0.00	0	1	0.24	1
RASH	2	0.48	3	6	1.44	6
RASH GENERALISED	0	0.00	0	1	0.24	1
RASH MACULO-PAPULAR	0	0.00	0	1	0.24	1
RASH PRURITIC	1	0.24	1	0	0.00	0
RECTAL HAEMORRHAGE	0	0.00	0	0	0.00	0
RECTAL POLYP	0	0.00	0	0	0.00	0
RED BLOOD CELLS URINE POSITIVE	1	0.24	1	0	0.00	0

STEP 5: Reconstruct the combined sets (ready for Excel report sheets)

5.4 Using ordered lists of the remaining, non-duplicate *Active* and *Comparator* count columns

```
proc sql;
    select name
    into :tx_cols separated by ' '
    from nodupcnts
    ;
    select name
    into :stat_cols separated  by ', '
    from all_stats
    ;
    select name
    into :tx_cols_sql separated by ', '
    from nodupcnts
    ;
    create table dist_counts as
    select sheetlabel, variable, &tx_cols_sql
    from big&lib
    ;
quit;
```

Passed to the next proc SQL / SELECT query

Final, ordered, non-duplicate count section of the report sheet – to be re-merged with the test statistics block

STEP 5: Re-identify the remaining, non-duplicate count columns

5.5 … Last!: The non-duplicate count columns need to be re-indexed / re-identified again …

WHY?: To force proc REPORT to map and output exactly this arrangement of the count/proportion section

HOW?: *Use proc DATASETS with MODIFY / RENAME statements + %do-%end macro loop processing inside*

```
proc datasets library=work nolist;
    modify dist_counts;
rename
    %do i=1 %to %eval(&arm_cnt);
        %let var&i=%scan(&tx_cols, &i);
        %let vartype&i=%substr(&&var&i, %length(&&var&i));
        %let arm_type&i=%substr(&&var&i, %length(&&var&i)-1);

        %if &&vartype&i=1 %then
                %do;
                    &&var&i = &lib._act_%unquote(&&tx_id_num&i)_&&arm_type&i
                    %put &&var&i;
                %end;
        %else %if &&vartype&i=2 %then
                %do;
                    &&var&i = &lib._comp_%unquote(&&tx_id_num&i)_&&arm_type&i
                    %put &&var&i;
                %end;
    %end;
    ;
quit;
```

Active Tx count columns

Comparator Tx count columns

STEP 6: Produce the multi-tab Excel report

Ordered, non-duplicate count, combined sets (MedDRA term level / SMQ) will be passed to proc REPORT

```
/* Merge again the COUNT and STAT sets to get the
whole AE/SMQ term data ready for reporting */
data reportset_&lib.;
    merge dist_counts test_stats;
    by sheetlabel variable;
run;

%mend nodup_counts;
```

The reporting macro (*%excel_report3*) is called iteratively – through a list of MedDRA term level / SMQ combined sets

```
%global libnames filecount items libcount;
%let libnames=pt hlt hlgt soc smq1 smq2;

%let items=%sysfunc(countw(&goodlist)); %put &items;
%let libcount=%sysfunc(countw(&libnames)); %put &libcount;
%let meddraver=%sysfunc(tranwrd(&meddraver,.,-));
%let smqversion=%sysfunc(tranwrd(&smqversion,.,-));
%let meddratext=(MedDRA v&meddraver)

%do l=1 %to %eval(&libcount);

    %let libref&l=%scan(&libnames, &l, ' ');
    %let dataset&l=reportset_&&libref&l;
```

```
        /* ... other macro assignments and statements (SQL queries) in the inner loop
        creating tab headers, titles and some labels ... */
    %end;
    ;
    %excel_report3(dataset=&&dataset&l, cols=&&col_id&l, title=&&title&l,
    collhead=&&collhead&l, smqversion=&smqversion);
%end;
ods excel close;
ods listing;

proc datasets lib=work memtype=data;
    delete big: copy: __: pt: hlt: hlgt: soc: smq: smq1: smq2: ; run; quit;

%mend XLS_report3;
```

STEP 6: The multi-tab Excel report

ODS EXCEL with options controls for the overall structure of the workbook, *i.e. tabs (by groups / MedDRA level term sheets), embedded titles, auto filter (dropdown menus), frozen headers, etc.*

```
ods listing close;
ods excel file="&outfile" options(sheet_interval="bygroup"
suppress_bylines="yes" sheet_name="#byval1" frozen_headers='yes' embedded_titles="yes"
embed_titles_once="yes" embedded_footnotes='yes' tab_color='yellow' autofilter='all' zoom='75');
ods escapechar='^';
```

Proc REPORT does the rest

													SECTION 3: TEST STATISTICS FOR ALL PAIRS						
SECTION 2: ACTIVE (colored) and COMPARATOR (Grey) SUBJECT and EVENT COUNTS, RATES													Treatment Comparison: 1						
SECTION 1: MedDRA Item	Tx 1			Tx 2			Tx 1+2			Tx 3			Tx 1 vs Tx 3						
System/Organ/Class (MedDRA vXX-0)	Tx 1, Subjects w/Event (N=xx)	Tx 1 Proportion, Subj. w/Event(%)	Tx 1, Event Counts =yy	Tx 2, Subjects w/Event (N=xx)	Tx 2, Proportion, Subj. w/Event(%)	Tx 2, Event Counts =yy	Tx 1+2, Subjects w/Event (N=xx)	Tx 1+2, Proportion, Subj. w/Event(%)	Tx 1+2, Event Counts =yy	Tx 3, Subjects w/Event (N=xx)	Tx 3, Proportion, Subj. w/Event(%)	Tx 3, Event Counts =yy	Fisher's right-sided p-value *	Risk Difference, %	Lower CI Bound, RD, % *	Upper CI Bound, RD, % *	Relative Risk	Lower CI Bound, RR *	Upper CI Bound, RR *
CONGENITAL, FAMILIAL AND GENETIC DISORDERS	0	0.00	0	1	0.24	1	1	0.12	1	1	0.24	1	1.000	-0.242	-0.716	0.232	0.000	.	.
EAR AND LABYRINTH DISORDERS	2	0.48	2	7	1.68	7	9	1.08	9	6	1.21	6	0.939	-0.729	-1.976	0.519	0.398	0.078	2.040
ENDOCRINE DISORDERS	1	0.24	1	1	0.24	1	2	0.24	2	5	0.97	5	0.970	-0.728	-1.783	0.328	0.249	0.028	2.217
EYE DISORDERS	3	0.72	3	6	1.44	10	10	1.08	13	5	0.97	6	0.776	-0.246	-1.493	1.002	0.746	0.168	3.314
GASTROINTESTINAL DISORDERS	42	10.12	61	45	10.10	67	84	10.11	128	46	11.14	68	0.722	-1.018	-5.216	3.181	0.909	0.612	1.349
GENERAL DISORDERS AND ADMINISTRATION SITE CONDITIONS	90	20.24	122	62	14.90	92	146	17.57	214	62	15.01	92	0.030	5.229	0.051	10.407	1.348	1.000	1.817
HEPATOBILIARY DISORDERS	6	1.45	7	0	0.00	0	6	0.72	7	6	1.21	6	0.503	0.235	-1.324	1.794	1.194	0.367	3.882
IMMUNE SYSTEM DISORDERS	2	0.48	2	2	0.48	2	4	0.48	4	0	0.00	0	0.251	0.482	-0.184	1.148	I	.	.
INFECTIONS AND INFESTATIONS	47	11.33	61	39	9.38	49	86	10.35	110	38	9.20	42	0.186	2.124	-2.007	6.256	1.231	0.821	1.846
INJURY, POISONING AND PROCEDURAL COMPLICATIONS	20	4.34	25	18	4.09	30	35	4.21	55	24	5.81	27	0.870	-1.474	-4.462	1.515	0.746	0.411	1.354

STEP 6: Inside PROC REPORT: Managing Grouping and Common (MedDRA term) columns

Define the common *part* <u>once</u> (MedDRA term / SMQ)

```
%if &smq_by^= %then %do;
  column sheetlabel variable pt_cnt pt_name;
  define variable / group "&collhead" flow style(column)={width=2in font_weight=bold font_style=italic} left;
  define pt_cnt / order style(column)={width=0.6in font_weight=bold font_style=italic} center;
                  %end;
                  %else %do;
  column sheetlabel variable;
  define variable /display "&collhead" flow width=30 left;
                       %end;
 by sheetlabel;
 define sheetlabel / group noprint;
```

STEP 6: Inside PROC REPORT: Setting Count and Proportion columns

Displaying the non-duplicate *Count and Proportion* columns. Switches between styles / formats and iterates through colors depending on the type of Tx arm (*Active vs Comparator*)

```
%do _j_ = 1 %to %eval(&act_pairs);
* Shows all Active TX (subject and AE counts) columns: Block #2;
    column ("&&tx_label&_j_" %sysfunc(trim(&cols._act_&_j_._c1)) %sysfunc(trim(&cols._act_&_j_._e1)));
    %let color&_j_=%scan(&color_list, &_j_, ' ');
    %let style1=style(column)=[width=0.8in color=black backgroundcolor=&&color&_j_ font_weight=bold
                               borderrightcolor=red borderrightwidth=4]
         style(header)=[width=1.4in height=1.2in color=red backgroundcolor=aliceblue font_weight=bold];
    define %sysfunc(trim(&cols._act_&_j_._c1)) / display &style1 center;

    %if &smq_by^= %then %do;
        define %sysfunc(trim(&cols._act_&_j_._e1)) / analysis sum &style1 center;
                        %end;
                        %else %do;
        define %sysfunc(trim(&cols._act_&_j_._e1)) / display &style1 center;
                        %end;
%end;
%do k=%eval(&start_comp_pairs) %to %eval(&all_pairs);
* Shows Comparator/Placebo subject and AE count columns: Block #2;
    column ("&&tx_label&k" %sysfunc(trim(&cols._comp_&k._c2)) %sysfunc(trim(&cols._comp_&k._e2)));
    define %sysfunc(trim(&cols._comp_&k._c2)) / display &style2 center;

    %if &smq_by^= %then %do;
        define %sysfunc(trim(&cols._comp_&k._e2)) / analysis sum &style2 center;
                        %end;
                        %else %do;
        define %sysfunc(trim(&cols._comp_&k._e2)) / display &style2 center;
                        %end;
%end;
```

STEP 6: Inside PROC REPORT: Test Statistics Section

Block with test statistics for all *selected pairs (Tx comparison sets)*

```
%do _y_=1 %to %eval(&items);
        column (("&trtlabel.: &_y_" ("&&tx&_y_._cl vs &&tx&_y_._c&n_arm"
                %sysfunc(trim(&cols&_y_.p_e)) %sysfunc(trim(&cols&_y_.p_rd))
                %sysfunc(trim(&cols&_y_.pl)) %sysfunc(trim(&cols&_y_.p2))
                %sysfunc(trim(&cols&_y_.p_rr)) %sysfunc(trim(&cols&_y_.p3))
                %sysfunc(trim(&cols&_y_.p4)))));
;
        define %sysfunc(trim(&cols&_y_.p_e))  /display order=internal f=5.2
                "Fisher's |right-sided |p-value *" style(header)=[color=red
                backgroundcolor=yellow] style(column)=[width=0.8in font_weight=bold
                borderrightcolor=green borderrightwidth=3] format=5.3 center;

        /* ... MORE DEFINE statements here ... */
;
        compute %sysfunc(trim(&cols&_y_.p_rd));
            %sysfunc(trim(&cols&_y_.p_rd)) = %sysfunc(trim(&cols&_y_.p_rd))*100;
        endcomp;

        /* ... TWO MORE COMPUTE blocks here ... */

    %if &input=%sysfunc(trim(Yes)) or &input=%upcase(Yes) %then %do;

        /* COMPUTE blocks with conditional formatting (CALL DEFINE) from macro
        user input - as seen earlier ... */
;                                                          %end;
%end;
```

Other ways to list and store file or variable names as macro variables?

Use of UNNAMED PIPE: FILENAME + PIPE + CALL SYMPUT in a DATA step

```
filename pipedir pipe "dir ""%unquote(&dir)"" /b" lrecl=32767;
data _null_;
     infile pipedir truncover; input popname $char1000.;
     popname=scan(popname, 1); count = left(put(_n_,6.));
     call symput('popset'||count, popname);
     call symput('filecount', count);
     put popname=;
run;
filename pipedir clear;
```

Location (path) of the sets

Number of files found – stored as global macro variable

Global, indexed macro variables holding the names of the files in the directory

Use of DICTONNARIES / PROC SQL

```
proc sql noprint;
    select distinct memname, count (distinct memname)
    into :set1-:set999, :setnum
    from dictionary.columns
    where libname=upcase("&lib");
 quit;
```

Final Notes

Thank You + Questions?

- **What this <u>IS NOT</u> about?:**
  - Presenting and discussing the original SAS macro code
  - Claiming a genuinely new reporting tool was created
- **Then, what this <u>IS</u> about?:**
  - Sharing my SAS learning and programming experience with using common but powerful macro tools and functions in the context of this project
  - Hoping to demonstrate that "*Learning-by-Doing*" really works

Resources and Paper References

- ***MAED Service: FDA-Developed Tool for Clinical AE Data Signal Detection,*** *Xin (Joy) Li; Chuck Cooper; Zhongjun Luo CDER/Office of Translational Sciences/Office of Computational Sciences*
- ***A SAS® tool for analysis of AE data,*** *Chris Holland, 2010 (Paper AD13, PharmaSUG 2010)*
- ***Creating an Excel report: A comparison of the different techniques,*** Romain Miralles, Clinovo, Sunnyvale, CA (Paper 074-2011, SAS Global Forum 2011)
- ***Creating Multi-Sheet Excel Workbooks the Easy Way with SAS®,*** Vincent DelGobbo, SAS Institute Inc., Cary, NC (Paper 120-2007, SAS Global Forum 2007)
- ***Beyond the Basics: Advanced PROC REPORT Tips and Tricks,*** Allison McMahill, SAS Institute Inc., Cary, NC (Paper 276-2007, SAS Global Forum 2007)
- ***Safely Merging Many Datasets,*** Taylor Young, The Broad Institute of Harvard and MIT, Cambridge, MA, USA (Paper 409-2011, SAS Global Forum 2011)
- ***Getting More Out of "INTO" in PROC SQL: An Example for Creating Macro Variables,*** Mary-Elizabeth Eddlestone, SAS Institute Inc., Albany, NY
- ***Storing and Using a List of Values in a Macro Variable,*** Arthur L. Carpenter California Occidental Consultants, Oceanside, California (Paper 028-30, SUGI 30)
- ***The Mystery of the PROC SORT Options NODUPRECS and NODUPKEY Revealed,*** Britta Kelsey, MGI Pharma, Inc., Bloomington, MN (Paper 037-30, SUGI 30)
- ***Changing Data Set Variables into Macro Variables,*** William C. Murphy, Howard M. Proskin and Associates, Inc., Rochester, NY (Paper 050-2007, SAS Global Forum 2007)
- ***Unleash the Power of PROC REPORT with the ODS EXCEL Destination,*** Devi Sekar, RTI International (Paper 2479-2018)
- ***The FILENAME Statement: Interacting with the world outside of SAS®,*** Chris Schacherer, Clinical Data Management Systems, LLC (119-2012, SAS Global Forum 2012)
- ***A General-Purpose Macro to Obtain a List of Files: Plus Macro Programming Techniques*** Mel Widawski, UCLA, Los Angeles, California