

Paper 170-2012

**Practical Application of SAS® Capabilities for Pharma Goaling and Performance Review**

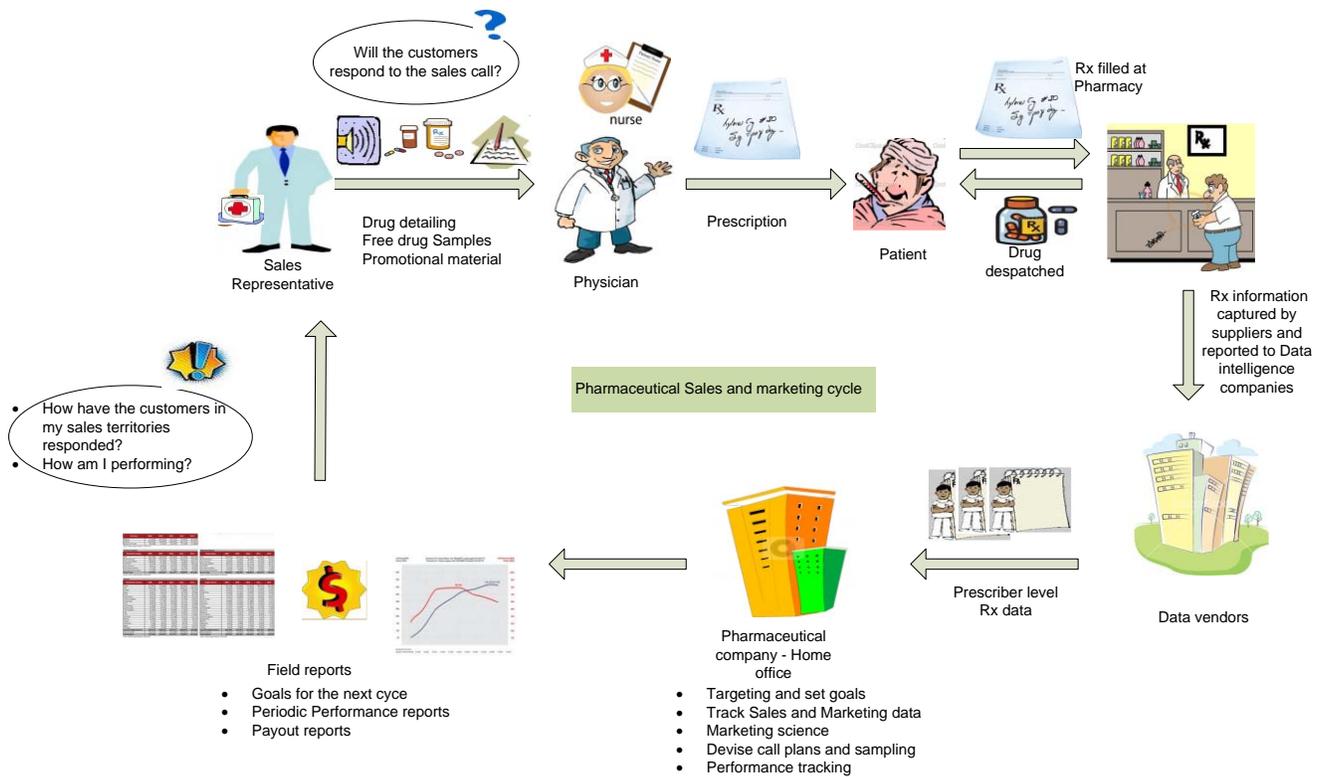
Ramya Purushothaman, Cognizant Technology Solutions, Teaneck, NJ

**ABSTRACT:**

This paper discusses a Pharma application that uses SAS to leverage internal and purchased information such as Sales and Marketing data including drug prescriptions, dollar and unit demand, target prescribers, and key customer account profiles to set goals, measure sales performance and identify trends across the geography. The capability of SAS to handle huge volumes of data seamlessly provides an advantage over other technologies. The reusability of SAS macros makes SAS solutions extensible across various brands, sales teams, and geo levels for reporting. All of these tasks are performed through familiar Base SAS procedures, functions, statements, and options. The paper explains how the business need is addressed using SAS by accessing, cleansing, and transforming information.

**INTRODUCTION:**

It is a challenging time for sales forces especially in the Pharma arena. Facing limited time with customers and growing competition, field sales forces need to perform at best-in-class levels to keep the top line growing.



**Figure 1. Typical Pharmaceutical Sales and marketing cycle**

The sales rep calls on a physician/organization to detail about the drug in his portfolio. The physician may choose to prescribe that product to his patients. The patient produces the prescription in a Retail pharmacy and buys the product. The

prescription may be filled through a number of other means including Mail order or Long Term Care pharmacies which focus on Nursing homes and other LTC facilities. The prescription fills and refill information is collected by suppliers and sold to Data collection companies have a wide network of suppliers across the country to cover the nation. These Data Intelligence companies then cleanse the data, consolidate them, project for missing information etc. The Pharmaceutical companies purchase this information (which includes their own sales and competitor sales also) from these vendors and use it for various purposes including Targeting, Call planning, Market research etc. One other important use of this data is to set goals for the sales representatives for the future and also to measure their performance in the selling cycle that was just completed. This is the function of the Incentive Compensation (IC) group in the company. Most pharmaceutical companies look at the historical prescribing pattern of their customers and extrapolate to the future as goals for their sales team. The approach detailed in this paper assumes this backward looking method of targeting using prescription and sales data as quota-setting factors for the compensation plan.

### **NEED FOR A STABLE PROCESS FOR INCENTIVE COMPENSATION:**

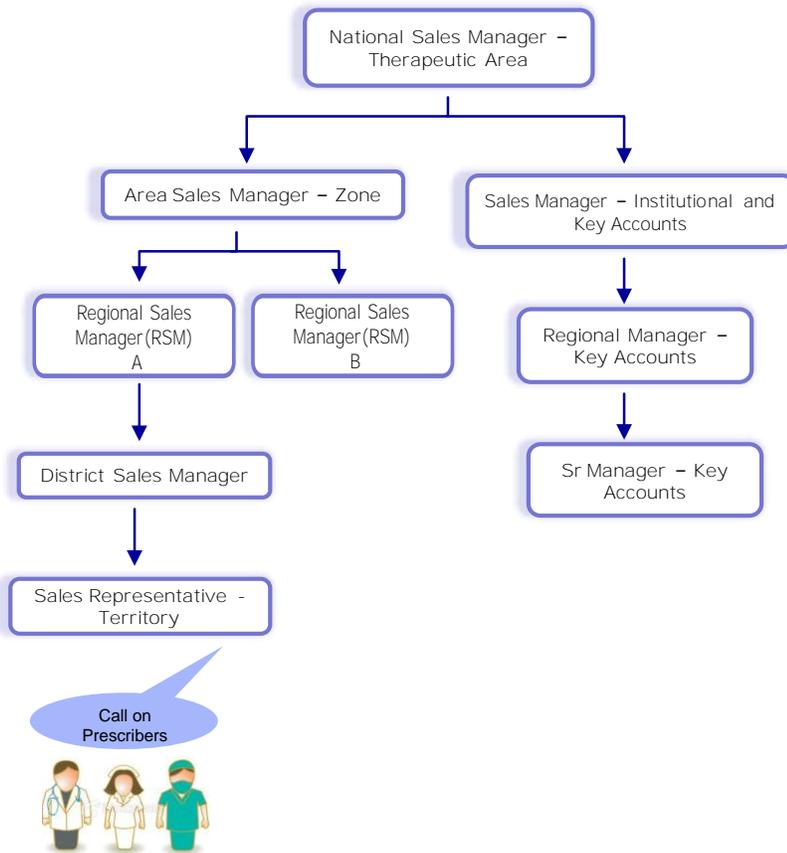
“Sales organizations that fail to execute reporting and payments of incentive commissions in a timely manner will decrease sales force productivity by 20% because of lost selling time, reduced motivation and the absence of organizational trust.” - Gartner

Many pharmaceutical companies tend to use rudimentary spreadsheet-based Incentive Compensation. Such applications are slow and inaccurate, do not provide easy visibility to the business rules of the plan, lack audit trails and version control, do not easily support the complex rules required for new sales force models, and require significant manual entry and ongoing updates. With such processes it often takes four to six weeks after receipt of the data to produce IC reports, and at least an additional 2 weeks for the payouts to be made. These applications create frequent disputes over sales credits and compensation payouts, which are time-consuming and costly to resolve and they create an atmosphere of distrust within the sales.

In addition, the sales model of a Pharma rep is a lot different from other Sales reps. Thinking from a Pharma sales rep's perspective, when he goes into an office and makes a sale, he does not hand over a prescription for the doctor to give to a patient. One has to count on them to write it after he is gone. So he does not get the satisfaction of making that sale every day like he would if he were a regular salesperson selling a product that he carried. He has to monitor that data later on. He does not reap his crop right there but has to wait until later to see how it turned out. So the lag time in letting him know his performance has to be reduced. Delivery of periodic reports to measure performance against goals will help sales force to find opportunities and also the payouts can be processed in a timely fashion.

Now let us take a look at the typical sales force hierarchy. As you could see, the sales performance is of interest to the whole team at their respective levels.

### Pharma Sales organization hierarchy



Level	Description
Organization	The top level of the sales organization that is sometimes referred to as <b>Franchise</b> . For example, <b>Allergy Franchise</b> refers to a sales force commissioned to promote allergy/immunology products. This level does not have any parents.
Sales Force	Child of Organization. Refers to a sales forces promoting products within a sales organization.
Area	Child of Sales Force. Refers to a large geographic sales area, such as US East, US West, or US South.
Region	Child of Area. Refers to a sales geographic region.
District	Child of Region. Refers to a sales geographic district.
Territory	Child of District. Refers to a sales geographic territory composed of ZIP codes, bricks, contacts, or accounts. Sales reps are assigned to the territories.

Figure 2. Pharmaceutical Sales Organization hierarchy

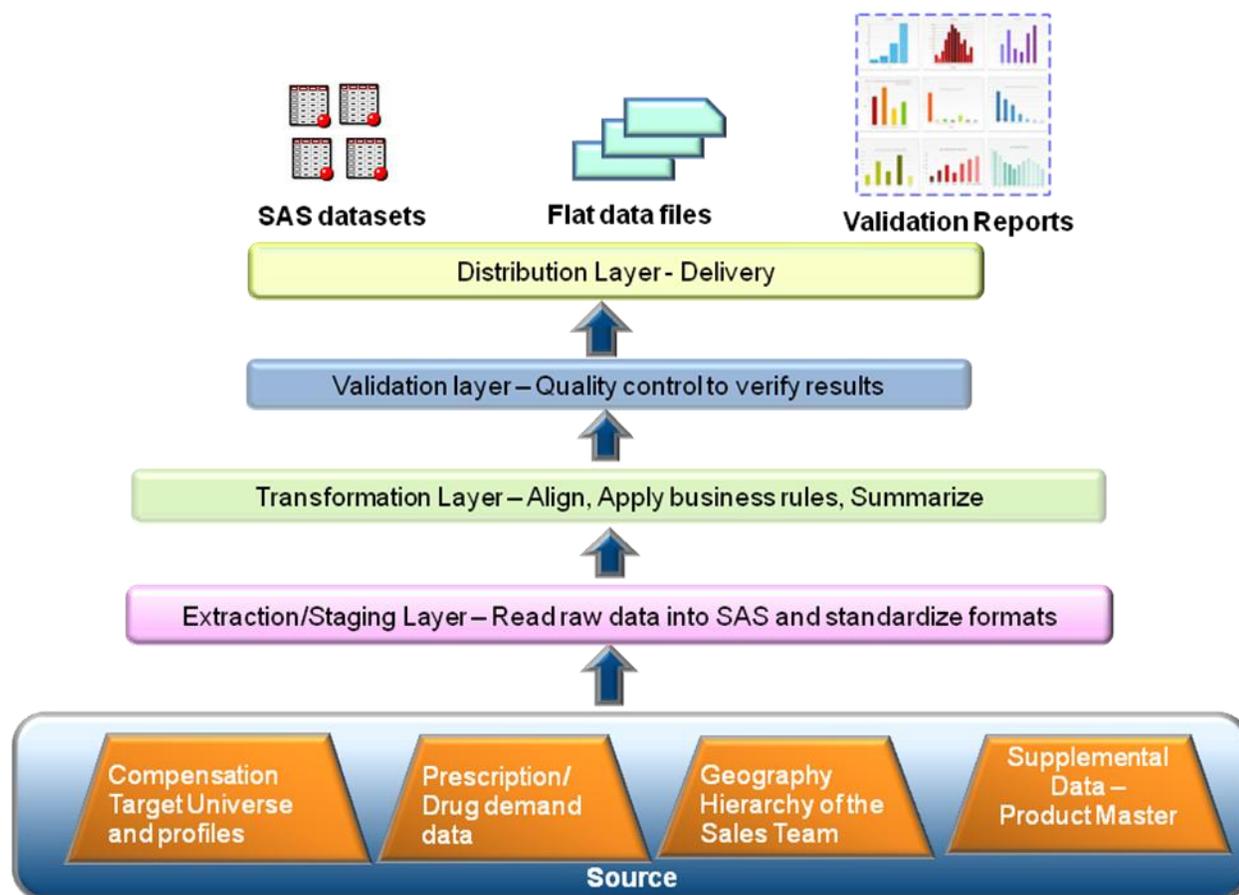
#### PRE-REQUISITES:

We are going to assume that the Pharmaceutical/Biotech Company purchases or leverages these pre-requisites available in-house, as applicable.

- Customer level prescription or Drug demand data (Dollars or Units of Sales) – Various providers (most notably, IMS Health) provide after-market sales information to the pharmaceutical industry. These firms track the flow of pharmaceutical products through the entire distribution network from the warehouse till the consumer. They also tabulate prescription volume at the physician level. Most pharmaceutical and biotech companies purchase this type of data.
- Aligned Compensation Universe/Target Customer – The sales rep will each be assigned to a territory and will have a call plan to call on a list of doctors in that territory of which the targets need to be visited more frequently in a 1-2 week cycle. His goal quotas will be set based on various factors including the prescribing capacity of the doctors defined in that territory and it is very important that the performance is measured against the same set of doctors for the selling period in order to be able to compare the goal Vs attainment.
- Physician profiles or Key Account profiles with nominal information for Physician specialty, Accounts classification like Clinics, hospitals, Nursing homes etc – The customer profiles are collected from the sales reps themselves based on the latest addresses they see the doctors at, his primary practice area (specialty of practice) etc. and also secured from the data vendors while purchasing sales information. This information may be used for applying business rules, exclusions etc based the customer profile.

- Hierarchy to derive higher geography levels for the Sales Team – The typical Pharmaceutical sales hierarchy is illustrated above. The relevant sales reports need to be provisioned to each level of hierarchy to monitor sales.
- Product master to identify Product Hierarchy that includes Product family, Therapeutic class, Market, sub-market definitions of home brands as well as the competition – to be able to calculate market shares if needed.

### Design overview of the SAS-based IC process:



**Figure 3. Framework of the SAS approach**

1. Load raw sales, customer universe, alignment and other feeds to SAS
2. Align Rx or dollars or units of sales to Physicians/Accounts in the Compensation Target Universe set of Customers
3. Align Sales team Hierarchy to their sales
4. Apply brand specific business rules
5. Summarize at required Geography level
6. Set goals and measure sales performance against same Compensation universe for the selling period with latest sales data.

For the most part in this paper, we will be referencing to the drug prescription sales which is the sell-out sales when drug is dispensed to the consumer. However the same procedures can be applied to drug sales into the health care facilities also called as accounts which is the sell-in data (dollars or units of sale).

## STEP BY STEP APPROACH TO THE PROPOSED SOLUTION:

### 1. EXTRACTION / STANDARDIZATION

The raw data from the source could potentially be in any structured or non-structured format. Given that the pieces required for IC essentially come from various cross-functional groups within the company and also purchased from third party vendor, the data need to be cleansed and standardized before use. SAS has multiple ways of reading such information one of which is IMPORT procedure. This procedure writes its own inbuilt 'infile' 'format' 'informat' and 'input' statements and takes care of the formatting itself.

Here is how PROC IMPORT could be used to read in Profile information, aligned Target customers, hierarchy and other supporting files.

```

***Extraction***
*** Import Procedure to read in target list ***;
filename phys "../adhoc/sugi2012/raw/in/targets.csv";
proc import file = phys
  out = target_list (keep = Prescriber_id Territory_id Sales_team Specialty)
  dbms =csv replace;
  delimiter=',';
  GETNAMES = yes; *Get SAS variable names from the data values in the first row
  of the file;
  DATAROW=2;
  guessingrows=50000; *To avoid data truncation;

run;

*** Import Procedure to read in Product Master ***;
filename prod "../adhoc/sugi2012/raw/in/product_master.csv";
proc import file = prod
  out = prod_master (keep = source_report_num source_prod_grp prod mkt tc)
  dbms =csv replace;
  delimiter=',';
  GETNAMES = yes;
  DATAROW=2;
  guessingrows=1000;

run;

*** Import Procedure to read in hierarchy text file ***;
filename geo_hier "../adhoc/sugi2012/raw/in/geo_hierarchy.txt";
data keep.geo_hierarchy_&cycle.;
infile geo_hier dlm='09'x dsd lrecl=4096 truncover firstobs=2 ;
length
  terr $6.
  district $5.
  region $3.
;
input
  terr $
  district $
  region $
;

run;

****Read Specialty filter ***;
data specialty_filter;
input prod $1 - 10 specialty $12 - 32;
cards;
PYREL allergy
SORUX immunology
product1 pediatrics
product2 oncology
product3 immunology;

run;

```

Often the crude data may need to be checked for duplicates for example the same prescriber could be aligned to the same territory twice which may result in double counting of the associates sales. Similarly any geography identifier is unique to a sales force (like employee is). Use SORT procedure to cleanse and validate using nodupkey option.

```

*** Remove duplicates if same prescriber territory combination appears twice ***;
proc sort data = target_list_adj (drop = Prescriber_id) nodupkey
  out = keep.target_list_&cycle.(rename = (Presc_id = Prescriber_id
  Territory_id = Territory ))
  dupout = wrk.reject_target_list;
  by Presc_id Territory_id;

run;

*** Check for duplicates if same territory_id appears for more than one sales team ***;
proc sort data = target_list_adj nodupkey
  out = chk2 ;
  by Territory_id Sales_team;

run;

```

Also as we go up in the hierarchy there should be no customer duplication at the level of reporting. For example the same customer could be called by two different sales force reps for different products both reporting to the same District Manager. In such a scenario the customer data needs to be included only once at the district aggregate level and not duplicated.

```

***Using Geography hierarchy to build customer universe at higher levels of sales team
***;
*Read Hierarchy mapping and make sure no duplicates by territories, the lowest hierarchy
level;
proc sort data=keep.geo_hierarchy_&cycle. nodupkey
  out=hier (keep=terr district region);
  by terr;

run;

proc sort data=keep.target_list_&cycle.
  out=mp_presc_terr;
  by territory;

run;

data keep.mp_presc_territory;
merge mp_presc_terr (in=a) hier (in=b rename = (terr = territory));
by territory;
if a=1 and b=1;
*derive new variable area using substring function on the territory variable
extracting first 2 characters only;
area=substr(territory,1,2);

run;

data mp_presc_hier;
set keep.mp_presc_territory;
* The sales teams are re-designated to one Field level manager and hierarchies
above;
if index(sales_team, "ALLERGY") > 0 or
index(sales_team, "INFLAMMATION") > 0 then
sales_team='IMMUNOLOGY_BU';

run;

*macro to create higher hierarchies by handling overlaying customers shared across
geographies;

%macro lvls(lvl);
proc sort nodupkey data=mp_presc_hier
out=keep.mp_presc_&lvl (keep=&lvl prescriber_id sales_team specialty);
by &lvl prescriber_id sales_team; run;
%mend lvls;

```

```
*call macro for any Geo level;
%lvls(district);%lvls(region);%lvls(area);
```

Importing the raw prescription sales data, quality control, summarization and standardization can all be encompassed inside a SAS macro and can be used repeatedly for multiple files across therapeutic classes. Please see an example of how this can be accomplished.

```
** Read Prescription sales raw data from thrid party vendor ***;

%macro      runit(cli,file,rpt,tc,mkt);
filename   vpdata "../adhoc/sugi2012/raw/in/sales.C&cli.R&file..001";

data       SASloadRaw;
infile     vpdata
dsd        delimiter = '~'
lrecl =    1200 missover;
length
  client_num          $      3
  s_report_num        $      2
  s_prod_grp          $      3
  prescriber_id       $      5
  <define length of other variables>
;

input
  client_num          $
  presc_id            $
  spec                $
  data_date           $
  raw_nrx1-raw_nrx24
  raw_trx1-raw_trx24
  <input other variables>
;

if _error_ > 0 then abort abend;
prescriber_id = substr(presc_id,1,5);
spec = upcase(spec);

*Set up rpt_prod key for Product Merge;
s_report_num = put(&rpt.,z2.) ;
s_prod_grp = put(source_prod_grp,z3.) ;
length rpt_prod $ 5 ;
rpt_prod = trim(s_report_num) || trim(s_prod_grp);

*Merge in Product Info;
set prod_master (keep = rpt_prod report_num prod_grp prod mkt tc)
key = rpt_prod / unique ;
if _iorc_ = 0 then do; /* Match found */
end;

run;

***QC - Reject Records with blanks in key columns***;
Data       SASload wrk.SASload_reject_&tc._&mkt ;
set sasloadraw;
if tc="" or prod = "" or dstrb_chnl = "" or prescriber_id = ""
then output wrk.SASload_reject_&tc._&mkt; ***Send rejects to the source;
else output SASload;

run;
```

```

***Data transformation - Rounding ****
Data      Prescriber;
          set sasload;
          array raw_trx(24) raw_trx1-raw_trx24 ;
          array trx(24)   trx1-trx24 ;
          <define array for all Rx variables>

          /* Numeric precision to 2 decimal places*/
          do i = 1 to 24 ;
              trx(i) = round(raw_trx(i), .01);
              <round other variables>
          end ;

run;

***Summarize and save as SAS data set***;
proc      summary nway data = prescriber;
          by prescriber_id ;
          class dstrb_chnl tc mkt prod;
          var nrx1-nrx24 trx1-trx24 ;
          output out = wrk.prescriber_&tc.(index = (prescriber_id)
          drop= _type_ _freq_ compress=binary reuse=yes)
          sum = ;

run;

%mend runit;

*Call the macro to import, transform raw sales file for any market;
%runit(567, 61, 61, allergy, allergy);

```

Now that we have the sales information and also the doctors list aligned to territories, we can now map the sales to doctors and hence territories.

## 2. APPLICATION OF BUSINESS RULES/TRANSFORMATION:

Tagging the sales information to aligned customers is built into a generic SAS macro. This macro can be used for any products and across any field forces (sales teams) – so essentially the same code can be extended to any number of products across various therapeutic classes and sales teams too. This way we tap the capabilities of Macros to re-use the same logic and conditionally execute certain data steps and procedures.

As we saw in the Sales force hierarchy, managers at various levels of the hierarchy will need a different level of summary which pertains to them. For example for the reports sent to the Sales reps, a territory level report of all the brands they call physicians on will give a good insight of how their territory is performing. Similarly for reports sent to Field Level manager, Regional Directors and Operations Unit managers need to be at the respective levels of reporting to provide a bird's eye view of performance in their sales zones.

SQL and SUMMARY procedures can be used to roll up to the level of reporting needed by aligning the physician universe to respective geographic hierarchies and summarize at the desired geography level.

We can customize the way the data is to be reported based on business needs. Data transformations like creation of new variables, calculations with existing variables, filtering of data, extract and concatenate strings etc. can be performed from within SAS

- For example if a particular sales force is expected to co-promote two brands in a therapeutic class, we can combine those products and create a customize product bucket and report their sales performance together. Create a new product definition column in the SAS DATA STEP and do a PROC SUMMARY by the new product definition.
- A drug is approved by the regulatory agency for specific indications and in certain cases, for specific age groups only; if so, selling the drug to a customer who treats a non-indicated disease condition or a non-approved age group of patients is not appropriate. To satisfy the legal requirements, off-label promotion is to be discouraged among the sales groups. So performance is measured for only those prescribers of approved specialty of practice by applying a filter for those specialties in a DATA step. For example prescription written by a Pediatrician cannot

be counted for performance reporting of a product not approved in children by FDA. In such a case, those sales generated by pediatricians will have to be eliminated. Please note that target customers should be all of approved specialty designations only, in the first place.

```

***   Align Prescriber and Rxs   ***;
%macro   align(lvl,sf);
%macro   tcs(tc);

*Physician Rxs;
proc      sql ;
create    view vr as
select    mp.&lvl, mp.sales_team, tc.prod, prod.mkt, prod.tc, tc.dstrb_chnl, specialty,
          trx1, trx2, trx3, trx4, trx5, trx6, trx7,trx8,trx9,trx10,trx11,trx12,
          nrx1, nrx2, nrx3, nrx4, nrx5, nrx6, nrx7, nrx8, nrx9, nrx10, nrx11, nrx12
          from keep.mp_presc_&lvl mp, wrk.prescriber_&tc. tc,prod_master prod
where     mp.sales_team = "&sf." and mp.prescriber_id = tc.prescriber_id and
          prod.prod=tc.prod;
quit;

Data     wrk.&tc._&lvl.(compress=binary);
set      vr;
array    trx(12) trx1-trx12;
         <Define array for required variables>
if dstrb_chnl='01' then do;
         do i=1 to 12;
           monrx(i)=0;
           motrx(i)=0;
           ltcnrx(i)=0;
           ltctrx(i)=0;
         end;
end;
if dstrb_chnl='02' then do;
         do i=1 to 12;
           monrx(i)=nrx(i);
           motrx(i)=trx(i);
           ltcnrx(i)=0;
           ltctrx(i)=0;
         end;
end;
if dstrb_chnl='03' then do;
         do i=1 to 12;
           monrx(i)=0;
           motrx(i)=0;
           ltcnrx(i)=nrx(i);
           ltctrx(i)=trx(i);
         end;
end;
drop i;

run;

proc      summary nway data = wrk.&tc._&lvl. ;
class    &lvl sales_team specialty prod mkt tc dstrb_chnl ;
var      <all Rx variables>;
output  out = tc_sum (drop = _type_ _freq_) sum = ;

run;

*Rounding off Prescriber Rxs;
data     round ;
set      tc_sum ;
array    trx(12)   trx1-trx12;

```

```

    <Define array for required variables>
do i = 1 to 12;
    rtrx(i) = round(trx(i),1);
    <Round required variables>
end;

*Co-promoted products are custom combined to one product;
If prod in ("PYREL" "SORUX") then prod = "PYRUX";

run;

proc summary nway data = round ;
class &lvl sales_team specialty prod mkt tc;
var nrx1-nrx12 rtrx1-rtrx12 rmonrx1-rmonrx12 rmotrx1-rmotrx12 rltcnrx1-
rltcnrx12 rltctrx1-rltctrx12;
output out = &tc. (drop = _type_ _freq_)
sum = nrx1-nrx12 trx1-trx12 monrx1-monrx12 motrx1-motrx12 ltcnrx1-ltcnrx12
ltctrx1-ltctrx12;

run;

* Include approved physician specialties only*;
proc sql ;
create table wrk.tgt_&lvl._&sf. as
select a.*
from &tc. a, specialty_filter b
where a.prod = upcase(b.prod) and a.specialty = upcase(b.specialty);

quit;

proc delete data=&tc.; run; /*Good practice to delete intermediate data sets after use */
%mend tcs;

/* Call this macro to compute summaries for all the products/Therapeutic classes that is
called on by a field force */
%tcs(allergy);
%mend align;

/* Call this macro to compute summaries for all the field forces and required level of
geo hierarchy */
%align(territory,ALLERGY_TEAM);
%align(district,IMMUNOLOGY_BU);

```

### 3. VALIDATION / PRESENTATION / ANALYSIS

A number of Validations can be accomplished using SAS to Quality Assure the reported data. Here are a couple of must do validations for uniqueness and non-null values.

```

***** QC checks on the final files *****;
%macro go_geo(lvl,sf);
data &lvl._&sf._chk;
set wrk.tgt_&lvl._&sf.;
*Null check for key variables - none expected if the data is good;
if &lvl. = "" or specialty = "" or prod = "" or sales_team = "";

run;

proc sort data = wrk.tgt_&lvl._&sf. nodupkey
out = chk1 dupout = chk2;
by &lvl sales_team specialty prod; * uniqueness check by key variables;

run;
%mend go_geo;

%go_geo(territory,ALLERGY_TEAM);
%go_geo(district,IMMUNOLOGY_BU);

```

We can build comparison reports to see variation from month to month across territories and across brands, compare goals with the previous cycle. Use EXPORT procedure to output to Microsoft Excel or CSV file

```

***** QC Current files vs Previous files ***;

%macro   compare_sum(lvl,sf);
%let     checksf="%sf.";
%let     checklvl="%lvl.";
data     curr;
set      wrk.tgt_&lvl._&sf.;
where    prod in ('PYRUX');
         array   trx(12)   trx1-trx12;
         array   currtrx(12) currtrx1-currtrx12;
         do i=1 to 6;
           currtrx(i)=trx(i+1);
         end;
         keep   &lvl prod currtrx1-currtrx6;
run;

proc     summary nway data=curr;
class   &lvl prod;
var     currtrx1-currtrx6 ;
output  out=curr (drop=_type_ _freq_) sum=;
run;

data     prev;
set      wrk.tgt_&lvl._&sf.;
where    prod in ('PYRUX');
         <Define array for required variables>
         do i=1 to 6;
           prevtrx(i)=trx(i);
         end;
         keep   &lvl prod prevtrx1-prevtrx6;
run;

proc     summary nway data=prev;
class   &lvl prod;
var     prevtrx1-prevtrx6 ;
output  out=prev (drop=_type_ _freq_) sum=;
run;

data     curr_v_prev ;
merge   curr prev ;
by      &lvl prod;
<define required arrays for Rxs>
do i=1 to 6;
         if prevtrx(i)=0 then prctdifftrx(i)=0;
         if prevtrx(i) ne 0 then prctdifftrx(i)=ABS((currtrx(i)-
           prevtrx(i))/prevtrx(i));
         difftrx(i)=currtrx(i)-prevtrx(i);
       end;
max_diff= MAX(prctdifftrx1, prctdifftrx2,
prctdifftrx3,prctdifftrx4,prctdifftrx5,prctdifftrx6);
drop i prctdifftrx1-prctdifftrx6;
run;

proc     sort data=curr_v_prev ; by descending max_diff; run;

proc     export data=curr_v_prev
outfile="../adhoc/sugi2012/raw/out/curr_v_prev_summ_&sf._&lvl._tgt.csv"
replace;
run;
%mend   compare_sum;

```

```
%compare_sum(territory,ALLERGY_TEAM);
%compare_sum(district,IMMUNOLOGY_BU);
```

Here is how the comparison report will look like. It shows the Rx count that was reported previous month and being reported in the current month for each territory in this case. We can also improvise it to spit out the exception report if the differences exceed a pre-defined threshold.

Territory	Prod	currtrx1	currtrx2	currtrx3	currtrx4	currtrx5	currtrx6	prevtrx1	prevtrx2	prevtrx3	prevtrx4	prevtrx5	prevtrx6	difftrx1	difftrx2	difftrx3	difftrx4	difftrx5	difftrx6	max_diff
K12C3G	PYRUX	0	0	0	0	1	4	0	0	0	0	0	1	0	0	0	0	1	3	3
K12P3A	PYRUX	3	1	3	3	6	5	1	3	1	3	3	6	2	-2	2	0	3	-1	2
K12F3F	PYRUX	5	3	8	8	3	6	3	5	3	8	8	3	2	-2	5	0	-5	3	1.666666667
K12F3C	PYRUX	5	2	5	6	3	7	9	5	2	5	6	3	-4	-3	3	1	-3	4	1.5
K12C3J	PYRUX	4	3	4	2	1	0	9	4	3	4	2	1	-5	-1	1	-2	-1	-1	1
K12F3H	PYRUX	1	1	1	2	3	1	2	1	1	1	2	3	-1	0	0	1	1	-2	1
K12P3C	PYRUX	2	0	0	0	1	0	2	2	0	0	0	1	0	-2	0	0	1	-1	1
K12C3F	PYRUX	9	17	16	10	12	11	13	9	17	16	10	12	-4	8	-1	-6	2	-1	0.888888889
K12F3D	PYRUX	37	30	25	44	35	33	50	37	30	25	44	35	-13	-7	-5	19	-9	-2	0.76
K12D3H	PYRUX	7	9	8	8	5	8	6	7	9	8	8	5	1	2	-1	0	-3	3	0.6
K12D3J	PYRUX	5	8	6	9	8	4	8	5	8	6	9	8	-3	3	-2	3	-1	-4	0.6
K12E3F	PYRUX	5	6	5	8	6	4	8	5	6	5	8	6	-3	1	-1	3	-2	-2	0.6
K12E3B	PYRUX	40	41	29	46	28	33	37	40	41	29	46	28	3	1	-12	17	-18	5	0.586206897
K12D3G	PYRUX	9	14	16	17	12	11	7	9	14	16	17	12	2	5	2	1	-5	-1	0.555555556
K12C3D	PYRUX	21	15	23	19	18	17	19	21	15	23	19	18	2	-6	8	-4	-1	-1	0.533333333
K12A3G	PYRUX	27	28	25	36	17	22	24	27	28	25	36	17	3	1	-3	11	-19	5	0.527777778
K12A3B	PYRUX	31	22	33	36	27	21	31	31	22	33	36	27	0	-9	11	3	-9	-6	0.5
K12A3D	PYRUX	11	10	15	12	10	13	14	11	10	15	12	10	-3	-1	5	-3	-2	3	0.5
K12D3E	PYRUX	35	39	33	40	38	19	54	35	39	33	40	38	-19	4	-6	7	-2	-19	0.5
K12A3A	PYRUX	25	37	43	25	27	19	37	25	37	43	25	27	-12	12	6	-18	2	-8	0.48
K12A3F	PYRUX	25	33	30	21	31	24	26	25	33	30	21	31	-1	8	-3	-9	10	-7	0.476190476
K12F3E	PYRUX	21	12	14	14	15	16	24	21	12	14	14	15	-3	-9	2	0	1	1	0.428571429
K12D3A	PYRUX	24	30	31	35	39	23	28	24	30	31	35	39	-4	6	1	4	4	-16	0.41025641
K12B3D	PYRUX	33	29	24	33	25	22	40	33	29	24	33	25	-7	-4	-5	9	-8	-3	0.375

Figure 4. Sample validation report

The presentation of data is an essential part of virtually every study and there are a number of tools within SAS that allow the user to create a large variety of charts, reports, and data summaries. REPORT procedure is particularly powerful and valuable that can be used in this process. It can be used to both summarize and display data, and is highly customizable and highly flexible. We can report key metrics for business validation using this procedure. Such reports provide a quick view of the validity of the numbers and easy to review for non-technical users too.

```
***** PROC REPORT*****;

%macro go_report1(lvl,sf);

    title1 'Using Proc REPORT for validation outputs';
    title2 'Summary by product for each Therapeutic class for the recent 4 months trend';

proc report data=wrk.tgt_&lvl._&sf. ;
    columns prod tc trx1- trx4;
    define tc / group 'Therapeutic Class*---' ;
    define prod / group 'Product*---' ;
    define trx1- trx4 / analysis sum;

run;
%mend go_report1;
%go_report1(territory,ALLERGY_TEAM);
```

Using Proc REPORT for validation outputs					
Summary by product in the Therapeutic class for the recent 4 months trend					
10:15 Wednesday, March 14					
Product*---	Therapeutic Class*---	trx1	trx2	trx3	trx4
PYRUX	IMMUNOSUPPRESSANT	1919	1706	1648	1648

Figure 5.a. Sample report output

```

%macro go_report2(lvl,sf);

    title1 'Using Proc REPORT for validation outputs';
    title2 'How does Total prescription sales look across the Physician specialty
    group for the recent 6 months';

proc report data=wrk.tgt_&lvl._&sf. headline headskip;
    columns sales_team specialty trx1- trx6;
    define sales_team / group width = 15;
    define specialty / group width = 15;
    define trx1- trx6 / analysis sum format = 8.2;
run;
%mend go_report2;

%go_report2(territory,ALLERGY_TEAM);

```

Using Proc REPORT for validation outputs								9
How does Total Rx sales look across the Physician specialty group for the recent 6 months								
								10:15 Wednesday, March 14, 2012
Sales_team	Specialty	trx1	trx2	trx3	trx4	trx5	trx6	
ALLERGY_TEAM	ALLERGY	1919.00	1706.00	1648.00	1648.00	1750.00	1576.00	

Figure 5.b. Sample report output

We can also visually represent the sales trend across the reporting period for the brands of interest using GPLOT procedure. Any unexpected drop or spike in the sales which cannot be correlated to a market event may be a potential data issue needing a closer look.

```

***** PROC GPLOT*****;

%macro go_plot1(lvl,sf);

    *Summarize to product level;
proc summary nway data=wrk.tgt_&lvl._&sf.;
    class prod;
    var nrx1-nrx12 trx1-trx12 ;
    output out=curr (drop=_type_ _freq_) sum=;
run;

data curr;
    length flag $30.;
    set curr;
    Sales_force = upcase(left(trim("&lvl."||"_"||"&sf.")));
run;

data temp1 (rename = (nrx1 - nrx12 = mnth1 - mnth12));
    set curr;
    metric_flag = "NRx";
    keep prod Sales_force metric_flag nrx1 - nrx12;
run;

data temp2 (rename = (trx1 - trx12 =mnth1 - mnth12));
    set curr;
    metric_flag = "TRx";
    keep prod Sales_force metric_flag trx1 - trx12;
run;

data final_concat;
    set temp1 temp2 ;
    keep prod Sales_force metric_flag mnth1 - mnth12;

```

```

run;

proc      sort data = final_concat ;
          by prod Sales_force metric_flag; run;

*Transpose to normalize;
proc      transpose data = final_concat out = tempx  prefix = Metrics  ;
          by prod Sales_force metric_flag ;
          var mnth1-mnth12;

run;

* Set the graphics environment ;
goptions  reset=all border cback=white htitle=12pt htext=10pt  horigin=1 cm vorigin=1
          cm vsize=10 cm hsize=16 cm ;
          dm log 'clear';
          dm output 'clear';

proc      sql noprint;
select    quote(_name_) into :nam separated by " " from tempx; quit;

*Define axis characteristics;
axis1     label=('Metrics' justify=right 'Scripts') ;
axis2     label=('Month') order=(&nam.) ;

          ODS PDF FILE = ../adhoc/sugi2012/raw/out/Validation_Chart_&lvl._&sf..pdf";
          GOPTIONS DEVICE=pdf ;
          ods pdf startpage=no;

%macro    plot(met1,met2,prod);

* Define titles and footnotes;
          title1"Metrics Trend for - '&prod.' ";
          footnote1 j=1 " Data Month: &month.&year.";
          footnote2 j=1 " Incentive Cycle: &cycle.";
          footnote3 ' ';

proc      gplot data= tempx ;
          symbol i=join v=star h=1; *Define symbol characteristics ;
          plot metrics1*_name_ = metric_flag /vaxis=axis1 haxis=axis2 hminor =0;
          where metric_flag in ("&met1." "&met2.") and prod = "&prod." ;by Sales_force ;
          run;

quit;

%mend plot;

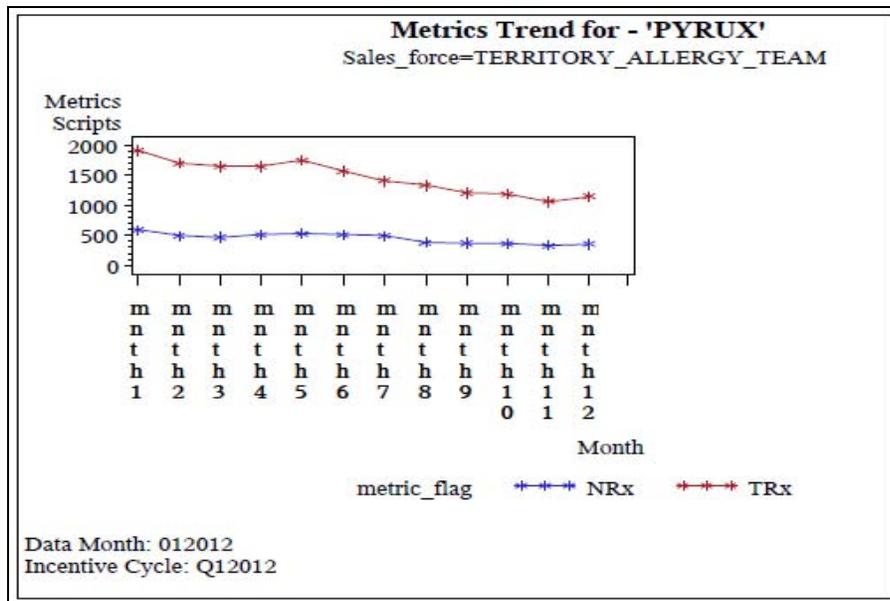
%plot(NRx,TRx,PYRUX);

ODS PDF CLOSE;

%mend go_plot1;

%go_plot1(territory,ALLERGY_TEAM);
%go_plot1(district,IMMUNOLOGY_BU);

```



**Figure 6. Sample graphical validation report**

Export the data to flat files for further Field reporting on performance, Payout processing, analytics etc. Use DATA \_NULL\_ step to write to a delimited text file using PUT statement to position the output variables with @.

The outputs can be fed to downstream applications for field reporting and also performance payout system which calculates the payouts based on Goal Vs attainment for each territory. Such systems are maintained in-house or by a third party and are strictly regulated for compliance to adopt an ethical approach in calculating the payouts and are rigid about their input file formats too. The below code will produce the formatted file month over month, consistently.

```
***** Export to flat files for downstream processing *****;
%macro      export_tgt (lvl,sf);
  data _null_ ;
  set wrk.tgt_&lvl. &sf.;
  operating_unit=substr(territory,1,1);
  file "../adhoc/sugi2012/raw/out/terr_&sf._targets.DAT" lrecl=800 ;
  array trx(12) trx1-trx12 ;
  <define more arrays>
    put @1 sales_team $20.
      @22 specialty $10.
      @34 Operating_unit $1.
      @36 territory $6.
      @43 mkt $20.
      @65 prod $20. @ ;
    do i = 1 to 12 ;
      j = 87 + (i-1) * 26 ;
      put @j nrx(i) 13. trx(i) 13. @ ;
    end ;
    do i = 1 to 12 ;
      j = 400 + (i-1) * 26 ;
      put @j nun_(i) 13. tun_(i) 13. @ ;
    end ;
  put @714 tc $20.
    @736 "TARGET" ;

run;
%mend      export_tgt ;

%export_tgt(territory,ALLERGY_TEAM);
```

Team	Product Code	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6	Value 7	Value 8	Value 9
ALLERGY_TEAM	K K12A3A	9	37	5	25	11				
ALLERGY_TEAM	K K12A3B	15	31	11	31	6				
ALLERGY_TEAM	K K12A3C	6	24	8	29	5				
ALLERGY_TEAM	K K12A3D	3	14	6	11	2				
ALLERGY_TEAM	K K12A3E	38	151	36	137	27				
ALLERGY_TEAM	K K12A3F	5	26	9	25	10				
ALLERGY_TEAM	K K12A3G	4	24	4	27	5				
ALLERGY_TEAM	K K12B3A	7	38	10	43	10				
ALLERGY_TEAM	K K12B3B	11	53	9	42	13				
ALLERGY_TEAM	K K12B3C	18	49	11	44	14				
ALLERGY_TEAM	K K12B3D	8	40	7	33	9				
ALLERGY_TEAM	K K12B3E	6	27	8	25	9				
ALLERGY_TEAM	K K12B3F	22	64	17	59	13				
ALLERGY_TEAM	K K12C3A	63	139	59	130	40				
ALLERGY_TEAM	K K12C3B	18	69	14	54	14				
ALLERGY_TEAM	K K12C3C	14	37	4	30	4				
ALLERGY_TEAM	K K12C3D	4	19	5	21	4				
ALLERGY_TEAM	K K12C3E	22	140	23	139	26				
ALLERGY_TEAM	K K12C3F	1	13	3	9	5				
ALLERGY_TEAM	K K12C3G	0	0	0	0	0				
ALLERGY_TEAM	K K12C3H	10	28	2	27	7				
ALLERGY_TEAM	K K12C3J	3	9	1	4	0				

Figure 7. Sample flat file output

Using SAS/CONNECT, the whole process can be run in a remote server where the data resides and the job runs faster than locally. Just encompass the whole SAS code starting with `rsubmit;` and end with `endsubmit;`. The code is all set to work for you remotely!

Please note that the code snippets in this paper are provided for illustration purposes only and may not be the exact code. Please contact the author for further information.

## CONCLUSION

As each pharmaceutical company is different in their performance compensation design and strategy there may be more than one approach to it. Excerpts from one such case study is presented here, assuming a traditional way using the historical sales data to set quotas and compensate the sales representatives. However this can form the basis for developing an extensible SAS based solution to accommodate a multitude of other business needs like apportioning of sales for shared territories, bundled products with different weight for promotion, compensate based on Market Share of the drug as opposed to the Volume of sales etc. Also field force modifications like call plan changes, re-organizations, hierarchy changes etc can be accommodated with some tweaks to the process, as applicable. I hope this paper gave an overview of some of the related concepts and was thought provoking to help you approach a similar requirement using SAS.

## ACKNOWLEDGMENTS

The author would like to thank all who helped her understand the presented concepts, her managers, colleagues, customers and family for their support and encouragement. She also would like to thank SAS Global forum and Cognizant Technology solutions for this opportunity to present the paper.

## CONTACT INFORMATION

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

Ramya Purushothaman  
 Cognizant Technology Solutions  
 500 Frank W. Burr Blvd.  
 Teaneck, NJ 07666  
[Ramya.Purushothaman@Cognizant.com](mailto:Ramya.Purushothaman@Cognizant.com)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration. Other brand and product names are trademarks of their respective companies.