
UTHSC

PROGRAM NAME :Medication Utilization Issues - Medicare Star Ratings Program

PROJECT NAME :Finding Solutions for Disparities Associated with Star Ratings

DESCRIPTION :Calculate the following performance metrics to evaluate racial/ethnic disparities associated with Medicare Star Ratings Program -

- a. Proportion of Days Covered (PDC) Adherence Measure: Diabetes (PDC-DR)
- b. PDC Adherence Measure: Renin Angiotensin System Antagonists (PDC-RASA)
- c. PDC Adherence Measure: Statins (PDC-STA)
- d. PDC Adherence Measure: Beta-Blockers (PDC-BB)
- e. PDC Adherence Measure: Calcium Channel Blockers (PDC-CBB)
- f. PDC Adherence Measure: Biguanides (PDC-BG)
- g. PDC Adherence Measure: DPP-4 Inhibitors(PDC-DPP)
- h. PDC Adherence Measure: Sulfonylureas (PDC-SFU)
- i. PDC Adherence Measure: Thiazolidinediones (PDC-TZD)
- j. PDC Adherence Measure: Non-Warfarin Oral Anticoagulants (PDC-NOAC)
- k. PDC Adherence Measure: Long-Acting Inhaled Bronchodilator Agents in COPD Patients (PDC-COPD)
- l. PDC Adherence Measure: Antiretroviral Medications (PDC-ARV)
- m. PDC Adherence Measure: Non-Infused Disease-Modifying Agents Used to Treat Multiple Sclerosis (PDC-MS)
- n. Diabetes Medication Dosing (DOS)
- o. Statins Use in Person with Diabetes (SUPD)
- p. Cholesterol Management in Coronary Artery Disease (CMC)
- q. Drug-Drug Interaction (DDI)
- r. Antipsychotic Use in Person with Dementia (APD)
- s. Use of High-Risk Medications in the Elderly (HRM)
- t. Use of Benzodiazepine Sedative Hypnotics in the Elderly (BSH)

SOFTWARE VERSION :SAS Windows9.4

RELATED PROGRAMS :N/A

REQUIREMENTS:

Ver#	Author & Program History Description	Peer reviewer
------	--------------------------------------	---------------

01	Steve Tsang	production version of the program
----	-------------	-----------------------------------

```
*****  
***;
```

```
LIBNAME STAR "/sas/vrdc/users/*****/files/_uploads/STAR";  
/*User ID is censored throughout the document*/
```

```
*Take a 10% systematic random sample with patient IDs ending in zero;
```

```
data MBSFt;  
set MBSF.MBSF_ABCD_2017;  
where mod(bene_id,10)=0;  
keep bene_id;  
run;
```

```
proc sort data= MBSFt;by bene_id; run;  
proc sort data=MBSF_2017_6;by bene_id; run; /*file contains patient  
characteristics*/
```

```
data MBSFt_1;  
merge MBSFt (in=in1) MBSF_2017_6;  
by bene_id;  
if in1;  
run;
```

```
data rate17_exp;  
set MBSFt_1;  
keep bene_id esrd_ind hospice age_BOY age ssacounty;  
run;
```

```
data rate17_explist;  
set MBSFt_1;  
keep bene_id;  
run;
```

```
* Get prescription records from Part D Event Files;
```

```
data jj;  
set PDE2017.PDE_DEMO_2017_01  
PDE2017.PDE_DEMO_2017_02  
PDE2017.PDE_DEMO_2017_03  
PDE2017.PDE_DEMO_2017_04  
PDE2017.PDE_DEMO_2017_05  
PDE2017.PDE_DEMO_2017_06  
PDE2017.PDE_DEMO_2017_07  
PDE2017.PDE_DEMO_2017_08  
PDE2017.PDE_DEMO_2017_09  
PDE2017.PDE_DEMO_2017_10  
PDE2017.PDE_DEMO_2017_11  
PDE2017.PDE_DEMO_2017_12;  
where mod(bene_id,10)=0;  
ndc1=PROD_SRVC_ID*1;  
drop PROD_SRVC_ID;  
run;
```

```

proc sort data=jj; by bene_id; run;
proc sort data=ratel7_exp; by bene_id; run;

data aa;
merge ratel7_exp(in=in1) jj;
by bene_id;
if in1;
run;

proc sort data= aa; by ndc1; run;

/*****
*PDC Adherence Measure for Diabetes Medications (PDC-DR);
*****/

/*Inclusion criteria
1) 18 years and older
2) First prescription occurred at least 91 days before the end of the year
3) At least two prescriptions on two unique dates
4) Exclusion of Insulin Users
5) Exclusion of ESRD Population */

/*Determine inclusion outcomes*/

data diabetes1; set _UPLDS.diabetes; ndc1=ndc*1; drop ndc; run;

proc sort data=diabetes1; by ndc1; run;

data d;
merge aa (in=in1) diabetes1 (in=in2);
by ndc1;
if in1 and in2;
keep bene_id SRVC_DT diabetes DAYS_SUPPLY_NUM;
diabetes=1;
run;

* Two medications on different dates;

proc sort data=d out=d ; by bene_id ;
run;

proc sort nodupkey data=d out=da; by bene_id; run;

proc sort data=d out=d1 nodupkey; by bene_id SRVC_DT ;
run;
proc freq data=d1 noprint;
table bene_id/out=d2; run;

data d2a;
set d2;
if count>=2;
keep bene_id number_diabetes count;
number_diabetes=1;
run;

```

```

*91 days;
data d3;
set d1;
by bene_id;
if first.bene_id then output;
run;

data d4;
set d3;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_diabetes;
duration_diabetes=1;
run;

*Exclude insulin;
data insulin1; set _UPLDS.insulin; ndc1=ndc*1; drop ndc; run;
data dd;
merge aa(in=in1) insulin1(in=in2);
by ndc1;
if in1 and in2;
keep bene_id insulin;
insulin=1;
run;

proc sort data=dd out=dd1 nodupkey; by bene_id;
run;

* Record results for meeting inclusion criteria;

data aaa;
merge rate17_exp (in=in1) da d2a d4 dd1;
by bene_id;
if in1;
run;

data incdiab;
set aaa;
if diabetes=. then diabetes=0;
if number_diabetes=. then number_diabetes=0;
if duration_diabetes=. then duration_diabetes=0;
if insulin=. then insulin=0;
if age>=18 and number_diabetes=1 and duration_diabetes=1 and insulin=0 and
esrd_ind="0" and hospice=0 then include=1;
else include=0;
keep bene_id ESRD_IND HOSPICE diabetes number_diabetes duration_diabetes
insulin include age age_boy;
run;

/*Calculate PDC (Measure Assessment)*/

data diabetes_cx; set _UPLDS.diabetes2017_vxx; ndc1=ndc*1; drop ndc; run;

data diabetes_cx1;
set diabetes_cx;
if Albiglutide_Flag ='X' then label1=1; else label1=.;
if Chlorpropamide_Flag ='X' then label2=2; else label2=.;

```

```

if Dapagliflozin_Flag = 'X' then label3=3; else label3=.;
if Dulaglutide_Flag = 'X' then label4=4; else label4=.;
if Empagliflozin_Flag = 'X' then label5=5; else label5=.;
if Glimepiride_Flag = 'X' then label6=6; else label6=.;
if Glipizide_Flag = 'X' then label7=7; else label7=.;
if Glyburide_Flag = 'X' then label8=8; else label8=.;
if Metformin_Flag = 'X' then label9=9; else label9=.;
if Pioglitazone_Flag = 'X' then label10=10; else label10=.;
if Rosiglitazone_Flag = 'X' then label11=11; else label11=.;
if Tolazamide_Flag = 'X' then label12=12; else label12=.;
if Tolbutamide_Flag = 'X' then label13=13; else label13=.;
if Alogliptin_Flag = 'X' then label14=14; else label14=.;
if Sitagliptin_Flag = 'X' then label15=15; else label15=.;
if Linagliptin_Flag = 'X' then label16=16; else label16=.;
if Saxagliptin_Flag = 'X' then label17=17; else label17=.;
if Exenatide_Flag = 'X' then label18=18; else label18=.;
if Liraglutide_Flag = 'X' then label19=19; else label19=.;
if Lixisenatide_Flag = 'X' then label20=20; else label20=.;
if Nateglinide_Flag = 'X' then label21=21; else label21=.;
if Repaglinide_Flag = 'X' then label22=22; else label22=.;
if Canagliflozin_Flag = 'X' then label23=23; else label23=.;
if Semaglutide_Flag = 'X' then label24=24; else label24=.;

```

```

keep ndc1 label1-label24;
run;

```

```

proc sort data=diabetes_cx1; by ndc1; run;

```

```

proc transpose data=diabetes_cx1 out=diabetes_cx2;
  by ndc1;
run;

```

```

data diabetes_cx3;
set diabetes_cx2;
labels=coll;
drop _name_ coll;
if labels ne .;
run;

```

```

proc sql;
create table want as
select aa.*, diabetes_cx3.labels
from aa, diabetes_cx3
where aa.ndc1 = diabetes_cx3.ndc1;
quit;

```

```

data want1;
set want;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM;
run;

```

```

proc sort data=want1; by bene_id labels SRVC_DT ; run;

```

```

data want2;
do until (last.labels);
  set want1 ;

```

```

    by bene_id labels;
    SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;
    output ;
end;
format SRVC_END DATE9. ;
run;

data want3;
set want2;
by bene_id labels ;
format adjStart adjEnd lastdate mmddyy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd   = SRVC_DT + DAYS_SUPLY_NUM -1;
    if first.labels then lastdate = .;
    else do ;
** If overlap then adjust start/end dates, else keep as is;
    if adjStart <= lastdate then do;
        adjStart = lastdate + 1;
        adjEnd   = adjStart + DAYS_SUPLY_NUM -1 ;
    end;
end;
** Save last ending date;
lastdate = adjEnd;
run;

proc sort data=want3 out=want4 nodupkey; by bene_id adjStart descending
DAYS_SUPLY_NUM;
run;

proc sort data=want4 out=d_dia nodupkey;
by bene_id adjStart;
run;

data aa_d1; set d_dia;
member_id=bene_id;
fill_dt=adjStart;
format fill_dt DDMMYY10. ;
days_supply=DAYS_SUPLY_NUM;
run;

data aa_d2; set aa_d1;
keep member_id fill_dt days_supply; run;

data claims_diabetes; set aa_d2;
run;

*Step 1;

proc sort data=claims_diabetes; by member_id fill_dt;
run;

proc transpose data=claims_diabetes out=fill_dates (drop=_name_)
prefix=fill_dt;
by member_id;
var fill_dt;
run;

```

```

proc transpose data=claims_diabetes out=days_supply (drop=_name_)
prefix=days_supply;
by member_id;
var days_supply;
run;

data both;
merge fill_dates days_supply;
by member_id;
format start_dt end_dt mmddyy10. ;
start_dt=fill_dt1;
end_dt='31DEC2017'd;
run;

*Step 2;
data pdc_diabetes;
set both;
array daydummy(365) day1-day365;
serv_days=end_dt-start_dt+1;
array filldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
    do i=1 to dim(filldates) while (filldates(i) ne .);
        if filldates(i)<=first_dt+ii-1<=filldates(i)+days_supply(i)-1
            then daydummy(ii)=1;
    end;
end;
drop i ii ;
dayscovered=sum(of daydummy[*]);
label dayscovered='Total Days Covered';
p_dayscovered=dayscovered/serv_days;
label p_dayscovered='Proportion of Days Covered';
drop fill_dt1-fill_dt365 days_supply1-days_supply365;
run;

data pdc_diabetes_1;
set pdc_diabetes;
bene_id=member_id;
pdc_diabetes = p_dayscovered;
sd_diabetes = serv_days;
dc_diabetes = dayscovered;
keep bene_id pdc_diabetes sd_diabetes dc_diabetes;
run;

proc sort data=pdc_diabetes_1; by bene_id;
run;

data incdiab_1;
merge incdiab (in=in1) pdc_diabetes_1;
by bene_id;
if in1;
run;

```

```

/*****/
*PDC Adherence Measure for Renin Angiotensin System Antagonists (RASA-
Hypertension Medications, PDC-RASA);
/*****/

/*Inclusion criteria
1) 18 years and older
2) First prescription occurred at least 91 days before the end of the year
3) At least two prescriptions on two unique dates
4) Exclusion of Sacubitril/Valsartan Users
5) Exclusion of ESRD Population*/

/*Determine inclusion outcomes*/
data ras1; set _UPLDS.rasa; ndc1=ndc*1; drop ndc; run;

proc sort data=ras1; by ndc1; run;

data d_hyp;
merge aa (in=in1) ras1 (in=in2);
by ndc1;
if in1 and in2;
keep bene_id SRVC_DT hyperten DAYS_SUPLY_NUM;
hyperten=1;
run;

* Two medications on different dates;
proc sort data=d_hyp out=d_hyp ; by bene_id ;
run;

proc sort nodupkey data=d_hyp out=d_hypa; by bene_id; run;

proc sort data=d_hyp out=d_hyp1 nodupkey; by bene_id SRVC_DT ;
run;
proc freq data=d_hyp1 noprint;
table bene_id/out=d_hyp2; run;

data d_hyp2a;
set d_hyp2;
if count>=2;
keep bene_id number_hyperten count;
number_hyperten=1;
run;

*91 days;
data d_hyp3;
set d_hyp1;
by bene_id;
if first.bene_id then output;
run;

data d_hyp4;
set d_hyp3;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_hyperten;
duration_hyperten=1;

```



```

run;

*Exclude sacubitril/valsartan;
data varsar2; set _UPLDS.varsar; ndc1=ndc*1; drop ndc; run;
proc sort data=varsar2; by ndc1; run;

data dd_hyp;
merge aa(in=in1) varsar2(in=in2);
by ndc1;
if in1 and in2;
keep bene_id varsar;
varsar=1;
run;

proc sort data=dd_hyp out=dd_hyp1 nodupkey; by bene_id;
run;

data newgroup; set rate17_exp; keep bene_id; run;

* Record inclusion outcomes;
data aaa_hyp;
merge newgroup (in=in1) d_hypa d_hyp2a d_hyp4 dd_hyp1;
by bene_id;
if in1;
run;

data increasa;
set aaa_hyp;
if hyperten=. then hyperten=0;
if number_hyperten=. then number_hyperten=0;
if duration_hyperten=. then duration_hyperten=0;
if varsar=. then varsar=0;
keep bene_id hyperten number_hyperten duration_hyperten varsar ;
run;

/*Calculate PDC (Measure Assessment)*/

data rasa_cx; set _UPLDS.rasa2017_vxx; ndc1=ndc*1; drop ndc; run;

data rasa_cx1;
set rasa_cx;
if Aliskiren_Flag = 'X' then label1=1; else label1=.;
if Azilsartan_Flag = 'X' then label2=2; else label2=.;
if Benazepril_Flag = 'X' then label3=3; else label3=.;
if Candesartan_Flag = 'X' then label4=4; else label4=.;
if Captopril_Flag = 'X' then label5=5; else label5=.;
if Enalapril_Flag = 'X' then label6=6; else label6=.;
if Eprosartan_Flag = 'X' then label7=7; else label7=.;
if Fosinopril_Flag = 'X' then label8=8; else label8=.;
if Irbesartan_Flag = 'X' then label9=9; else label9=.;
if Lisinopril_Flag = 'X' then label10=10; else label10=.;
if Losartan_Flag = 'X' then label11=11; else label11=.;
if Moexipril_Flag = 'X' then label12=12; else label12=.;
if Olmesartan_Flag = 'X' then label13=13; else label13=.;
if Perindopril_Flag = 'X' then label14=14; else label14=.;
if Quinapril_Flag = 'X' then label15=15; else label15=.;
if Ramipril_Flag = 'X' then label16=16; else label16=.;

```

```

if Telmisartan_Flag ='X'      then label17=17; else label17=.;
if Trandolapril_Flag ='X'    then label18=18; else label18=.;
if Valsartan_Flag ='X'      then label19=19; else label19=.;

keep ndc1 label1-label19;
run;

proc sort data=rasa_cx1; by ndc1; run;

proc transpose data=rasa_cx1 out=rasa_cx2;
  by ndc1;
run;

data rasa_cx3;
set rasa_cx2;
labels=coll1;
drop _name_ coll1;
if labels ne .;
run;

proc sql;
create table want as
select aa.*, rasa_cx3.labels
from aa, rasa_cx3
where aa.ndc1 = rasa_cx3.ndc1;
quit;

data want1;
set want;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM;
run;

proc sort data=want1; by bene_id labels SRVC_DT ; run;

data want2;
do until (last.labels);
  set want1 ;
  by bene_id labels;
  SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;
  output ;
end;
format SRVC_END DATE9. ;
run;

data want3;
set want2;
by bene_id labels ;
format adjStart adjEnd lastdate mmdyy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd   = SRVC_DT + DAYS_SUPLY_NUM -1;
  if first.labels then lastdate = .;
  else do ;
** If overlap then adjust start/end dates, else keep as is;
  if adjStart <= lastdate then do;
    adjStart = lastdate + 1;
    adjEnd   = adjStart + DAYS_SUPLY_NUM -1 ;
  end;
end;

```

```

        end;
    end;
    ** Save last ending date;
    lastdate = adjEnd;
run;

proc sort data=want3 out=want4 nodupkey; by bene_id adjStart descending
DAYS_SUPLY_NUM;
run;

proc sort data=want4 out=d_hyp nodupkey;
by bene_id adjStart;
run;

data aa_h1; set d_hyp;
member_id=bene_id;
fill_dt=adjStart;
format fill_dt DDMMYY10. ;
days_supply=DAYS_SUPLY_NUM;
run;

data aa_h2; set aa_h1;
keep member_id fill_dt days_supply; run;

data claims_hyper; set aa_h2;
run;

*Step 1;

proc sort data=claims_hyper; by member_id fill_dt;
run;

proc transpose data=claims_hyper out=fill_dates (drop=_name_) prefix=fill_dt;
by member_id;
var fill_dt;
run;

proc transpose data=claims_hyper out=days_supply (drop=_name_)
prefix=days_supply;
by member_id;
var days_supply;
run;

data both;
merge fill_dates days_supply;
by member_id;
format start_dt end_dt mmddyy10. ;
start_dt=fill_dt1;
end_dt='31DEC2017'd;
run;

*Step 2;
data pdc_hyper;
set both;
array daydummy(365) day1-day365;
serv_days=end_dt-start_dt+1;

```

```

array fillldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
    do i=1 to dim(fillldates) while (fillldates(i) ne .);
        if fillldates(i)<=first_dt+ii-1<=fillldates(i)+days_supply(i)-1
            then daydummy(ii)=1;
        end;
    end;
drop i ii ;
dayscovered=sum(of daydummy[*]);
label dayscovered='Total Days Covered';
p_dayscovered=dayscovered/serv_days;
label p_dayscovered='Proportion of Days Covered';
drop fill_dt1-fill_dt365 days_supply1-days_supply365;
run;

data pdc_hyper_1;
set pdc_hyper;
bene_id=member_id;
pdc_hyper = p_dayscovered;
sd_hyper = serv_days;
dc_hyper = dayscovered;
keep bene_id pdc_hyper sd_hyper dc_hyper;
run;

proc sort data=pdc_hyper_1; by bene_id;
run;

data increasa_1;
merge increasa (in=in1) pdc_hyper_1;
by bene_id;
if in1;
run;

proc sort data=incdiab_1 out=incdiabs; by bene_id; run;
proc sort data=increasa_1 out=increasas; by bene_id; run;

data inc2; merge incdiabs increasas;
by bene_id;
if age>=18 and number_hyperten=1 and duration_hyperten=1 and esrd_ind="0"
and hospice=0 and varsar=0 then include_hyperten=1;
else include_hyperten=0;
run;

/*****
*PDC Adherence Measure for Statins (Hyperlipidemia Medications, PDC-Sta);
*****/

/*Inclusion criteria
1) 18 years and older
2) First prescription occurred at least 91 days before the end of the year
3) At least two prescriptions on two unique dates
4) Exclusion of ESRD Population */

```

```

/*Determine inclusion outcomes*/
data STA1; set _UPLDS.statins; ndc1=ndc*1; drop ndc; run;

proc sort data=sta1; by ndc1; run;

data d_sta;
merge aa (in=in1) sta1 (in=in2);
by ndc1;
if in1 and in2;
keep bene_id SRVC_DT statins DAYS_SUPLY_NUM;
statins=1;
run;

* Two medications on different dates;
proc sort data=d_sta out=d_sta ; by bene_id ;
run;

proc sort nodupkey data=d_sta out=d_staa; by bene_id; run;

proc sort data=d_sta out=d_stal nodupkey; by bene_id SRVC_DT ;
run;
proc freq data=d_stal noprint;
table bene_id/out=d_sta2; run;

data d_sta2a;
set d_sta2;
if count>=2;
keep bene_id number_statins count;
number_statins=1;
run;

*91 days;

data d_sta3;
set d_stal;
by bene_id;
if first.bene_id then output;
run;

data d_sta4;
set d_sta3;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_statins;
duration_statins=1;
run;

data group3; set rate17_exp; keep bene_id; run;

* Only include records that meets the inclusion criteria;
data aaa_sta;
merge group3 (in=in1) d_staa d_sta2a d_sta4;
by bene_id;
if in1;
run;

data incstan;
set aaa_sta;

```

```

if statins=. then statins=0;
if number_statins=. then number_statins=0;
if duration_statins=. then duration_statins=0;
keep bene_id statins number_statins duration_statins ;
run;

/*Calculate PDC (Measure Assessment)*/

data statins_cx; set _UPLDS.statins2017_vxx; ndc1=ndc*1; drop ndc; run;

data statins_cx1;
set statins_cx;
if Lovastatin_Flag ='X' then label1=1; else label1=.;
if Rosuvastatin_Flag ='X' then label2=2; else label2=.;
if Fluvastatin_Flag ='X' then label3=3; else label3=.;
if Atorvastatin_Flag ='X' then label4=4; else label4=.;
if Pravastatin_Flag ='X' then label5=5; else label5=.;
if Pitavastatin_Flag ='X' then label6=6; else label6=.;
if Simvastatin_Flag ='X' then label7=7; else label7=.;

keep ndc1 label1-label7;
run;

proc sort data=statins_cx1; by ndc1; run;

proc transpose data=statins_cx1 out=statins_cx2;
by ndc1;
run;

data statins_cx3;
set statins_cx2;
labels=coll1;
drop _name_ coll1;
if labels ne .;
run;

proc sql;
create table want as
select aa.*, statins_cx3.labels
from aa, statins_cx3
where aa.ndc1 = statins_cx3.ndc1;
quit;

data want1;
set want;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM;
run;

proc sort data=want1; by bene_id labels SRVC_DT ; run;

data want2;
do until (last.labels);
set want1 ;
by bene_id labels;
SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;
output ;

```

```

    end;
    format SRVC_END DATE9. ;
run;

data want3;
set want2;
by bene_id labels ;
format adjStart adjEnd lastdate mmddyy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd   = SRVC_DT + DAYS_SUPLY_NUM -1;
    if first.labels then lastdate = .;
    else do ;
** If overlap then adjust start/end dates, else keep as is;
    if adjStart <= lastdate then do;
        adjStart = lastdate + 1;
        adjEnd   = adjStart + DAYS_SUPLY_NUM -1 ;
    end;
end;
** Save last ending date;
lastdate = adjEnd;
run;

proc sort data=want3 out=want4 nodupkey; by bene_id adjStart descending
DAYS_SUPLY_NUM;
run;

proc sort data=want4 out=d_sta nodupkey;
by bene_id adjStart;
run;

data aa_s1; set d_sta;
member_id=bene_id;
fill_dt=adjStart;
format fill_dt DDMMYY10. ;
days_supply=DAYS_SUPLY_NUM;
run;

data aa_s2; set aa_s1;
keep member_id fill_dt days_supply; run;

data claims_sta; set aa_s2;
run;

*Step 1;

proc sort data=claims_sta; by member_id fill_dt;
run;

proc transpose data=claims_sta out=fill_dates (drop=_name_) prefix=fill_dt;
by member_id;
var fill_dt;
run;

proc transpose data=claims_sta out=days_supply (drop=_name_)
prefix=days_supply;
by member_id;

```

```

var days_supply;
run;

data both;
merge fill_dates days_supply;
by member_id;
format start_dt end_dt mmddyy10. ;
start_dt=fill_dt1;
end_dt='31DEC2017'd;
run;

*Step 2;
data pdc_sta;
set both;
array daydummy(365) day1-day365;
serv_days=end_dt-start_dt+1;
array filldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
    do i=1 to dim(filldates) while (filldates(i) ne .);
        if filldates(i)<=first_dt+ii-1<=filldates(i)+days_supply(i)-1
            then daydummy(ii)=1;
    end;
end;
drop i ii ;
dayscovered=sum(of daydummy[*]);
label dayscovered='Total Days Covered';
p_dayscovered=dayscovered/serv_days;
label p_dayscovered='Proportion of Days Covered';
drop fill_dt1-fill_dt365 days_supply1-days_supply365;
run;

data pdc_sta_1;
set pdc_sta;
bene_id=member_id;
pdc_sta = p_dayscovered;
sd_sta = serv_days;
dc_sta = dayscovered;
keep bene_id pdc_sta sd_sta dc_sta;
run;

proc sort data=pdc_sta_1; by bene_id;
run;

data incstan_1;
merge incstan (in=in1) pdc_sta_1;
by bene_id;
if in1;
run;

proc sort data=inc2 out=inc2s; by bene_id; run;
proc sort data=incstan_1 out=incstans; by bene_id; run;

```



```

data inc3b; merge inc2s incstans;
by bene_id;
if age>=18 and number_statins=1 and duration_statins=1 and esrd_ind="0" and
hospice=0 then include_statins=1;
else include_statins=0;
run;

data star2017unr_example1;
set inc3b;
run;

/*****
*Drug-Drug Interactions (DDI);
*****/

/*Inclusion Criteria
1) Patients with a prescription claim for a target medication*/

/*Measure Assessment
1) Patients with a prescription claim for a precipitant medication*/

/*Measure Assessment*/

data int;
set _UPLDS.did;
ndc1=ndc*1;
drop ndc;
run;

data did1;
set int;
if Category_Flag ne 'I';
if Step_1_Flag = 'X' then Step=1; else Step=2;
keep ndc1 Category_Flag Step;
run;

proc sort data= aa;
by ndc1;
run;

proc sort data= aa out= pdcint1 nodupkey;
by bene_id ndc1 SRVC_DT descending DAYS_SUPLY_NUM;
run;

data pdcint2;
format end_dt date9.;
set pdcint1;
end_dt=SRVC_DT+DAYS_SUPLY_NUM-1;
keep bene_id SRVC_DT end_dt DAYS_SUPLY_NUM ndc1;
run;

data t1;
set did1;
if step=1;

```

```

run;

data p1;
set did1;
if step=2;
run;

proc sql;
create table drug_t
as select a.bene_id, a.ndc1, a.SRVC_DT as SRVC_DT_t, a.end_dt as end_dt_t,
b.Category_Flag, b.Step
from pdcint2 a, t1 b
where a.ndc1=b.ndc1;
quit;

proc sort data= drug_t; by Category_Flag bene_id SRVC_DT_t;run;

proc sql;
create table drug_p
as select a.bene_id, a.ndc1, a.SRVC_DT as SRVC_DT_p, a.end_dt as end_dt_p,
b.Category_Flag, b.Step
from pdcint2 a, p1 b
where a.ndc1=b.ndc1;
quit;

proc sort data= drug_p; by Category_Flag bene_id SRVC_DT_p;run;

proc sql;
create table drug_tp
as select a.bene_id, a.Category_Flag, a.SRVC_DT_t, a.end_dt_t, b.SRVC_DT_p,
b.end_dt_p
from drug_t a, drug_p b
where a.bene_id=b.bene_id and a.Category_Flag=b.Category_Flag;
quit;

data drug_interaction;
set drug_tp;
if SRVC_DT_t<=SRVC_DT_p<=end_dt_t then did=1;
else if SRVC_DT_t<=end_dt_p<=end_dt_t then did=1;
else did=0;
run;

proc sort data= drug_interaction out= drug_interaction1 nodupkey;
by bene_id Category_Flag did;
run;

proc means data= drug_interaction1 noprint;
class bene_id;
var did;
output out= drug_interaction2 sum=;
run;

data drug_interaction3;
set drug_interaction2;
if did>=1 then dids=1;
if did=0 then dids=0;
if bene_id ne '';

```

```

run;

proc sort data= drug_t out= didall nodupkey; by bene_id; run;

data didflag;
merge didall(in=in1) drug_interaction3(in=in2);
by bene_id;
if dids=. then dids=0;
keep bene_id dids;
run;

proc sort data= didflag; by bene_id; run;

/* Determine inclusion outcomes */

data int_a;
set UPLDS.did;
ndc1=ndc*1;
drop ndc;
run;

data int_a1;
set int_a;
if Step_1_Flag = 'X';
run;

proc sort data= int_a1; by ndc1; run;

data int1;
merge aa(in=in1) int_a1(in=in2);
by ndc1;
if in1 and in2;
keep BENE_ID ;
run;

proc sort data=int1 nodupkey out=int2;
by bene_id;
run;

data incint;
set int2;
incint=1;
keep bene_id incint;
run;

/*****
*Diabetes Dosing (DOS);
*****/

/*Inclusion criteria
1) Patients who were dispensed with a prescription claim for the four types
of diabetes medication*/

/*Measure Assessment

```

1) Patients with a dose of diabetes medication that exceeds US Food & Drug Administration approved dosing guidelines*/

```
data dos;
set _UPLDS.dos;
ndc1=ndc*1;
drop ndc;
run;
```

```
proc sort data=dos;
by ndc1;
run;
```

*Align drug type to strength and maximum dosage level;

```
data sfu;
set dos;
if sfu_flag ne "";
if SFU_FLAG="Drug1" then strength=DRUGSTRENGTH1*1 ;
if SFU_FLAG="Drug2" then strength=DRUGSTRENGTH2*1 ;

if SFU_FLAG="Drug1" then maxdose=maxdose1*1;
if SFU_FLAG="Drug2" then maxdose=maxdose2*1;
keep ndc1 sfu_flag strength maxdose;
run;
```

```
data BIGUANIDE;
set dos;
if BIGUANIDE_FLAG ne "";
if BIGUANIDE_FLAG="Drug1" then strength=DRUGSTRENGTH1*1;
if BIGUANIDE_FLAG="Drug2" then strength=DRUGSTRENGTH2*1 ;

if BIGUANIDE_FLAG="Drug1" then maxdose=maxdose1*1;
if BIGUANIDE_FLAG="Drug2" then maxdose=maxdose2*1;
keep ndc1 BIGUANIDE_FLAG strength maxdose;
run;
```

```
data tzd;
set dos;
if TZD_FLAG ne "";
if TZD_FLAG="Drug1" then strength=DRUGSTRENGTH1*1 ;
if TZD_FLAG="Drug2" then strength=DRUGSTRENGTH2*1 ;

if TZD_FLAG="Drug1" then maxdose=maxdose1*1;
if TZD_FLAG="Drug2" then maxdose=maxdose2*1;
keep ndc1 TZD_FLAG strength maxdose;
run;
```

```
data DDPIV;
set dos;
if DDPIV_FLAG ne "";
if DDPIV_FLAG="Drug1" then strength=DRUGSTRENGTH1*1 ;
if DDPIV_FLAG="Drug2" then strength=DRUGSTRENGTH2*1 ;

if DDPIV_FLAG="Drug1" then maxdose=maxdose1*1;
if DDPIV_FLAG="Drug2" then maxdose=maxdose2*1;
```

```

keep ndc1 DDPIV_FLAG strength maxdose;
run;

/*Overdose for SFU*/

proc sort data= sfu;
by ndc1;
run;

data PQAa;
merge aa(in=in1) sfu(in=in2);
by ndc1;
if in1 and in2;
run;

proc sort data=PQAa out=dosz_sfu nodupkey;
by bene_id;
run;

data med_sfu;
set dosz_sfu;
med_sfu=1;
keep bene_id med_sfu;
run;

/*Define Inclusion Outcomes for SFU*/
proc sort data=PQAa out=dosz_sfu;
by bene_id SRVC_DT;
run;

data dosx_sfu;
set dos_sfu;
by bene_id;
if first.bene_id then output;
run;

data dosx_sfu1;
set dosx_sfu;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_sfu age age_boy;
duration_sfu=1;
run;

data inc_sfu;
set dosx_sfu1;
if age>=18 and duration_sfu=1 then inc_dos_sfu=1;else inc_dos_sfu=0;
keep bene_id duration_sfu inc_dos_sfu;
run;

/*Measure Assessment for SFU*/

data PQAa;
merge aa(in=in1) sfu(in=in2);
by ndc1;
if in1 and in2;
run;

```

```

proc sort data= PQAa; by bene_id SRVC_DT;
run;

data PQAa1;
set PQAa;
by bene_id;
if first.bene_id then output;
run;

data PQAa2;
set PQAa1;
if '31Dec2017'd-SRVC_DT>=0;
keep bene_id;
run;

data PQAa3;
merge PQAa2(in=in1) PQAa (in=in2);
by bene_id;
if in1;
run;

data sfuoverdose;
set PQAa3;
dailydose=strength*QTY_DSPNSD_NUM/DAYS_SUPLY_NUM;
if dailydose>maxdose then snfoverflag=1;
else snfoverflag=0;
run;

proc sort data= sfuoverdose out= sfuoverdose_nondup nodupkey equals;
by bene_id descending snfoverflag ;
run;

data sfuoverdose_dummy;
set sfuoverdose_nondup;
by bene_id;
if first.bene_id then output;
keep bene_id snfoverflag;
run;

/*Overdose for Biguanide*/

proc sort data= BIGUANIDE; by ndc1;run;

data PQAAb;
merge aa(in=in1) BIGUANIDE(in=in2);
by ndc1;
if in1 and in2;
run;

proc sort data=PQAAb out=dosz_bigu nodupkey;
by bene_id;
run;

data med_bigu;
set dosz_bigu;

```

```

med_bigu=1;
keep bene_id med_bigu;
run;

/*Define Inclusion Outcomes for Biguanide*/

proc sort data=PQAb out=dos_bigu;
by bene_id SRVC_DT;
run;

data dosx_bigu;
set dos_bigu;
by bene_id;
if first.bene_id then output;
run;

data dosx_bigu1;
set dosx_bigu;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_bigu age age_boy;
duration_bigu=1;
run;

data inc_bigu;
set dosx_bigu1;
if age>=18 and duration_bigu=1 then inc_dos_bigu=1;else inc_dos_bigu=0;
keep bene_id duration_bigu inc_dos_bigu;
run;

/*Measure Assessment for Biguanide */
data PQAb;
merge aa(in=in1) BIGUANIDE(in=in2);
by ndc1;
if in1 and in2;
run;
proc sort data= PQAb; by bene_id SRVC_DT;
run;

data PQAb1;
set PQAb;
by bene_id;
if first.bene_id then output;
run;

data PQAb2;
set PQAb1;
if '31Dec2017'd-SRVC_DT>=0;
keep bene_id;
run;

data PQAb3;
merge PQAb2(in=in1) PQAb(in=in2);
by bene_id;
if in1;
run;

data biguanideover;

```

```

set PQAb3;
dailydose=strength*QTY_DSPNSD_NUM/DAYS_SUPLY_NUM;
if dailydose>maxdose then BIGUANIDEoverflag=1;
else BIGUANIDEoverflag=0;
run;

proc sort data= biguanideover out= biguanideover_nondup nodupkey equals;
by bene_id descending BIGUANIDEoverflag;
run;

data biguanideover_dummy;
set biguanideover_nondup;
by bene_id;
if first.bene_id then output;
keep bene_id BIGUANIDEoverflag;
run;

/*Overdose for TZD*/

proc sort data= tzd; by ndc1;run;

data PQAt;
merge aa(in=in1) tzd(in=in2);
by ndc1;
if in1 and in2;
run;

proc sort data=PQAt out=dosz_tzd nodupkey;
by bene_id;
run;

data med_tzd;
set dosz_tzd;
med_tzd=1;
keep bene_id med_tzd;
run;

/*Define Inclusion Outcomes for TZD*/

proc sort data=PQAt out=dos_tzd;
by bene_id SRVC_DT;
run;

data dosx_tzd;
set dos_tzd;
by bene_id;
if first.bene_id then output;
run;

data dosx_tzd1;
set dosx_tzd;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_tzd age age_boy;
duration_tzd=1;
run;

data inc_tzd;

```



```

set dosx_tzd1;
if age>=18 and duration_tzd=1 then inc_dos_tzd=1;else inc_dos_tzd=0;
keep bene_id duration_tzd inc_dos_tzd;
run;

/*Measure Assessment for TZD */
data PQAt;
merge aa(in=in1) tzd(in=in2);
by ndc1;
if in1 and in2;
run;

proc sort data= PQAt; by bene_id SRVC_DT;
run;

data PQAt1;
set PQAt;
by bene_id;
if first.bene_id then output;
run;

data PQAt2;
set PQAt1;
if '31Dec2017'd-SRVC_DT>=0;
keep bene_id;
run;

data PQAt3;
merge PQAt2(in=in1) PQAt(in=in2);
by bene_id;
if in1;
run;

data tzdoverdose;
set PQAt3;
dailydose=strength*QTY_DSPNSD_NUM/DAYS_SUPLY_NUM;
if dailydose>maxdose then tzdoverflag=1;
else tzdoverflag=0;
run;

proc sort data= tzdoverdose out= tzdoverdose_nondup nodupkey equals;
by bene_id descending tzdoverflag;
run;

data tzdoverdose_dummy;
set tzdoverdose_nondup;
by bene_id;
if first.bene_id then output;
keep bene_id tzdoverflag;
run;

/*Overdose for DDPIV*/

proc sort data= ddpiv; by ndc1;run;

data PQAd;

```

```

merge aa(in=in1) ddpiv(in=in2);
by ndc1;
if in1 and in2;
run;

proc sort data=PQAd out=dosz_ddpiv nodupkey;
by bene_id;
run;

data med_ddpiv;
set dosz_ddpiv;
med_ddpiv=1;
keep bene_id med_ddpiv;
run;

/*Define Inclusion Outcomes for DDPIV*/

proc sort data=PQAd out=dos_ddpiv;
by bene_id SRVC_DT;
run;

data dosx_ddpiv;
set dos_ddpiv;
by bene_id;
if first.bene_id then output;
run;

data dosx_ddpiv1;
set dosx_ddpiv;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_ddpiv age age_boy;
duration_ddpiv=1;
run;

data inc_ddpiv;
set dosx_ddpiv1;
if age>=18 and duration_ddpiv=1 then inc_dos_ddpiv=1;else inc_dos_ddpiv=0;
keep bene_id duration_ddpiv inc_dos_ddpiv;
run;

/*Measure Assessment for DDPIV */
data PQAd;
merge aa(in=in1) ddpiv(in=in2);
by ndc1;
if in1 and in2;
run;

proc sort data= PQAd; by bene_id SRVC_DT;
run;

data PQAd1;
set PQAd;
by bene_id;
if first.bene_id then output;
run;

data PQAd2;

```

```

set PQAd1;
if '31Dec2017'd-SRVC_DT>=0;
keep bene_id;
run;

data PQAd3;
merge PQAd2(in=in1) PQAd(in=in2);
by bene_id;
if in1;
run;

data ddivover;
set PQAd3;
dailydose=strength*QTY_DSPNSD_NUM/DAYS_SUPLY_NUM;
if dailydose>maxdose then ddpivoverflag=1;
else ddpivoverflag=0;
run;

proc sort data= ddivover out= ddivover_nondup nodupkey equals;
by bene_id descending ddpivoverflag;
run;

data ddivover_dummy;
set ddivover_nondup;
by bene_id;
if first.bene_id then output;
keep bene_id ddpivoverflag;
run;

*Final Calculation;

data excessivedoseall;
merge sfuoverdose_dummy biguanideover_dummy tzdoverdose_dummy ddivover_dummy;
by bene_id;
if BIGUANIDEoverflag=1 or snfoverflag=1 or tzdoverflag=1 or ddpivoverflag=1
then excessivedoseall=1;
else excessivedoseall=0;
run;

*Confirm no duplicates;
proc sort data= excessivedoseall nodupkey; by bene_id; run;
proc sort data=incint nodupkey; by bene_id; run;
proc sort data= didflag nodupkey; by bene_id; run;
proc sort data=inc_sfu nodupkey; by bene_id; run;
proc sort data=inc_bigu nodupkey; by bene_id; run;
proc sort data=inc_tzd nodupkey; by bene_id; run;
proc sort data=inc_ddpiv nodupkey; by bene_id; run;
proc sort data=med_sfu nodupkey; by bene_id; run;
proc sort data=med_bigu nodupkey; by bene_id; run;
proc sort data=med_tzd nodupkey; by bene_id; run;
proc sort data=med_ddpiv nodupkey; by bene_id; run;

* Merge DDI and DOS measure calculation with previous measures;
proc sort data=inc3b out=inc3bs; by bene_id; run;
data inc4;

```

```

merge inc3bs (in=in1) excessivedoseall incint didflag inc_sfu inc_bigu
inc_tzd inc_ddpiv med_sfu med_bigu med_tzd med_ddpiv;
by bene_id;
if in1;
if incint=. then incint=0;
if inc_dos_bigu = . then inc_dos_bigu =0;
if inc_dos_sfu = . then inc_dos_sfu =0;
if inc_dos_ddpiv = . then inc_dos_ddpiv =0;
if inc_dos_tzd = . then inc_dos_tzd =0;
if duration_bigu = . then duration_bigu =0;
if duration_sfu = . then duration_sfu =0;
if duration_tzd = . then duration_tzd =0;
if duration_ddpiv = . then duration_ddpiv =0;
if med_bigu = . then med_bigu =0;
if med_sfu = . then med_sfu =0;
if med_tzd = . then med_tzd =0;
if med_ddpiv = . then med_ddpiv =0;
if inc_dos_bigu=1 or inc_dos_sfu=1 or inc_dos_ddpiv=1 or inc_dos_tzd=1 then
incdos=1; else incdos=0;
if BIGUANIDEoverflag=. then BIGUANIDEoverflag=0;
if snfoverflag=. then snfoverflag=0;
if tzdooverflag=. then tzdooverflag=0;
if ddpivoverflag=. then ddpivoverflag=0;
if excessivedoseall=. then excessivedoseall=0;
run;

data inc4; set inc4; run;

/*****
*Statin Use among Patients with Diabetes (SUPD);
*****/

/*Inclusion Criteria
1) Exclude hospice
2) Age=40-75 at the beginning of the year
3) Two diabetes meds regardless of the dates
4) Exclude ESRD */

/*Measure Assessment
1)Patients with more than one fills for a statin medication */

/*Determine Inclusion Outcomes*/

* Get list of medications;

data supd_cx; set _UPLDS.supd2017_vxx; ndc1=ndc*1; drop ndc; run;

data supd_cx1;
set supd_cx;
if Biguanide_Flag ='X' then label1=1; else label1=.;
if SFU_Flag ='X' then label2=2; else label2=.;
if TZD_Flag ='X' then label3=3; else label3=.;
if DDPIV_Flag ='X' then label4=4; else label4=.;
if IncretinMimetic_Flag ='X' then label5=5; else label5=.;
if Meglitinide_Flag ='X' then label6=6; else label6=.;

```

```

if SGLT2Inhib_Flag = 'X'          then label7=7; else label7=.;
if AlphaGlucosidaseInhibitor_Flag = 'X'      then label8=8; else label8=.;
if AmylinAnalog_Flag = 'X'          then label9=9; else label9=.;
if Human_Insulin_Flag = 'X'         then label10=10; else label10=.;

keep ndc1 label1-label10;
run;

proc sort data=supd_cx1; by ndc1; run;

proc transpose data=supd_cx1 out=supd_cx2;
by ndc1;
run;

data supd_cx3;
set supd_cx2;
labels=col1;
drop _name_ col1;
if labels ne .;
run;

proc sql;
create table want as
select aa.*, supd_cx3.labels
from aa, supd_cx3
where aa.ndc1 = supd_cx3.ndc1;
quit;

*Include diabetes meds;

data want1;
set want;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM diabetes_sta;
diabetes_sta=1;
run;

proc sort data=want1 nodupkey out=d; by bene_id; run;

data das;
set d;
keep bene_id diabetes_sta;
run;

* Two medications;
proc sort data=want1 out=ds; by bene_id labels SRVC_DT ; run;

proc freq data=ds noprint;
table bene_id*labels/out=d2; run;

proc sort data=d2 nodupkey out=d2_x1; by bene_id labels descending count;
run;

data d2a;
set d2_x1;
if count>=2;
keep bene_id number_diasta;
number_diasta=1;

```

```

run;

data d2as;
set d2a;
by bene_id;
if first.bene_id then output;
run;

data cohort; set inc4; run;

proc sort data=cohort out=cohorts; by bene_id; run;
proc sort data=das; by bene_id; run;
proc sort data=d2as; by bene_id; run;

data aaa;
merge cohorts (in=in1) das d2as;
by bene_id;
if in1;
run;

data inc_StaUse;
set aaa;
if diabetes_sta =. then diabetes_sta =0;
if number_diasta=. then number_diasta=0;
if diabetes_sta=1 and number_diasta=1 and esrd_ind="0" and hospice=0 and
39<age_BOY<76 then include_StaUse=1;
else include_StaUse=0;
keep bene_id include_StaUse diabetes_sta number_diasta;
run;

/* Measure Assessment*/

data statins_dia;
set _UPLDS.statins;
ndc1=ndc*1;
drop ndc;
run;

data d_statins_dia;
merge aa (in=in1) statins_dia (in=in2);
by ndc1;
if in1 and in2;
keep bene_id statins_dia;
statins_dia=1;
run;

proc sort data=d_statins_dia nodupkey out=d_statins_dia_1; by bene_id; run;
proc sort data=inc_StaUse; by bene_id; run;

data inc_StaUse1;
merge inc_StaUse (in=in1) d_statins_dia_1;
by bene_id;
if in1;
run;

data inc_StaUse2;
set inc_StaUse1;

```

```

if statins_dia =. then statins_dia =0;
run;

/*****
* Antipsychotic Use Among Patients with Dementia (APD);
*****/

/* Inclusion Criteria
1) 65 years or older
2) Dementia diagnosis or two dementia meds (APD-A) with more than 60 days of
supply*/

/* Measure Assessment
1) One or more antipsychotic prescription claims with more than 30 days of
supply
2) Do not have a diagnosis for schizophrenia, bipolar disorder, Huntington's
disease, or Tourette's syndrome*/

/*Determine inclusion outcomes*/

*Dementia Diagnosis;

data dementiadiag;
set star.DIAG_2017; /*2017 diagnosis files*/
BENE_ID = HICNO;
dxcode=upcase(diag);
dx_3=substr(dxcode,1,3);
dx_4=substr(dxcode,1,4);
dx_5=substr(dxcode,1,5);

if dx_5='A8100' | dx_5='A8101' | dx_5='A8109' | dx_5='F0150' |
dx_5='F0151' | dx_5='F0280' | dx_5='F0281' | dx_5='F0390' | dx_5='F0391' |
dx_5='F1027' | dx_5='F1097' | dx_5='F1327' | dx_5='F1397' | dx_5='F1897' |
dx_5='F1917' | dx_5='F1927' | dx_5='F1997' | dx_4='G300' | dx_4='G301' |
dx_4='G308' | dx_4='G309' | dx_5='G3101' | dx_5='G3109' | dx_4='G311' |
dx_5='G3183' | dx_5='F1096' |

dx_5='04619' | dx_5='04611' | dx_5='29040' |
dx_5='29041' | dx_5='29410' | dx_5='29411' | dx_4='2900' | dx_5='29010' |
dx_5='29011' | dx_5='29013' | dx_5='29021' | dx_4='2908' | dx_4='2909' |
dx_5='29420' | dx_5='29421' |
dx_4='2912' | dx_4='2911' | dx_5='29282' | dx_5='30410' |
dx_5='30460' | dx_4='3310' |
dx_5='33111' | dx_5='33119' | dx_4='3312' |
dx_5='33182' then dementia_diagdum=1; else dementia_diagdum=0;
if dementia_diagdum=1;
keep BENE_ID dementia_diagdum;
run;

proc sort data= dementiadiag out= dementiadiag2 nodupkey; by bene_id; run;

*dementia medication;
data dementia2016_cx; set _UPLDS.dementia2017_vxx; ndc1=ndc*1; drop ndc; run;
proc sort data=dementia2016_cx; by ndc1; run;

data DementiaMeds_2015;

```

```

set dementia2016_cx;
if Galantamine_flag ='X' then label=1;
if Rivastigmine_flag ='X' then label=2;
if Donepezil_flag ='X' then label=3;
if Memantine_flag ='X' then label=4;
keep ndc1 label;
run;

*merge PDE file and dementia medication file by NDC;

data DementiaMeds_2015a;
merge aa(in=in1) DementiaMeds_2015(in=in2);
by NDC1;
if in1 and in2;
run;

proc sort data= DementiaMeds_2015a out=dj nodupkey; by bene_id; run;

data dj1;
set dj;
dementiap_med=1;
keep bene_id dementiap_med;
run;

data DementiaMeds_2015b;
set DementiaMeds_2015a;
keep bene_id ndc1 label srvc_dt DAYS_SUPLY_NUM;
run;

proc sort data= DementiaMeds_2015b out = DementiaMeds_2015c;
by BENE_ID label SRVC_DT descending DAYS_SUPLY_NUM;
run;

*patients have two or more prescription claims;

data DementiaMeds_2015d;
  set DementiaMeds_2015c;
  by bene_id label;
  if last.label;
  count=_n_-sum(lag(_n_),0);
  do _n_=1 to count;
    set DementiaMeds_2015c;

    output;
  end;
run;

data DementiaMeds_2015d_1;
set DementiaMeds_2015d;
if count>=2;
drop count ndc1;
run;

proc sort data= DementiaMeds_2015d_1 out = DementiaMeds_2015d_2;
by BENE_ID label SRVC_DT descending DAYS_SUPLY_NUM;
run;

```


*Patients must have more than 60 days supply for a cholinesterase inhibitor or NMDA receptor antagonist. Do not adjust for overlapping days. If multiple prescription claims for the same target medication are dispensed on the same day, keep prescriptions with the longest days supply;

```
proc sort data= DementiaMeds_2015d_2 out= DementiaMeds_2015e nodupkey;
by bene_id label srvc_dt;
run;
```

```
data transit;
set DementiaMeds_2015e;
fill_dt=SRVC_DT;
format fill_dt mmddyy10. ;
days_supply=DAYS_SUPLY_NUM;
run;
```

```
data claims;
set transit;
keep bene_id label fill_dt days_supply;
run;
```

*Step 1;

```
proc sort data= claims; by bene_id label fill_dt;
run;
```

```
proc transpose data= claims out= fill_dates (drop=_name_) prefix=fill_dt;
by bene_id label;
var fill_dt;
run;
```

```
proc transpose data= claims out= days_supply (drop=_name_)
prefix=days_supply;
by bene_id label;
var days_supply;
run;
```

```
data both;
merge fill_dates days_supply;
by bene_id label;
format start_dt end_dt mmddyy10.;
start_dt=fill_dt1;
end_dt=input('12/31/2017', mmddyy10.);
run;
```

*Step 2;

```
data dementiadiag3;
set both;
array daydummy(365) day1-day365;
array filldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
```

```

        do i=1 to dim(filldates) while (filldates(i) ne .);
            if filldates(i) <= first_dt + ii -1 <=
filldates(i)+days_supply(i)-1
                then daydummy(ii)=1;
            end;
        end;
drop i ii;
dayscovered_demmed = sum(of day1-day365);
if dayscovered_demmed>60 then medcovered_dem = 1; else medcovered_dem = 0;
keep bene_id dayscovered_demmed medcovered_dem;
run;

proc sort data= dementiadiag3; by bene_id descending dayscovered_demmed; run;

data dementiadiag3x;
set dementiadiag3;
by bene_id;
if first.bene_id then output;
run;

* Record inclusion outcomes for APD and merge with SUPD;
proc sort data= dementiadiag2 nodupkey; by bene_id; run;
proc sort data=dj1 nodupkey; by bene_id; run;
proc sort data= dementiadiag3x nodupkey; by bene_id; run;
proc sort data=inc_StaUse2 nodupkey; by bene_id; run;

data aaa;
merge cohorts (in=in1) dementiadiag2 dementiadiag3x inc_StaUse2 dj1;
by bene_id;
if in1;
run;

data incdem;
set aaa;
if dementiap_med = . then dementiap_med = 0;
if dayscovered_demmed = . then dayscovered_demmed = 0;
if medcovered_dem = . then medcovered_dem = 0;
if dementia_diagdum =. then dementia_diagdum = 0;
if (age_BOY>64 and dementia_diagdum=1) or (age_BOY>64 and medcovered_dem=1)
then include_APD=1;
else include_APD=0;
keep bene_id diabetes_sta statins_dia include_StaUse number_diasta
dementia_diagdum include_APD dementiap_med dayscovered_demmed medcovered_dem;
run;

*Add results associated with this measure to previous measures;
proc sort data=incdem; by bene_id; run;

data finalcohortn;
Merge cohorts incdem;
by bene_id;
run;

data finalcohortn; set finalcohortn; run;

data apdpop;
set incdem;

```

```

keep bene_id;
run;

proc sort data= apdpop out= apdpop1 nodupkey; by bene_id;
run;

/* Measure Assessment
Psychotic Disorders:
Schizophrenia - F200 F201 F202 F203 F205 F2081 F2089 F209
Schizotypal Disorder - F21
Schizoaffective Disorder - F250 F251 F258 F259
Manic Episode - F3010 F3011 F3012 F3013 F302 F303 F304 F308 F309
Bipolar Disorder - F310 F3110 F3111 F3112 F3113 F312 F3130 F3131 F3132 F314
F315 F3160 F3161 F3162 F3163 F3164
F3170 F3171 F3172 F3173 F3174 F3175 F3176 F3177 F3178 F3181 F3189 F319
Tourette's Disorder - F952
Huntington's Disease - G10
Others - F060 F062 F0633 F23 F24 F28 F29 F323 F333 F53*/

data Antipsychoticdiag_1;
set star.DIAG_2017;
BENE_ID = HICNO;
dxcode=upcase(diag);
dx_3=substr(dxcode,1,3);
dx_4=substr(dxcode,1,4);
dx_5=substr(dxcode,1,5);

if
dx_4='F200' |dx_4='F201' |dx_4='F202' |dx_4='F203' | dx_4='F205' |
dx_5='F2081' |dx_5='F2089' |dx_4='F209' |

dx_4='F310' |dx_5='F3110' |dx_5='F3111' |dx_5='F3112' |
dx_5='F3113' |dx_4='F312' |dx_5='F3130' |dx_5='F3131' |dx_5='F3132' |
dx_4='F314' |dx_4='F315' |
dx_5='F3160' |dx_5='F3161' |dx_5='F3162' |dx_5='F3163' |dx_5='F3164' |
dx_5='F3170' |dx_5='F3171' |dx_5='F3172' |
dx_5='F3173' |dx_5='F3174' |dx_5='F3175' |dx_5='F3176' |dx_5='F3177' |
dx_5='F3178' |dx_5='F3181' |dx_5='F3189' |dx_4='F319' |

dx_4='F952' |
dx_3='G10' |

dx_4='F259' |dx_5='F3010' |
dx_5='F3011' |dx_5='F3012' |dx_5='F3013' |dx_4='F302' |dx_4='F303' |
dx_4='F304' |dx_4='F308' |
dx_4='F060' |
dx_4='F062' |dx_5='F0633' |dx_3='F21' |dx_3='F23' |
dx_3='F24' |dx_4='F250' |dx_4='F251' |dx_4='F258' |dx_3='F28' |
dx_3='F29' |dx_4='F309' |dx_4='F323' |dx_4='F333' |
dx_3='F53' |

dx_5='29530' |dx_5='29510' |dx_5='29520' |dx_5='29590' | dx_5='29560' |
dx_5='29540' |dx_5='29580' |

dx_5='29640' |dx_5='29641' |dx_5='29642' |

```

```
dx_5='29643' |dx_5='29644' |dx_5='29650' |dx_5='29651' |dx_5='29652'|
dx_5='29653' |dx_5='29654' |
dx_5='29660' |dx_5='29661' |dx_5='29662' |dx_5='29663' |dx_5='29664'|
dx_4='2967' |
dx_5='29645' |dx_5='29646' |dx_5='29655' |dx_5='29656' |dx_5='29665'|
dx_5='29666' |dx_5='29689' |dx_5='29680' |
```

```
dx_5='30723' |
dx_4='3334' |
```

```
dx_5='29570' |dx_5='29600' |
dx_5='29601' |dx_5='29602' |dx_5='29603' |dx_5='29604' |dx_5='29605'|
dx_5='29606' |dx_5='29681' |
dx_5='29382' |
dx_5='29381' |dx_5='29383' |dx_5='30122' |dx_4='2983' |
dx_4='2984' |dx_4='2988' |
dx_4='2973' |dx_4='2989'|
dx_5='29624' |dx_4='2980' |dx_5='29634' |
dx_5='29389' then antipsych_diagdum =1; else antipsych_diagdum =0;
if antipsych_diagdum =1;
keep BENE_ID antipsych_diagdum;
run;
```

```
proc sort data= Antipsychoticdiag_1 out= Antipsychoticdiag_2 nodupkey;
by bene_id; run;
```

```
*Antipsychotic Medication;
```

```
data antipsy2016_cx; set _UPLDS.antipsy2017_vxx; ndc1=ndc*1; drop ndc; run;
proc sort data=antipsy2016_cx; by ndc1; run;
```

```
data AntipsychoticMeds_2015;
set antipsy2016_cx;
if Aripiprazole_flag='X' then label=1;
if Asenapine_flag='X' then label=2;
if Brexpiprazole_flag='X' then label=3;
if Chlopromazine_flag='X' then label=4;
if Fluphenazine_flag='X' then label=5;
if Haloperidol_flag='X' then label=6;
if Iloperidone_flag='X' then label=7;
if Loxapine_flag='X' then label=8;
if Lurasidone_flag='X' then label=9;
if Molindone_flag='X' then label=10;
if Olanzapine_flag='X' then label=11;
if Pipliperidone_flag='X' then label=12;
if Perphenazine_flag='X' then label=13;
if Pimavanserin_flag='X' then label=14;
if Cariprazine_flag='X' then label=15;
if Pimozide_flag='X' then label=16;
if Quetiapine_flag='X' then label=17;
if Risperidone_flag='X' then label=18;
if Thioridazine_flag='X' then label=19;
if Thiothixene_flag='X' then label=20;
if Trifluoperazine_flag='X' then label=21;
if Ziprasidone_flag='X' then label=22;
if Clozapine_flag='X' then label=23;
```

```

keep ndc1 label;
run;

*merge PDE file and the dementia medication file by NDC;

data AntipsychoticMeds_2015a;
merge aa(in=in1) AntipsychoticMeds_2015(in=in2);
by NDC1;
if in1 and in2;
run;

data AntipsychoticMeds_2015b;
set AntipsychoticMeds_2015a;
keep bene_id label srvc_dt DAYS_SUPLY_NUM;
run;

*patients have more than 30 days supply for any antipsychotic medication. Do
not credit overlapping days;

proc sort data= AntipsychoticMeds_2015b out = AntipsychoticMeds_2015c;
by BENE_ID label SRVC_DT descending DAYS_SUPLY_NUM;
run;

*if multiple prescription claims for the same target medication are dispensed
on the same day, keep prescriptions with the longest days supply;

proc sort data= AntipsychoticMeds_2015c out= AntipsychoticMeds_2015e
nodupkey;
by bene_id label srvc_dt;
run;

data transit_1;
set AntipsychoticMeds_2015e;
fill_dt=SRVC_DT;
format fill_dt DDMMYY10. ;
days_supply=DAYS_SUPLY_NUM;
run;

data claims1;
set transit_1;
keep bene_id label fill_dt days_supply;
run;

*Step 1;

proc sort data= claims1; by bene_id label fill_dt;
run;

proc transpose data= claims1 out= fill_dates_1 (drop=_name_) prefix=fill_dt;
by bene_id label;
var fill_dt;
run;

proc transpose data= claims1 out= days_supply_1 (drop=_name_)
prefix=days_supply;
by bene_id label;

```

```

var days_supply;
run;

data both_1;
merge fill_dates_1 days_supply_1;
by bene_id label;
format start_dt end_dt mmddyy10.;
start_dt=fill_dt1;
end_dt=input('12/31/2017', mmddyy10.);
run;

*Step 2;
data dementia_cal;
set both_1;
array daydummy(365) day1-day365;
array filldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
  do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
  do ii=1 to 365;
    do i=1 to dim(filldates) while (filldates(i) ne .);
      if filldates(i) <= first_dt + ii -1 <=
filldates(i)+days_supply(i)-1
      then daydummy(ii)=1;
    end;
  end;
drop i ii;
dayscovered = sum(of day1-day365);
meddays_antipsych=dayscovered;
keep bene_id meddays_antipsych;
run;

proc sort data= dementia_cal; by bene_id descending meddays_antipsych; run;

data dementia_call;
set dementia_cal;
by bene_id;
if first.bene_id then output;
run;

proc sort data= dementia_call; by bene_id; run;
proc sort data= Antipsychoticdiag_2; by bene_id; run;
proc sort data= apdpop1; by bene_id; run;

data apdnumerator;
merge apdpop1(in=in1) dementia_call Antipsychoticdiag_2 ;
by bene_id;
if in1;
keep bene_id meddays_antipsych antipsych_diagdum;
run;

data apdnumerator1;
set apdnumerator;
if meddays_antipsych=. then meddays_antipsych=0;
if antipsych_diagdum=. then antipsych_diagdum=0;

```

```

if meddays_antipysch>30 and antipysch_diagdum=0 then antipsychotic_flag=1;
else antipsychotic_flag=0;
run;

proc sort data= apdnumerator1 out= apdnumerator2 nodupkey; by bene_id;
run;

proc sort data=finalcohortn; by bene_id; run;

data aaa;
merge finalcohortn (in=in1) apdnumerator2;
by bene_id;
if in1;
run;

data finalcohortz;
set aaa;
if meddays_antipysch=. then meddays_antipysch=0;
if antipysch_diagdum=. then antipysch_diagdum=0;
if antipsychotic_flag=. then antipsychotic_flag=0;
antipsychotic_ndementia = antipsychotic_flag;
run;

data star2017unr_finalcohortz;
set finalcohortz;
run;

/*****
*High-Risk Medication in the Elderly (HRM);
*****/

/*Inclusion criteria
1) 65 years and older
2) Exclusion of Hospice Population */

/*Measure Assessment
1) 2 or more prescription claims on different dates for the same medication
2) Five medications are only considered as an HRM if the following are
satisfied
    a. Nitrofurantoin - more than 90 cumulative days of supply
    b. Nonbenzodiazepine Hypnotics - more than 90 cumulative days of supply
    c. Reserpine - more than 0.1 mg of average daily dose
    d. Digoxin - more than 0.125 mg of average daily dose
    e. Doxepin - more than 6 mg of average daily dose*/

/* Determine inclusion outcomes*/

data finalcohortz1;
set star2017unr_finalcohortz;
if hospice = 0 and age_boy > 64 then include_hrm = 1;
else include_hrm=0;
run;

/* Measure Assessment*/

```

```
data c_hrm;  
set _UPLDS.hrm;  
ndc1=ndc*1;  
drop ndc;  
run;
```

```
data c_hrm1;  
set c_hrm;  
if Amitriptyline_flag='X' then label=1;  
if Amobarbital_flag='X' then label=2;  
if Benzotropine_flag='X' then label=3;  
if Brompheniramine_flag='X' then label=4;  
if Butabarbital_flag='X' then label=5;  
if Butalbital_flag='X' then label=6;  
if Carbinoxamine_flag='X' then label=7;  
if Carisoprodol_flag='X' then label=8;  
if Dexbrompheniramine_flag='X' then label=9;  
if Chlorpheniramine_flag='X' then label=10;  
if Chlorpropamide_flag='X' then label=11;  
if Chlorzoxazone_flag='X' then label=12;  
if Clemastine_flag='X' then label=13;  
if Clomipramine_flag='X' then label=14;  
if Cyclobenzaprine_flag='X' then label=15;  
if Cyproheptadine_flag='X' then label=16;  
if Dexchlorpheniramine_flag='X' then label=17;  
if Digoxin_flag='X' then label=18;  
if Diphenhydramine_flag='X' then label=19;  
if Dipyridamole_flag='X' then label=20;  
if Disopyramide_flag='X' then label=21;  
if Doxepin_flag='X' then label=22;  
if Doxylamine_flag='X' then label=23;  
if Guanabenz_flag='X' then label=24;  
if Ergoloid_mesylates_flag='X' then label=25;  
if Estrogens_flag='X' then label=26;  
if Eszopiclone_flag='X' then label=27;  
if Glyburide_flag='X' then label=28;  
if Guanfacine_flag='X' then label=29;  
if Hydroxyzine_flag='X' then label=30;  
if Imipramine_flag='X' then label=31;  
if Indomethacin_flag='X' then label=32;  
if Isoxsuprine_flag='X' then label=33;  
if Ketorolac_flag='X' then label=34;  
if Megestrol_flag='X' then label=35;  
if Meperidine_flag='X' then label=36;  
if Meprobamate_flag='X' then label=37;  
if Metaxalone_flag='X' then label=38;  
if Methocarbamol_flag='X' then label=39;  
if Methyl dopa_flag='X' then label=40;  
if Nifedipine_flag='X' then label=41;  
if Nitrofurantoin_flag='X' then label=42;  
if Orphenadrine_flag='X' then label=43;  
if Pentazocine_flag='X' then label=44;  
if Pentobarbital_flag='X' then label=45;  
if Phenobarbital_flag='X' then label=46;  
if Promethazine_flag='X' then label=47;  
if Reserpine_flag='X' then label=48;
```



```

if Secobarbital_flag ='X' then label=49;
if Mephobarbital_flag ='X' then label=50;
if Thyroid_flag ='X' then label=51;
if Ticlopidine_flag='X' then label=52;
if Trihexyphenidyl_flag ='X' then label=53;
if Dimenhydrinate_flag ='X' then label=54;
if Trimipramine_flag ='X' then label=55;
if Triprolidine_flag ='X' then label=56;
if Zaleplon_flag ='X' then label=57;
if Zolpidem_flag='X' then label=58;
if Meclizine_flag='X' then label=59;
if Atropine_flag='X' then label=60;
if BelladonnaAlkaloid_flag='X' then label=61;
if Clidinium_CDP_flag='X' then label=62;
if Dicyclomine_flag='X' then label=63;
if Hyoscyamine_flag='X' then label=64;
if Scopolamine_flag='X' then label=65;
if Amoxapine_flag='X' then label=66;
if Nortriptyline_flag='X' then label=67;
if Desipramine_flag='X' then label=68;
if Paroxetine_flag='X' then label=69;
if Propantheline_flag='X' then label=70;
if Protriptyline_flag='X' then label=71;
if Therapeutic_Category='*Non-Benzodiazepine - GABA-Receptor Modulators***'
then nonben=1; else nonben=0;
DrugStrength1=DrugStrength*1;
keep ndc1 label nonben DrugStrength1 Generic_Drug_Name Therapeutic_Category;
run;

```

```

proc sort data=c_hrm1; by ndc1;
run;

```

```

data pde2015hrm;
set aa ;
keep bene_id ndc1 SRVC_DT QTY_DSPNSD_NUM DAYS_SUPLY_NUM;
run;

```

```

proc sort data=pde2015hrm out=a; by ndc1;
run;

```

```

data b;
merge c_hrm1(in=in1) a(in=in2);
by ndc1;
if in1 and in2;
run;

```

```

data ddd;
merge c_hrm1(in=in1) aa(in=in2);
by ndc1;
if in1 and in2;
run;

```

*if multiple prescription claims for the same target medication are dispensed on the same day, keep prescriptions with the longest days supply;

```
proc sort data=b out=c; by bene_id label SRVC_DT descending DAYS_SUPLY_NUM;
run;
```

```
proc sort data=c out=d nodupkey; by bene_id label SRVC_DT ;
run;
```

```
proc sort data=ddd out=dj nodupkey; by bene_id; run;
```

```
data dj1;
set dj;
hrmp_med=1;
keep bene_id hrmp_med;
run;
```

*Identify first group of HRM => greater than two unique prescription dates;

```
data d_zz;
set d;
if label=42 or label= 48 or label=18 or label=22 or (label=27 and nonben=1)
or (label=57 and nonben=1)
or (label=58 and nonben=1) then delete;
run;
```

```
proc freq data=d_zz noprint;
tables bene_id*label/out=e;
run;
```

```
data e1;
set e;
hrm1=1;
hrm_cnt1 = count;
if count>=2;
keep bene_id label hrm_cnt1 hrm1;
run;
```

```
proc sort data=e1 out=d_j; by bene_id hrm1 descending hrm_cnt1; run;
```

```
data d_j1;
set d_j;
by bene_id;
if first.bene_id then output;
run;
```

```
data d_hrm1;
set d_j1;
keep bene_id hrm1 hrm_cnt1;
run;
```

* Identify second group of HRM => For nitrofurantoin, a patient is included in the numerator if he/she received at least two prescription fills for the medication and if the cumulative days supply for any nitrofurantoin product is greater than 90 days during the measurement period;

```
data nitro1;
set d;
if label=42;
```

```
run;
```

```
proc means data=nitro1 noprint;  
class bene_id ;  
var DAYS_SUPLY_NUM;  
output out=nitro2 sum= ;  
run;
```

```
data nitro3;  
set nitro2;  
if DAYS_SUPLY_NUM>90 and _FREQ_>=2;  
if bene_id ne '';  
run;
```

```
data d_hrm2;  
set nitro3;  
hrm2=1;  
hrm_cnt2=_FREQ_;  
keep bene_id hrm2 hrm_cnt2;  
run;
```

* Identify third group of HRM. For reserpine, a patient is included in the numerator if he/she received at least two prescription fills for the medication and if the average daily dose is greater than 0.1mg. More than two counts of average daily dose/per fill constitutes to being over the threshold;

```
data rese1;  
set d;  
if label=48;  
run;
```

```
data rese2;  
  set rese1;  
  by bene_id label;  
  if last.label;  
  count=_n_-sum(lag(_n_),0);  
  do _n_=1 to count;  
    set rese1;  
    output;  
  end;  
run;
```

```
*keep count;  
data rese_ext;  
set rese2;  
hrm_cnt3=count;  
if count>=2;  
run;
```

```
proc sort data=rese_ext out=rese_ext1; by bene_id descending count; run;
```

```
data rese_ext2;  
set rese_ext1;  
by bene_id;
```

```

if first.bene_id then output;
run;

data rese_ext3;
set rese_ext2;
keep bene_id hrm_cnt3;
run;

data rese3;
set rese2;
if count>=2;
drop count;
run;

proc sort data=rese3 out=rese4; by bene_id srvc_dt; run;

data rese5 (drop=ndc1 );
set rese4;
averagedigodose = (QTY_DSPNSD_NUM*DrugStrength1)/DAYS_SUPLY_NUM;
if averagedigodose > 0.1 then count=1; else count=0;
run;

proc means data=rese5 noprint;
class bene_id;
var averagedigodose count;
output out=rese6 sum=;
run;

data rese7;
set rese6;
if bene_id ne '';
if count>=2;
run;

data d_hrm3;
set rese7;
hrm3=1;
keep bene_id hrm3;
run;

proc sort data=d_hrm3; by bene_id; run;
proc sort data=rese_ext3; by bene_id; run;

data d_hrm3_1;
merge d_hrm3(in=in1) rese_ext3;
by bene_id;
if in1;
run;

*Identify fourth group of HRM. For digoxin, a patient is included in the
numerator if he/she received at least two prescription fills for the
medication and if the average daily dose is greater than 0.125mg. More than
two counts of average daily dose/per fill constitutes to being over the
threshold;

data digol;
set d;

```

```

if label=18;
run;

data digo2;
  set digo1;
  by bene_id label;
  if last.label;
  count=_n_-sum(lag(_n_),0);
  do _n_=1 to count;
    set digo1;
    output;
  end;
run;

*keep count;

data digo_ext;
set digo2;
hrm_cnt4=count;
if count>=2;
run;

proc sort data=digo_ext out=digo_ext1; by bene_id descending count; run;

data digo_ext2;
set digo_ext1;
by bene_id;
if first.bene_id then output;
run;

data digo_ext3;
set digo_ext2;
keep bene_id hrm_cnt4;
run;

/*****/

data digo3;
set digo2;
if count>=2;
drop count;
run;

proc sort data=digo3 out=digo4; by bene_id srvc_dt; run;

data digo5 (drop=ndc1 );
set digo4;
averagedigodose = (QTY_DSPNSD_NUM*DrugStrength1)/DAYS_SUPLY_NUM;
if averagedigodose > 0.125 then count=1; else count=0;
run;

proc means data=digo5 noprint;
class bene_id;
var averagedigodose count;
output out=digo6 sum=;
run;

```

```
data digo7;
set digo6;
if bene_id ne '';
if count>=2;
run;
```

```
data d_hrm4;
set digo7;
hrm4=1;
keep bene_id hrm4;
run;
```

```
proc sort data=d_hrm4; by bene_id; run;
proc sort data=digo_ext3; by bene_id; run;
```

```
data d_hrm4_1;
merge d_hrm4(in=in1) digo_ext3;
by bene_id;
if in1;
run;
```

*Identify fifth group of HRM. For doxepin, a patient is included in the numerator if he/she received at least two prescription fills for the medication and if the average daily dose is greater than 6mg.;

```
data doxel;
set d;
if label=22;
run;
```

```
data doxe2;
  set doxel;
  by bene_id label;
  if last.label;
  count=_n_-sum(lag(_n_),0);
  do _n_=1 to count;
    set doxel;
    output;
  end;
run;
```

```
*keep count;
```

```
data doxe_ext;
set doxe2;
hrm_cnt5=count;
if count>=2;
run;
```

```
proc sort data=doxe_ext out=doxe_ext1; by bene_id descending count; run;
```

```
data doxe_ext2;
set doxe_ext1;
by bene_id;
if first.bene_id then output;
run;
```

```

data doxe_ext3;
set doxe_ext2;
keep bene_id hrm_cnt5;
run;

/******/

data doxe3;
set doxe2;
if count>=2;
drop count;
run;

proc sort data=doxe3 out=doxe4; by bene_id srvc_dt; run;

data doxe5 (drop=ndc1 );
set doxe4;
averagedigodose = (QTY_DSPNSD_NUM*DrugStrength1)/DAYS_SUPLY_NUM;
if averagedigodose > 6 then count=1; else count=0;
run;

proc means data=doxe5 noprint;
class bene_id;
var averagedigodose count;
output out=doxe6 sum=;
run;

data doxe7;
set doxe6;
if bene_id ne '';
if count>=2;
run;

data d_hrm5;
set doxe7;
hrm5=1;
keep bene_id hrm5;
run;

proc sort data=d_hrm5; by bene_id; run;
proc sort data=doxe_ext3; by bene_id; run;

data d_hrm5_1;
merge d_hrm5(in=in1) doxe_ext3;
by bene_id;
if in1;
run;

```

*Identify sixth group of HRM. The cumulative calculation applies to the class of nonbenzodiazepine hypnotics and not for each individual medication. A patient is included in the numerator if he/she received at least two prescription fills for any medication in the class and if the cumulative days supply for any product is greater than 90 days during the measurement period. For example, if a patient received a 30 day supply of zolpidem, a second fill for 30 days supply of zolpidem and then a fill for 35 days supply eszopiclone, this would qualify for inclusion in the numerator;

```

data nonben;
set c;
if nonben=1;
run;

proc sort data=nonben out=nonben1; by bene_id nonben SRVC_DT descending
DAYS_SUPLY_NUM;
run;

proc sort data=nonben1 out=nonben2 nodupkey; by bene_id nonben SRVC_DT;
run;

proc means data=nonben2 noprint;
class bene_id ;
var DAYS_SUPLY_NUM;
output out=nonben4 sum= ;
run;

data nonben5;
set nonben4;
if DAYS_SUPLY_NUM>90 and _FREQ_>=2;
if bene_id ne '';
run;

proc sort data=nonben5 out=nonben6 ; by bene_id;
run;

data d_hrm6;
set nonben6;
hrm6=1;
hrm_cnt6=_FREQ_;
keep bene_id hrm6 hrm_cnt6;
run;

*merge all groups;

proc sort data=d_hrm1 nodupkey; by bene_id;run;
proc sort data=d_hrm2 nodupkey; by bene_id;run;
proc sort data=d_hrm3_1 nodupkey; by bene_id;run;
proc sort data=d_hrm4_1 nodupkey; by bene_id;run;
proc sort data=d_hrm5_1 nodupkey; by bene_id;run;
proc sort data=d_hrm6 nodupkey; by bene_id;run;

data d_hrm7;
merge d_hrm1 d_hrm2 d_hrm3_1 d_hrm4_1 d_hrm5_1 d_hrm6;
by bene_id;
run;

data d_hrm8;
set d_hrm7;
if hrm1=. then hrm1=0;
if hrm2=. then hrm2=0;
if hrm3=. then hrm3=0;
if hrm4=. then hrm4=0;
if hrm5=. then hrm5=0;
if hrm6=. then hrm6=0;

```



```

if hrm_cnt1=. then hrm_cnt1=0;
if hrm_cnt2=. then hrm_cnt2=0;
if hrm_cnt3=. then hrm_cnt3=0;
if hrm_cnt4=. then hrm_cnt4=0;
if hrm_cnt5=. then hrm_cnt5=0;
if hrm_cnt6=. then hrm_cnt6=0;
if hrm1=1 or hrm2=1 or hrm3=1 or hrm4=1 or hrm5=1 or hrm6=1 then hrm=1; else
hrm=0;
run;

data d_hrm9;
set d_hrm8;
array values hrm_cnt1-hrm_cnt6;
hrm_maxcnt = max(of values[*]);
run;

proc sort data=d_hrm9 out=d_hrm10 nodupkey; by bene_id; run;

*keep HRM med list;
data hrm_medlist;
set b;
hrm_medlist=1;
keep bene_id hrm_medlist;
run;

proc sort data=hrm_medlist out=hrm_medlist_1 nodupkey; by bene_id; run;

data d_hrm11;
set d_hrm10;
keep bene_id hrm hrm_maxcnt;
run;

proc sort data= finalcohortz1; by bene_id;run;
proc sort data=dj1; by bene_id;run;

data finalcohortz2;
merge finalcohortz1 (in=in1) d_hrm11 hrm_medlist_1 dj1;
by bene_id;
if in1;
if hrm=. then hrm=0;
if hrmp_med=. then hrmp_med=0;
if hrm_maxcnt=. then hrm_maxcnt=0;
if hrm_medlist=. then hrm_medlist=0;
run;

/*****
**PDC for Biguanides (PDC-BG);
*****/

/*Inclusion criteria
1) 18 years and older
2) First prescription occurred at least 91 days before the end of the year
3) At least two prescriptions on two unique dates
4) Exclude ESRD Population
5) Exclude Insulin Users */

/*Determine inclusion outcomes*/

```

```

data drmeds1; set _UPLDS.drmeds; ndc1=ndc*1; drop ndc; run;
proc sort data=drmeds1; by ndc1; run;

data drmeds2;
set drmeds1;
if Biguanide_Flag = 'X' then label=1;
if SFU_Flag = 'X' then label=2;
if TZD_Flag = 'X' then label=3;
if DDPIV_Flag = 'X' then label=4;
run;

data drmedsbg;
set drmeds2;
if label=1;
run;

proc sort data=drmedsbg; by ndc1; run;

data d;
merge aa (in=in1) drmedsbg (in=in2);
by ndc1;
if in1 and in2;
keep bene_id SRVC_DT dr_bg DAYS_SUPLY_NUM;
dr_bg=1;
run;

* Two medications on different dates;
proc sort data=d out=d ; by bene_id ;
run;

proc sort nodupkey data=d out=da; by bene_id; run;

proc sort data=d out=d1 nodupkey; by bene_id SRVC_DT ;
run;
proc freq data=d1 noprint;
table bene_id/out=d2; run;

data d2a;
set d2;
if count>=2;
count_dr_bg = count;
keep bene_id number_dr_bg count_dr_bg;
number_dr_bg=1;
run;

*91 days;
data d3;
set d1;
by bene_id;
if first.bene_id then output;
run;

data d4;
set d3;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_dr_bg;

```

```

duration_dr_bg=1;
run;

*Exclude insulin;
data insulin2; set _UPLDS.insulin; ndc1=ndc*1; drop ndc; run;
data dd;
merge aa(in=in1) insulin2(in=in2);
by ndc1;
if in1 and in2;
keep bene_id insulin_dr;
insulin_dr=1;
run;

proc sort data=dd out=dd1 nodupkey; by bene_id;
run;

* record inclusion outcomes;
data aaa;
merge finalcohortz2 (in=in1) da d2a d4 dd1;
by bene_id;
if in1;
drop SRVC_DT DAYS_SUPLY_NUM;
run;

data finalcohortz3;
set aaa;
if dr_bg=. then dr_bg=0;
if number_dr_bg=. then number_dr_bg=0;
if duration_dr_bg=. then duration_dr_bg=0;
if insulin_dr =. then insulin_dr=0;
if age_BOY>=18 and dr_bg=1 and number_dr_bg=1 and duration_dr_bg=1 and
insulin_dr=0 and esrd_ind="0" then include_dr_bg=1;
else include_dr_bg=0;
run;

/*Calculate PDC (Measure Assessment)*/

data diabetesbg_cx; set _UPLDS.diabetes2017_vxx; ndc1=ndc*1; drop ndc; run;

data diabetesbg_cx1;
set diabetesbg_cx;
if Metformin_Flag ='X' then labell1=1; else labell1=.;
keep ndc1 labell1;
run;

proc sort data=diabetesbg_cx1; by ndc1; run;

proc transpose data=diabetesbg_cx1 out=diabetesbg_cx2;
by ndc1;
run;

data diabetesbg_cx3;
set diabetesbg_cx2;
labels=coll1;
drop _name_ coll1;
if labels ne .;

```

```

run;

proc sql;
create table want as
select aa.*, diabetesbg_cx3.labels
from aa, diabetesbg_cx3
where aa.ndc1 = diabetesbg_cx3.ndc1;
quit;

data want1;
set want;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM;
run;

proc sort data=want1; by bene_id labels SRVC_DT ; run;

data want2;
do until (last.labels);
set want1 ;
by bene_id labels;
SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;
output ;
end;
format SRVC_END DATE9. ;
run;

data want3;
set want2;
by bene_id labels ;
format adjStart adjEnd lastdate mmdyy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd = SRVC_DT + DAYS_SUPLY_NUM -1;
if first.labels then lastdate = .;
else do ;
** If overlap then adjust start/end dates, else keep as is;
if adjStart <= lastdate then do;
adjStart = lastdate + 1;
adjEnd = adjStart + DAYS_SUPLY_NUM -1 ;
end;
end;
** Save last ending date;
lastdate = adjEnd;
run;

proc sort data=want3 out=want4 nodupkey; by bene_id adjStart descending
DAYS_SUPLY_NUM;
run;

proc sort data=want4 out=d_diadr nodupkey;
by bene_id adjStart;
run;

data aa_d1; set d_diadr;
member_id=bene_id;
fill_dt=adjStart;
format fill_dt DDMYY10. ;

```

```

days_supply=DAYS_SUPLY_NUM;
run;

data aa_d2; set aa_d1;
keep member_id fill_dt days_supply; run;

data claims_dr_bg; set aa_d2;
run;

/*Step 1*/

proc sort data=claims_dr_bg; by member_id fill_dt;
run;

proc transpose data=claims_dr_bg out=fill_dates (drop=_name_) prefix=fill_dt;
by member_id;
var fill_dt;
run;

proc transpose data=claims_dr_bg out=days_supply (drop=_name_)
prefix=days_supply;
by member_id;
var days_supply;
run;

data both;
merge fill_dates days_supply;
by member_id;
format start_dt end_dt mmddyy10. ;
start_dt=fill_dt1;
end_dt='31DEC2017'd;
run;

/*Step 2*/
data pdc_dr_bg;
set both;
array daydummy(365) day1-day365;
serv_days=end_dt-start_dt+1;
array filldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
    do i=1 to dim(filldates) while (filldates(i) ne .);
        if filldates(i)<=first_dt+ii-1<=filldates(i)+days_supply(i)-1
            then daydummy(ii)=1;
    end;
end;
drop i ii ;
dayscovered=sum(of daydummy[*]);
label dayscovered='Total Days Covered';
p_dayscovered=dayscovered/serv_days;
label p_dayscovered='Proportion of Days Covered';
drop fill_dt1-fill_dt365 days_supply1-days_supply365;
run;

```

```

data pdc_dr_bg_1;
set pdc_dr_bg;
bene_id=member_id;
pdc_dr_bg = p_dayscovered;
sd_dr_bg = serv_days;
dc_dr_bg = dayscovered;
keep bene_id pdc_dr_bg sd_dr_bg dc_dr_bg;
run;

proc sort data=pdc_dr_bg_1; by bene_id; run;
proc sort data= finalcohortz3; by bene_id; run;

data finalcohortz4;
merge finalcohortz3 (in=in1) pdc_dr_bg_1;
by bene_id;
if in1;
run;

/*****
**PDC for Sulfonylureas (PDC-SFU);
*****/

/*Inclusion criteria
1) 18 years and older
2) First prescription occurred at least 91 days before the end of the year
3) At least two prescriptions on two unique dates
4) Exclude ESRD Population
5) Insulin Users */

/*Determine inclusion outcomes*/

data drmedssfu;
set drmeds2;
if label=2;
run;

proc sort data=drmedssfu; by ndc1; run;

data d;
merge aa (in=in1) drmedssfu (in=in2);
by ndc1;
if in1 and in2;
keep bene_id SRVC_DT dr_sfu DAYS_SUPLY_NUM;
dr_sfu=1;
run;

* Two medications on different dates;
proc sort data=d out=d ; by bene_id ;
run;

proc sort nodupkey data=d out=da; by bene_id; run;

proc sort data=d out=d1 nodupkey; by bene_id SRVC_DT ;
run;
proc freq data=d1 noprint;

```

```

table bene_id/out=d2; run;

data d2a;
set d2;
if count>=2;
count_dr_sfu = count;
keep bene_id number_dr_sfu count_dr_sfu;
number_dr_sfu=1;
run;

*91 days of Treatment;
data d3;
set d1;
by bene_id;
if first.bene_id then output;
run;

data d4;
set d3;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_dr_sfu;
duration_dr_sfu=1;
run;

proc sort data=finalcohortz4; by bene_id;run;

* Only include records that meets the inclusion criteria;
data aaa;
merge finalcohortz4 (in=in1) da d2a d4 ;
by bene_id;
if in1;
drop SRVC_DT DAYS_SUPLY_NUM;
run;

data finalcohortz5;
set aaa;
if dr_sfu=. then dr_sfu=0;
if number_dr_sfu=. then number_dr_sfu=0;
if duration_dr_sfu=. then duration_dr_sfu=0;
if age_BOY>=18 and dr_sfu=1 and number_dr_sfu=1 and duration_dr_sfu=1 and
insulin_dr=0 and esrd_ind="0" then include_dr_sfu=1;
else include_dr_sfu=0;
run;

/*Calculate PDC (Measure Assessment)*/

data diabetessfu_cx; set _UPLDS.diabetes2017_vxx; ndc1=ndc*1; drop ndc; run;

data diabetessfu_cx1;
set diabetessfu_cx;
if Chlorpropamide_Flag ='X' then label1=1; else label1=.;
if Glimepiride_Flag ='X' then label2=2; else label2=.;
if Glipizide_Flag ='X' then label3=3; else label3=.;
if Glyburide_Flag ='X' then label4=4; else label4=.;
if Tolazamide_Flag ='X' then label5=5; else label5=.;
if Tolbutamide_Flag ='X' then label6=6; else label6=.;

```

```

keep ndc1 label1-label6;
run;

proc sort data=diabetessfu_cx1; by ndc1; run;

proc transpose data=diabetessfu_cx1 out=diabetessfu_cx2;
  by ndc1;
run;

data diabetessfu_cx3;
set diabetessfu_cx2;
labels=coll;
drop _name_ coll;
if labels ne .;
run;

proc sql;
create table want as
select aa.*, diabetessfu_cx3.labels
from aa, diabetessfu_cx3
where aa.ndc1 = diabetessfu_cx3.ndc1;
quit;

data want1;
set want;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM;
run;

proc sort data=want1; by bene_id labels SRVC_DT ; run;

data want2;
do until (last.labels);
  set want1 ;
  by bene_id labels;
  SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;
  output ;
end;
format SRVC_END DATE9. ;
run;

data want3;
set want2;
by bene_id labels ;
format adjStart adjEnd lastdate mmddyy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd   = SRVC_DT + DAYS_SUPLY_NUM -1;
  if first.labels then lastdate = .;
  else do ;
** If overlap then adjust start/end dates, else keep as is;
  if adjStart <= lastdate then do;
    adjStart = lastdate + 1;
    adjEnd   = adjStart + DAYS_SUPLY_NUM -1 ;
  end;
end;
** Save last ending date;

```



```

lastdate = adjEnd;
run;

proc sort data=want3 out=want4 nodupkey; by bene_id adjStart descending
DAYS_SUPLY_NUM;
run;

proc sort data=want4 out=d_diasfu nodupkey;
by bene_id adjStart;
run;

data aa_d1; set d_diasfu;
member_id=bene_id;
fill_dt=adjStart;
format fill_dt DDMMYY10. ;
days_supply=DAYS_SUPLY_NUM;
run;

data aa_d2; set aa_d1;
keep member_id fill_dt days_supply; run;

data claims_dr_sfu; set aa_d2;
run;

/*Step 1*/

proc sort data=claims_dr_sfu; by member_id fill_dt;
run;

proc transpose data=claims_dr_sfu out=fill_dates (drop=_name_)
prefix=fill_dt;
by member_id;
var fill_dt;
run;

proc transpose data=claims_dr_sfu out=days_supply (drop=_name_)
prefix=days_supply;
by member_id;
var days_supply;
run;

data both;
merge fill_dates days_supply;
by member_id;
format start_dt end_dt mmddyy10. ;
start_dt=fill_dt1;
end_dt='31DEC2017'd;
run;

/*Step 2*/
data pdc_dr_sfu;
set both;
array daydummy(365) day1-day365;
serv_days=end_dt-start_dt+1;
array filldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;

```

```

do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
    do i=1 to dim(filldates) while (filldates(i) ne .);
        if filldates(i)<=first_dt+ii-1<=filldates(i)+days_supply(i)-1
            then daydummy(ii)=1;
    end;
end;
drop i ii ;
dayscovered=sum(of daydummy[*]);
label dayscovered='Total Days Covered';
p_dayscovered=dayscovered/serv_days;
label p_dayscovered='Proportion of Days Covered';
drop fill_dt1-fill_dt365 days_supply1-days_supply365;
run;

data pdc_dr_sfu_1;
set pdc_dr_sfu;
bene_id=member_id;
pdc_dr_sfu = p_dayscovered;
sd_dr_sfu = serv_days;
dc_dr_sfu = dayscovered;
keep bene_id pdc_dr_sfu sd_dr_sfu dc_dr_sfu;
run;

proc sort data=pdc_dr_sfu_1; by bene_id; run;
proc sort data= finalcohortz5; by bene_id; run;

data finalcohortz6;
merge finalcohortz5 (in=in1) pdc_dr_sfu_1;
by bene_id;
if in1;
run;

/*****
**PDC for Thiazolidinediones (PDC-TZD);
*****/

/*Inclusion criteria
1) 18 years and older
2) First prescription occurred at least 91 days before the end of the year
3) At least two prescriptions on two unique dates
4) Exclude ESRD Population
5) Exclude Insulin Users */

/* Determine inclusion outcomes*/

data drmedstzd;
set drmeds2;
if label=3;
run;

proc sort data=drmedstzd; by ndc1; run;

data d;
merge aa (in=in1) drmedstzd (in=in2);

```

```

by ndc1;
if in1 and in2;
keep bene_id SRVC_DT dr_tzd DAYS_SUPLY_NUM;
dr_tzd=1;
run;

* Two medications on different dates;
proc sort data=d out=d ; by bene_id ;
run;

proc sort nodupkey data=d out=da; by bene_id; run;

proc sort data=d out=d1 nodupkey; by bene_id SRVC_DT ;
run;
proc freq data=d1 noprint;
table bene_id/out=d2; run;

data d2a;
set d2;
if count>=2;
count_dr_tzd = count;
keep bene_id number_dr_tzd count_dr_tzd;
number_dr_tzd=1;
run;

*91 days;
data d3;
set d1;
by bene_id;
if first.bene_id then output;
run;

data d4;
set d3;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_dr_tzd;
duration_dr_tzd=1;
run;

proc sort data=finalcohortz6; by bene_id;run;

* Only include records that meets the inclusion criteria;
data aaa;
merge finalcohortz6 (in=in1) da d2a d4 ;
by bene_id;
if in1;
drop SRVC_DT DAYS_SUPLY_NUM;
run;

data finalcohortz7;
set aaa;
if dr_tzd=. then dr_tzd=0;
if number_dr_tzd=. then number_dr_tzd=0;
if duration_dr_tzd=. then duration_dr_tzd=0;
if age_BOY>=18 and dr_tzd=1 and number_dr_tzd=1 and duration_dr_tzd=1 and
insulin_dr=0 and esrd_ind="0" then include_dr_tzd=1;
else include_dr_tzd=0;

```

```

run;

/*Calculate PDC (Measure Assessment)*/

data diabetestzd_cx; set _UPLDS.diabetes2017_vxx; ndc1=ndc*1; drop ndc; run;

data diabetestzd_cx1;
set diabetestzd_cx;
if Pioglitazone_Flag ='X' then label1=1; else label1=.;
if Rosiglitazone_Flag ='X' then label2=2; else label2=.;
keep ndc1 label1-label2;
run;

proc sort data=diabetestzd_cx1; by ndc1; run;

proc transpose data=diabetestzd_cx1 out=diabetestzd_cx2;
by ndc1;
run;

data diabetestzd_cx3;
set diabetestzd_cx2;
labels=col1;
drop _name_ col1;
if labels ne .;
run;

proc sql;
create table want as
select aa.*, diabetestzd_cx3.labels
from aa, diabetestzd_cx3
where aa.ndc1 = diabetestzd_cx3.ndc1;
quit;

data want1;
set want;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM;
run;

proc sort data=want1; by bene_id labels SRVC_DT ; run;

data want2;
do until (last.labels);
set want1 ;
by bene_id labels;
SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;
output ;
end;
format SRVC_END DATE9. ;
run;

data want3;
set want2;
by bene_id labels ;
format adjStart adjEnd lastdate mmddy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd = SRVC_DT + DAYS_SUPLY_NUM -1;

```

```

    if first.labels then lastdate = .;
    else do ;
** If overlap then adjust start/end dates, else keep as is;
    if adjStart <= lastdate then do;
        adjStart = lastdate + 1;
        adjEnd    = adjStart + DAYS_SUPLY_NUM -1 ;
    end;
    end;
** Save last ending date;
lastdate = adjEnd;
run;

proc sort data=want3 out=want4 nodupkey; by bene_id adjStart descending
DAYS_SUPLY_NUM;
run;

proc sort data=want4 out=d_diatzd nodupkey;
by bene_id adjStart;
run;

data aa_d1; set d_diatzd;
member_id=bene_id;
fill_dt=adjStart;
format fill_dt DDMYY10. ;
days_supply=DAYS_SUPLY_NUM;
run;

data aa_d2; set aa_d1;
keep member_id fill_dt days_supply; run;

data claims_dr_tzd; set aa_d2;
run;

/*Step 1*/

proc sort data=claims_dr_tzd; by member_id fill_dt;
run;

proc transpose data=claims_dr_tzd out=fill_dates (drop=_name_)
prefix=fill_dt;
by member_id;
var fill_dt;
run;

proc transpose data=claims_dr_tzd out=days_supply (drop=_name_)
prefix=days_supply;
by member_id;
var days_supply;
run;

data both;
merge fill_dates days_supply;
by member_id;
format start_dt end_dt mmddyy10. ;
start_dt=fill_dt1;
end_dt='31DEC2017'd;
run;

```

```

/*Step 2*/
data pdc_dr_tzd;
set both;
array daydummy(365) day1-day365;
serv_days=end_dt-start_dt+1;
array filldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
    do i=1 to dim(filldates) while (filldates(i) ne .);
        if filldates(i)<=first_dt+ii-1<=filldates(i)+days_supply(i)-1
            then daydummy(ii)=1;
        end;
    end;
end;
drop i ii ;
dayscovered=sum(of daydummy[*]);
label dayscovered='Total Days Covered';
p_dayscovered=dayscovered/serv_days;
label p_dayscovered='Proportion of Days Covered';
drop fill_dt1-fill_dt365 days_supply1-days_supply365;
run;

data pdc_dr_tzd_1;
set pdc_dr_tzd;
bene_id=member_id;
pdc_dr_tzd = p_dayscovered;
sd_dr_tzd = serv_days;
dc_dr_tzd = dayscovered;
keep bene_id pdc_dr_tzd sd_dr_tzd dc_dr_tzd;
run;

proc sort data=pdc_dr_tzd_1; by bene_id; run;
proc sort data= finalcohortz7; by bene_id; run;

data finalcohortz8;
merge finalcohortz7 (in=in1) pdc_dr_tzd_1;
by bene_id;
if in1;
run;

/*****
*PDC for DPP-4 Inhibitors (PDC-DPP);
*****/

/*Inclusion criteria
1) 18 years and older
2) First prescription occurred at least 91 days before the end of the year
3) At least two prescriptions on two unique dates
4) Exclude ESRD Population
5) Exclude Insulin Users */

/* Determine inclusion outcomes */

```

```

data drmedsdpp;
set drmeds2;
if label=4;
run;

proc sort data=drmedsdpp; by ndc1; run;

data d;
merge aa (in=in1) drmedsdpp (in=in2);
by ndc1;
if in1 and in2;
keep bene_id SRVC_DT dr_dpp DAYS_SUPLY_NUM;
dr_dpp=1;
run;

* Two medications on different dates;
proc sort data=d out=d ; by bene_id ;
run;

proc sort nodupkey data=d out=da; by bene_id; run;

proc sort data=d out=d1 nodupkey; by bene_id SRVC_DT ;
run;
proc freq data=d1 noprint;
table bene_id/out=d2; run;

data d2a;
set d2;
if count>=2;
count_dr_dpp = count;
keep bene_id number_dr_dpp count_dr_dpp;
number_dr_dpp=1;
run;

*91 days;

data d3;
set d1;
by bene_id;
if first.bene_id then output;
run;

data d4;
set d3;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_dr_dpp;
duration_dr_dpp=1;
run;

proc sort data=finalcohortz8; by bene_id; run;

* Only include records that meets the inclusion criteria;
data aaa;
merge finalcohortz8 (in=in1) da d2a d4 ;
by bene_id;
if in1;
drop SRVC_DT DAYS_SUPLY_NUM;

```

```

run;

data finalcohortz9;
set aaa;
if dr_dpp=. then dr_dpp=0;
if number_dr_dpp=. then number_dr_dpp=0;
if duration_dr_dpp=. then duration_dr_dpp=0;
if age_boy>= 18 and dr_dpp=1 and number_dr_dpp=1 and duration_dr_dpp=1 and
insulin_dr=0 and esrd_ind="0" then include_dr_dpp=1;
else include_dr_dpp=0;
run;

/*Calculate PDC (Measure Assessment)*/

data diabetesdpp_cx; set _UPLDS.diabetes2017_vxx; ndc1=ndc*1; drop ndc; run;

data diabetesdpp_cx1;
set diabetesdpp_cx;
if Sitagliptin_Flag ='X' then label1=1; else label1=.;
if Linagliptin_Flag ='X' then label2=2; else label2=.;
if Saxagliptin_Flag ='X' then label3=3; else label3=.;
if Alogliptin_Flag ='X' then label4=4; else label4=.;
keep ndc1 label1-label4;
run;

proc sort data=diabetesdpp_cx1; by ndc1; run;

proc transpose data=diabetesdpp_cx1 out=diabetesdpp_cx2;
by ndc1;
run;

data diabetesdpp_cx3;
set diabetesdpp_cx2;
labels=coll1;
drop _name_ coll1;
if labels ne .;
run;

proc sql;
create table want as
select aa.*, diabetesdpp_cx3.labels
from aa, diabetesdpp_cx3
where aa.ndc1 = diabetesdpp_cx3.ndc1;
quit;

data want1;
set want;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM;
run;

proc sort data=want1; by bene_id labels SRVC_DT ; run;

data want2;
do until (last.labels);
set want1 ;
by bene_id labels;
SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;

```



```

    output ;
    end;
    format SRVC_END DATE9. ;
run;

data want3;
set want2;
by bene_id labels ;
format adjStart adjEnd lastdate mmddyy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd   = SRVC_DT + DAYS_SUPLY_NUM -1;
    if first.labels then lastdate = .;
    else do ;
** If overlap then adjust start/end dates, else keep as is;
    if adjStart <= lastdate then do;
        adjStart = lastdate + 1;
        adjEnd   = adjStart + DAYS_SUPLY_NUM -1 ;
    end;
    end;
** Save last ending date;
lastdate = adjEnd;
run;

proc sort data=want3 out=want4 nodupkey; by bene_id adjStart descending
DAYS_SUPLY_NUM;
run;

proc sort data=want4 out=d_diadpp nodupkey;
by bene_id adjStart;
run;

data aa_d1; set d_diadpp;
member_id=bene_id;
fill_dt=adjStart;
format fill_dt DDMMYY10. ;
days_supply=DAYS_SUPLY_NUM;
run;

data aa_d2; set aa_d1;
keep member_id fill_dt days_supply; run;

data claims_dr_dpp; set aa_d2;
run;

/*Step 1*/

proc sort data=claims_dr_dpp; by member_id fill_dt;
run;

proc transpose data=claims_dr_dpp out=fill_dates (drop=_name_)
prefix=fill_dt;
by member_id;
var fill_dt;
run;

```

```

proc transpose data=claims_dr_dpp out=days_supply (drop=_name_)
prefix=days_supply;
by member_id;
var days_supply;
run;

data both;
merge fill_dates days_supply;
by member_id;
format start_dt end_dt mmddyy10. ;
start_dt=fill_dt1;
end_dt='31DEC2017'd;
run;

/*Step 2*/
data pdc_dr_dpp;
set both;
array daydummy(365) day1-day365;
serv_days=end_dt-start_dt+1;
array filldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
    do i=1 to dim(filldates) while (filldates(i) ne .);
        if filldates(i)<=first_dt+ii-1<=filldates(i)+days_supply(i)-1
            then daydummy(ii)=1;
    end;
end;
drop i ii ;
dayscovered=sum(of daydummy[*]);
label dayscovered='Total Days Covered';
p_dayscovered=dayscovered/serv_days;
label p_dayscovered='Proportion of Days Covered';
drop fill_dt1-fill_dt365 days_supply1-days_supply365;
run;

data pdc_dr_dpp_1;
set pdc_dr_dpp;
bene_id=member_id;
pdc_dr_dpp = p_dayscovered;
sd_dr_dpp = serv_days;
dc_dr_dpp = dayscovered;
keep bene_id pdc_dr_dpp sd_dr_dpp dc_dr_dpp;
run;

proc sort data=pdc_dr_dpp_1; by bene_id; run;
proc sort data= finalcohortz9; by bene_id; run;

data finalcohortz10;
merge finalcohortz9 (in=in1) pdc_dr_dpp_1;
by bene_id;
if in1;
run;

```

```

/*****
*PDC for Beta-Blockers (PDC-BB)*;
*****/

/*Inclusion criteria
1) 18 years and older
2) First prescription occurred at least 91 days before the end of the year
3) At least two prescriptions on two unique dates*/

/*Determine inclusion outcomes*/

data medsbb1; set _UPLDS.medsbb; ndc1=ndc*1; drop ndc; run;
proc sort data=medsbb1; by ndc1; run;

data d;
merge aa (in=in1) medsbb1 (in=in2);
by ndc1;
if in1 and in2;
keep bene_id SRVC_DT med_bb DAYS_SUPLY_NUM;
med_bb=1;
run;

* Two medications on different dates;
proc sort data=d out=d ; by bene_id ;
run;

proc sort nodupkey data=d out=da; by bene_id; run;

proc sort data=d out=d1 nodupkey; by bene_id SRVC_DT ;
run;
proc freq data=d1 noprint;
table bene_id/out=d2; run;

data d2a;
set d2;
if count>=2;
count_med_bb = count;
keep bene_id number_med_bb count_med_bb;
number_med_bb=1;
run;

*91 days of Treatment;
data d3;
set d1;
by bene_id;
if first.bene_id then output;
run;

data d4;
set d3;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_med_bb;
duration_med_bb=1;
run;

```

```

proc sort data=finalcohortz10; by bene_id; run;

* Record inclusion outcomes;
data aaa;
merge finalcohortz10 (in=in1) da d2a d4 ;
by bene_id;
if in1;
drop SRVC_DT DAYS_SUPLY_NUM;
run;

data finalcohortz11;
set aaa;
if med_bb=. then med_bb=0;
if number_med_bb=. then number_med_bb=0;
if duration_med_bb=. then duration_med_bb=0;
if age_BOY>= 18 and med_bb=1 and number_med_bb=1 and duration_med_bb=1 then
include_med_bb=1; else include_med_bb=0;
run;

/*Calculate PDC (Measure Assessment)*/

data medsbb_cx; set _UPLDS.bb2017_vxx; ndc1=ndc*1; drop ndc; run;

data medsbb_cx1;
set medsbb_cx;
if Acebutolol_Flag ='X' then label1=1; else label1=.;
if Atenolol_Flag ='X' then label2=2; else label2=.;
if Betaxolol_Flag ='X' then label3=3; else label3=.;
if Bisoprolol_Flag ='X' then label4=4; else label4=.;
if Carvedilol_Flag ='X' then label5=5; else label5=.;
if Labetalol_Flag ='X' then label6=6; else label6=.;
if Metoprolol_Flag ='X' then label7=7; else label7=.;
if Nebivolol_Flag ='X' then label8=8; else label8=.;
if Penbutolol_Flag ='X' then label9=9; else label9=.;
if Pindolol_Flag ='X' then label10=10; else label10=.;
if Propranolol_Flag ='X' then label11=11; else label11=.;
if Timolol_Flag ='X' then label12=12; else label12=.;
if Nadolol_Flag ='X' then label13=13; else label13=.;

keep ndc1 label1-label13;
run;

proc sort data=medsbb_cx1; by ndc1; run;

proc transpose data=medsbb_cx1 out=medsbb_cx2;
by ndc1;
run;

data medsbb_cx3;
set medsbb_cx2;
labels=coll;
drop _name_ coll;
if labels ne .;
run;

proc sql;

```

```

create table want as
select aa.*, medsbb_cx3.labels
from aa, medsbb_cx3
where aa.ndc1 = medsbb_cx3.ndc1;
quit;

data want1;
set want;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM;
run;

proc sort data=want1; by bene_id labels SRVC_DT ; run;

data want2;
do until (last.labels);
  set want1 ;
  by bene_id labels;
  SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;
  output ;
end;
format SRVC_END DATE9. ;
run;

data want3;
set want2;
by bene_id labels ;
format adjStart adjEnd lastdate mmddyy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd   = SRVC_DT + DAYS_SUPLY_NUM -1;
  if first.labels then lastdate = .;
  else do ;
** If overlap then adjust start/end dates, else keep as is;
  if adjStart <= lastdate then do;
    adjStart = lastdate + 1;
    adjEnd   = adjStart + DAYS_SUPLY_NUM -1 ;
  end;
end;
** Save last ending date;
lastdate = adjEnd;
run;

proc sort data=want3 out=want4 nodupkey; by bene_id adjStart descending
DAYS_SUPLY_NUM;
run;

proc sort data=want4 out=d_bb nodupkey;
by bene_id adjStart;
run;

data aa_d1; set d_bb;
member_id=bene_id;
fill_dt=adjStart;
format fill_dt DDMMYY10. ;
days_supply=DAYS_SUPLY_NUM;
run;

```

```

data aa_d2; set aa_d1;
keep member_id fill_dt days_supply; run;

data claims_med_bb; set aa_d2;
run;

/*Step 1*/

proc sort data=claims_med_bb; by member_id fill_dt;
run;

proc transpose data=claims_med_bb out=fill_dates (drop=_name_)
prefix=fill_dt;
by member_id;
var fill_dt;
run;

proc transpose data=claims_med_bb out=days_supply (drop=_name_)
prefix=days_supply;
by member_id;
var days_supply;
run;

data both;
merge fill_dates days_supply;
by member_id;
format start_dt end_dt mmddyy10. ;
start_dt=fill_dt1;
end_dt='31DEC2017'd;
run;

/*Step 2*/
data pdc_med_bb;
set both;
array daydummy(365) day1-day365;
serv_days=end_dt-start_dt+1;
array filldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
    do i=1 to dim(filldates) while (filldates(i) ne .);
        if filldates(i) <= first_dt + ii - 1 <= filldates(i) + days_supply(ii) - 1
            then daydummy(ii) = 1;
    end;
end;
drop i ii ;
dayscovered=sum(of daydummy[*]);
label dayscovered='Total Days Covered';
p_dayscovered=dayscovered/serv_days;
label p_dayscovered='Proportion of Days Covered';
drop fill_dt1-fill_dt365 days_supply1-days_supply365;
run;

data pdc_med_bb_1;

```

```

set pdc_med_bb;
bene_id=member_id;
pdc_med_bb = p_dayscovered;
sd_med_bb = serv_days;
dc_med_bb = dayscovered;
keep bene_id pdc_med_bb sd_med_bb dc_med_bb;
run;

proc sort data=pdc_med_bb_1; by bene_id; run;
proc sort data= finalcohortz11; by bene_id; run;

data finalcohortz12;
merge finalcohortz11 (in=in1) pdc_med_bb_1;
by bene_id;
if in1;
run;

/*****
*PDC for Calcium Channel Blockers (PDC-CBB)*
*****/

/*Inclusion criteria
1) 18 years and older
2) First prescription occurred at least 91 days before the end of the year
3) At least two prescriptions on two unique dates*/

/* Determine inclusion outcomes*/

data medscbb1; set _UPLDS.medscbb; ndc1=ndc*1; drop ndc; run;
proc sort data=medscbb1; by ndc1; run;

data d;
merge aa (in=in1) medscbb1 (in=in2);
by ndc1;
if in1 and in2;
keep bene_id SRVC_DT med_cbb DAYS_SUPLY_NUM;
med_cbb=1;
run;

* Two medications on different dates;
proc sort data=d out=d ; by bene_id ;
run;

proc sort nodupkey data=d out=da; by bene_id; run;

proc sort data=d out=d1 nodupkey; by bene_id SRVC_DT ;
run;
proc freq data=d1 noprint;
table bene_id/out=d2; run;

data d2a;
set d2;
if count>=2;
count_med_cbb = count;
keep bene_id number_med_cbb count_med_cbb;

```

```

number_med_cbb=1;
run;

*91 days;
data d3;
set d1;
by bene_id;
if first.bene_id then output;
run;

data d4;
set d3;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_med_cbb;
duration_med_cbb=1;
run;

proc sort data=finalcohortz12; by bene_id; run;

* Record inclusion outcomes;
data aaa;
merge finalcohortz12 (in=in1) da d2a d4 ;
by bene_id;
if in1;
drop SRVC_DT DAYS_SUPLY_NUM;
run;

data finalcohortz13;
set aaa;
if med_cbb=. then med_cbb=0;
if number_med_cbb=. then number_med_cbb=0;
if duration_med_cbb=. then duration_med_cbb=0;
if age_BOY>=18 and med_cbb=1 and number_med_cbb=1 and duration_med_cbb=1 then
include_med_cbb=1; else include_med_cbb=0;
run;

/*Calculate PDC (Measure Assessment)*/

data medsccb_cx; set _UPLDS.ccb2017_vxx; ndc1=ndc*1; drop ndc; run;

data medsccb_cx1;
set medsccb_cx;
if Amlodipine_Flag ='X' then label1=1; else label1=.;
if Diltiazem_Flag ='X' then label2=2; else label2=.;
if Felodipine_Flag ='X' then label3=3; else label3=.;
if Isradipine_Flag ='X' then label4=4; else label4=.;
if Nicardipine_Flag ='X' then label5=5; else label5=.;
if Nifedipine_Flag ='X' then label6=6; else label6=.;
if Nisoldipine_Flag ='X' then label7=7; else label7=.;
if Verapamil_Flag ='X' then label8=8; else label8=.;

keep ndc1 label1-label8;
run;

proc sort data=medsccb_cx1; by ndc1; run;

```



```

proc transpose data=medsccb_cx1 out=medsccb_cx2;
  by ndc1;
run;

data medsccb_cx3;
set medsccb_cx2;
labels=coll1;
drop _name_coll1;
if labels ne .;
run;

proc sql;
create table want as
select aa.*, medsccb_cx3.labels
from aa, medsccb_cx3
where aa.ndc1 = medsccb_cx3.ndc1;
quit;

data want1;
set want;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM;
run;

proc sort data=want1; by bene_id labels SRVC_DT ; run;

data want2;
do until (last.labels);
  set want1 ;
  by bene_id labels;
  SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;
  output ;
end;
format SRVC_END DATE9. ;
run;

data want3;
set want2;
by bene_id labels ;
format adjStart adjEnd lastdate mmdyy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd   = SRVC_DT + DAYS_SUPLY_NUM -1;
  if first.labels then lastdate = .;
  else do ;
** If overlap then adjust start/end dates, else keep as is;
  if adjStart <= lastdate then do;
    adjStart = lastdate + 1;
    adjEnd   = adjStart + DAYS_SUPLY_NUM -1 ;
  end;
end;
** Save last ending date;
lastdate = adjEnd;
run;

proc sort data=want3 out=want4 nodupkey; by bene_id adjStart descending
DAYS_SUPLY_NUM;

```

```

run;

proc sort data=want4 out=d_ccb nodupkey;
by bene_id adjStart;
run;

data aa_d1; set d_ccb;
member_id=bene_id;
fill_dt=adjStart;
format fill_dt DDMMYY10. ;
days_supply=DAYS_SUPLY_NUM;
run;

data aa_d2; set aa_d1;
keep member_id fill_dt days_supply; run;

data claims_med_cbb; set aa_d2;
run;

/*Step 1*/

proc sort data=claims_med_cbb; by member_id fill_dt;
run;

proc transpose data=claims_med_cbb out=fill_dates (drop=_name_)
prefix=fill_dt;
by member_id;
var fill_dt;
run;

proc transpose data=claims_med_cbb out=days_supply (drop=_name_)
prefix=days_supply;
by member_id;
var days_supply;
run;

data both;
merge fill_dates days_supply;
by member_id;
format start_dt end_dt mmddyy10. ;
start_dt=fill_dt1;
end_dt='31DEC2017'd;
run;

/*Step 2*/
data pdc_med_cbb;
set both;
array daydummy(365) day1-day365;
serv_days=end_dt-start_dt+1;
array fillldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
do i=1 to dim(fillldates) while (fillldates(i) ne .);

```

```

        if filldates(i) <= first_dt + ii - 1 <= filldates(i) + days_supply(i) - 1
        then daydummy(ii) = 1;
    end;
end;
drop i ii ;
dayscovered = sum(of daydummy[*]);
label dayscovered = 'Total Days Covered';
p_dayscovered = dayscovered / serv_days;
label p_dayscovered = 'Proportion of Days Covered';
drop fill_dt1 - fill_dt365 days_supply1 - days_supply365;
run;

data pdc_med_cbb_1;
set pdc_med_cbb;
bene_id = member_id;
pdc_med_cbb = p_dayscovered;
sd_med_cbb = serv_days;
dc_med_cbb = dayscovered;
keep bene_id pdc_med_cbb sd_med_cbb dc_med_cbb;
run;

proc sort data = pdc_med_cbb_1; by bene_id; run;
proc sort data = finalcohortz13; by bene_id; run;

data finalcohortz14;
merge finalcohortz13 (in=in1) pdc_med_cbb_1;
by bene_id;
if in1;
run;

data star2017unr_finalcohortz_p1;
set finalcohortz14;
run;

/*****
*PDC for Non-Warfarin Oral Anticoagulants (PDC-NOAC)*;
*****/

/*Inclusion criteria
1) Users of NOAC-A
2) 18 years and older
3) At least two prescriptions for a NOAC on 2 unique dates that are 180 days
apart
4) Received greater than 60 days of medication
5) Exclusion of NOAC-B Users;*/

/* Determine inclusion outcomes*/

data mednoacal; set _UPLDS.med_noaca; ndc1 = ndc * 1; drop ndc; run;
proc sort data = mednoacal; by ndc1; run;

data d;
merge aa (in=in1) mednoacal (in=in2);
by ndc1;
if in1 and in2;

```

```

keep bene_id SRVC_DT med_noaca DAYS_SUPPLY_NUM;
med_noaca=1;
run;

* Two medications on different dates;
proc sort data=d out=d ; by bene_id ;
run;

proc sort nodupkey data=d out=da; by bene_id; run;

proc sort data=d out=d1 nodupkey; by bene_id SRVC_DT ;
run;

proc freq data=d1 noprint;
table bene_id/out=d2; run;

data d2a;
set d2;
if count>=2;
count_med_noaca = count;
keep bene_id number_med_noaca count_med_noaca;
number_med_noaca=1;
run;

*2 unique service dates that are 180 days apart;

proc sort data=d1 out=d3; by bene_id SRVC_DT DAYS_SUPPLY_NUM;
run;

data d4 (drop=x y);
do until (last.bene_id);
set d3;
by bene_id;
if first.bene_id then x=SRVC_DT;
else if last.bene_id then y=SRVC_DT;
end;
do until (last.bene_id);
set d3;
by bene_id;
daysapt_noaca = y-x;
output;
end;
run;

data d5;
set d4;
by bene_id;
if first.bene_id then output;
run;

data d6;
set d5;
if daysapt_noaca >= 180 then duration_noaca = 1; else duration_noaca = 0;
keep bene_id duration_noaca daysapt_noaca;
run;

*greater than 60 days of supply;

```

```

data d7 ;
  set d1;
  by bene_id;
  if first.bene_id then ttlsupply_noaca =0;
  ttlsupply_noaca + DAYS_SUPLY_NUM;
  if last.bene_id;
run;

data d8;
set d7;
if ttlsupply_noaca > 60 then supply_noaca=1; else supply_noaca=0;
keep bene_id supply_noaca ttlsupply_noaca;
run;

*91 days;
data d9;
set d1;
by bene_id;
if first.bene_id then output;
run;

data d10;
set d9;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration1_noaca;
duration1_noaca=1;
run;

*exclusion of NOAC-B Medication;

data mednoacb1; set _UPLDS.med_noacb; ndc1=ndc*1; drop ndc; run;
proc sort data=mednoacb1; by ndc1; run;

data dnoacb;
merge aa (in=in1) mednoacb1 (in=in2);
by ndc1;
if in1 and in2;
keep bene_id med_noacb;
med_noacb=1;
run;

proc sort data=dnoacb out=dnoacb1 nodupkey; by bene_id; run;

data finalcohortw;
set star2017unr_finalcohortz_p1;
run;

proc sort data=finalcohortw; by bene_id; run;
proc sort data=dnoacb1; by bene_id; run;
proc sort data=d10; by bene_id; run;
proc sort data=d8; by bene_id; run;
proc sort data=d6; by bene_id; run;
proc sort data=d2a; by bene_id; run;
proc sort data=da; by bene_id; run;

* record inclusion outcomes;

```

```

data aaa;
merge finalcohortw (in=in1) da d2a d6 d8 d10 dnoacb1;
by bene_id;
if in1;
drop SRVC_DT DAYS_SUPLY_NUM;
run;

data finalcohortw1;
set aaa;
if med_noaca=. then med_noaca=0;
if number_med_noaca=. then number_med_noaca=0;
if duration_noaca=. then duration_noaca=0;
if duration1_noaca=. then duration1_noaca=0;
if supply_noaca=. then supply_noaca=0;
if med_noacb=. then med_noacb=0;
if age_BOY>=18 and med_noaca=1 and number_med_noaca=1 and duration_noaca=1
and duration1_noaca=1 and supply_noaca=1 and med_noacb=0
then include_noaca=1; else include_noaca=0;
run;

/*Calculate PDC (Measure Assessment)*/

data medsnoaca_cx; set _UPLDS.noaca2017_vxx; ndc1=ndc*1; drop ndc; run;

data medsnoaca_cx1;
set medsnoaca_cx;
if Dabigatran_Flag ='X' then label1=1; else label1=.;
if Apixaban_Flag ='X' then label2=2; else label2=.;
if Edoxaban_Flag ='X' then label3=3; else label3=.;
if Rivaroxaban_Flag ='X' then label4=4; else label4=.;

keep ndc1 label1-label4;
run;

proc sort data=medsnoaca_cx1; by ndc1; run;

proc transpose data=medsnoaca_cx1 out=medsnoaca_cx2;
by ndc1;
run;

data medsnoaca_cx3;
set medsnoaca_cx2;
labels=coll1;
drop _name_coll1;
if labels ne .;
run;

proc sql;
create table want as
select aa.*, medsnoaca_cx3.labels
from aa, medsnoaca_cx3
where aa.ndc1 = medsnoaca_cx3.ndc1;
quit;

data want1;
set want;

```

```

keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM;
run;

proc sort data=want1; by bene_id labels SRVC_DT ; run;

data want2;
do until (last.labels);
set want1 ;
by bene_id labels;
SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;
output ;
end;
format SRVC_END DATE9. ;
run;

data want3;
set want2;
by bene_id labels ;
format adjStart adjEnd lastdate mmddyy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd = SRVC_DT + DAYS_SUPLY_NUM -1;
if first.labels then lastdate = .;
else do ;
** If overlap then adjust start/end dates, else keep as is;
if adjStart <= lastdate then do;
adjStart = lastdate + 1;
adjEnd = adjStart + DAYS_SUPLY_NUM -1 ;
end;
end;
** Save last ending date;
lastdate = adjEnd;
run;

proc sort data=want3 out=want4 nodupkey; by bene_id adjStart descending
DAYS_SUPLY_NUM;
run;

proc sort data=want4 out=d_noaca nodupkey;
by bene_id adjStart;
run;

data aa_d1; set d_noaca;
member_id=bene_id;
fill_dt=adjStart;
format fill_dt DDMYY10. ;
days_supply=DAYS_SUPLY_NUM;
run;

data aa_d2; set aa_d1;
keep member_id fill_dt days_supply; run;

data claims_med_noaca; set aa_d2;
run;

/*Step 1*/

```

```

proc sort data=claims_med_noaca; by member_id fill_dt;
run;

proc transpose data=claims_med_noaca out=fill_dates (drop=_name_)
prefix=fill_dt;
by member_id;
var fill_dt;
run;

proc transpose data=claims_med_noaca out=days_supply (drop=_name_)
prefix=days_supply;
by member_id;
var days_supply;
run;

data both;
merge fill_dates days_supply;
by member_id;
format start_dt end_dt mmddyy10. ;
start_dt=fill_dt1;
end_dt='31DEC2017'd;
run;

/*Step 2*/
data pdc_med_noaca;
set both;
array daydummy(365) day1-day365;
serv_days=end_dt-start_dt+1;
array filldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
    do i=1 to dim(filldates) while (filldates(i) ne .);
        if filldates(i)<=first_dt+ii-1<=filldates(i)+days_supply(i)-1
            then daydummy(ii)=1;
    end;
end;
drop i ii ;
dayscovered=sum(of daydummy[*]);
label dayscovered='Total Days Covered';
p_dayscovered=dayscovered/serv_days;
label p_dayscovered='Proportion of Days Covered';
drop fill_dt1-fill_dt365 days_supply1-days_supply365;
run;

data pdc_med_noaca_1;
set pdc_med_noaca;
bene_id=member_id;
pdc_med_noaca = p_dayscovered;
sd_med_noaca = serv_days;
dc_med_noaca = dayscovered;
keep bene_id pdc_med_noaca sd_med_noaca dc_med_noaca;
run;

```



```

proc sort data=pdc_med_noaca_1; by bene_id; run;
proc sort data= finalcohortw1; by bene_id; run;

data finalcohortw2;
merge finalcohortw1 (in=in1) pdc_med_noaca_1;
by bene_id;
if in1;
run;

/*****
*PDC for Long-Acting Inhaled Bronchodilator (PDC-COPD)*;
*****/

/*Inclusion criteria
1) A diagnosis for COPD
2) At least 2 prescription claims for any long-acting inhaled bronchodilator
(COPD-A)
3) Exclusion of Nebulized Bronchodilator Users;
4) First prescription occurred at least 91 days before the end of the year */

/* Determine inclusion outcomes*/

data medcopd1; set _UPLDS.med_copd; ndc1=ndc*1; drop ndc; run;
proc sort data=medcopd1; by ndc1; run;

data medcopd2;
set medcopd1;
if Nebulized_Exclusion = 'X' then label=1; else label=0;
run;

data medcopda;
set medcopd2;
if label=0;
run;

proc sort data=medcopda; by ndc1; run;

data d;
merge aa (in=in1) medcopda (in=in2);
by ndc1;
if in1 and in2;
keep bene_id SRVC_DT med_copda DAYS_SUPPLY_NUM;
med_copda=1;
run;

* Two medications on different dates;
proc sort data=d out=d ; by bene_id ;
run;

proc sort nodupkey data=d out=da; by bene_id; run;

proc sort data=d out=d1 nodupkey; by bene_id SRVC_DT ;
run;

proc freq data=d1 noprint;

```

```

table bene_id/out=d2; run;

data d2a;
set d2;
if count>=2;
count_med_copda = count;
keep bene_id number_med_copda count_med_copda;
number_med_copda=1;
run;

*91 days;
data d9;
set d1;
by bene_id;
if first.bene_id then output;
run;

data d10;
set d9;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_copd;
duration_copd=1;
run;

*exclude nebulized bronchodilator;

data medcopdb;
set medcopd2;
if label=1;
run;

proc sort data=medcopdb; by ndc1; run;

data dcopdb;
merge aa (in=in1) medcopdb (in=in2);
by ndc1;
if in1 and in2;
keep bene_id med_copdb;
med_copdb=1;
run;

proc sort data=dcopdb out=dcopdb1 nodupkey; by bene_id; run;

*include COPD diagnosis;

data dem; set star2017unr_finalcohortz_p1; keep bene_id; run;
data dem1; set star.DIAG_2017; bene_id=hicno; drop hicno; run;
proc sort data=dem out=dems; by bene_id; run;
proc sort data=dem1 out=dem1s; by bene_id; run;

data dem2;
merge dems (in=in1) dem1s (in=in2);
by bene_id;
if in1;
run;

```

```

data dem3;
set dem2;
dxcode=upcase(diag);
dx_3=substr(dxcode,1,3);
dx_4=substr(dxcode,1,4);
dx_5=substr(dxcode,1,5);

if dx_4='J410' | dx_4='J411' | dx_4='J418' | dx_3='J42' |
dx_4='J430' | dx_4='J431' | dx_4='J432' | dx_4='J438' | dx_4='J439' |
dx_4='J440' | dx_4='J441' | dx_4='J449' |

dx_4='4910' | dx_4='4911' | dx_4='4918' | dx_4='4919' |
dx_4='4928' |
dx_5='49122' | dx_5='49321' | dx_5='49121' | dx_5='49322' |
dx_5='49120' | dx_5='49320'
then diagnosis_copd=1; else diagnosis_copd=0;
run;

data dem4; set dem3; keep bene_id diagnosis_copd; if diagnosis_copd=1; run;
proc sort data=dem4 nodupkey out=dem4s; by bene_id; run;

proc sort data=dem4s nodupkey; by bene_id; run;
proc sort data=dcopdb1 nodupkey; by bene_id; run;
proc sort data=d2a nodupkey; by bene_id; run;
proc sort data=d10 nodupkey; by bene_id; run;

data dacopda; set da; drop SRVC_DT DAYS_SUPLY_NUM; run;
proc sort data=dacopda nodupkey; by bene_id; run;
proc sort data=finalcohortw2 ; by bene_id; run;

data aaa;
merge finalcohortw2 (in=in1) dacopda dem4s d2a d10 dcopdb1;
by bene_id;
if in1;
run;

data finalcohortw3;
set aaa;
if med_copda=. then med_copda=0;
if number_med_copda=. then number_med_copda=0;
if duration_copd=. then duration_copd=0;
if med_copdb=. then med_copdb=0;
if diagnosis_copd=. then diagnosis_copd=0;
if age_BOY>=18 and diagnosis_copd=1 and med_copda=1 and number_med_copda=1
and duration_copd=1 and med_copdb=0 then include_copd=1;
else include_copd=0;
run;

/*Calculate PDC (Measure Assessment)*/

data medcopda_cx; set _UPLDS.copda2017_vxx; ndc1=ndc*1; drop ndc; run;

data medcopda_cx1;
set medcopda_cx;
if Aclidinium_Flag ='X' then label1=1; else label1=.;
if Formoterol_Flag ='X' then label2=2; else label2=.;

```

```

if Glycopyrrolate_Flag ='X'      then label3=3; else label3=.;
if Indacaterol_Flag   ='X'      then label4=4; else label4=.;
if Olodaterol_Flag    ='X'      then label5=5; else label5=.;
if Salmeterol_Flag    ='X'      then label6=6; else label6=.;
if Tiotropium_Flag    ='X'      then label7=7; else label7=.;
if Umeclidinium_Flag  ='X'      then label8=8; else label8=.;
if Vilanterol_Flag    ='X'      then label9=9; else label9=.;

keep ndc1 label1-label9;
run;

proc sort data=medcopda_cx1; by ndc1; run;
proc transpose data=medcopda_cx1 out=medcopda_cx2;
  by ndc1;
run;

data medcopda_cx3;
set medcopda_cx2;
labels=coll;
drop _name_ coll;
if labels ne .;
run;

proc sql;
create table want as
select aa.*, medcopda_cx3.labels
from aa, medcopda_cx3
where aa.ndc1 = medcopda_cx3.ndc1;
quit;

data want1;
set want;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM;
run;

proc sort data=want1; by bene_id labels SRVC_DT ; run;

data want2;
do until (last.labels);
  set want1 ;
  by bene_id labels;
  SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;
  output ;
end;
format SRVC_END DATE9. ;
run;

data want3;
set want2;
by bene_id labels ;
format adjStart adjEnd lastdate mmdyy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd   = SRVC_DT + DAYS_SUPLY_NUM -1;
  if first.labels then lastdate = .;
  else do ;
** If overlap then adjust start/end dates, else keep as is;

```

```

        if adjStart <= lastdate then do;
            adjStart = lastdate + 1;
            adjEnd   = adjStart + DAYS_SUPLY_NUM -1 ;
        end;
    end;
** Save last ending date;
lastdate = adjEnd;
run;

proc sort data=want3 out=want4 nodupkey; by bene_id adjStart descending
DAYS_SUPLY_NUM;
run;

proc sort data=want4 out=d_copda nodupkey;
by bene_id adjStart;
run;

data aa_d1; set d_copda;
member_id=bene_id;
fill_dt=adjStart;
format fill_dt DDMYY10. ;
days_supply=DAYS_SUPLY_NUM;
run;

data aa_d2; set aa_d1;
keep member_id fill_dt days_supply; run;

data claims_med_copd; set aa_d2;
run;

/*Step 1*/

proc sort data=claims_med_copd; by member_id fill_dt;
run;

proc transpose data=claims_med_copd out=fill_dates (drop=_name_)
prefix=fill_dt;
by member_id;
var fill_dt;
run;

proc transpose data=claims_med_copd out=days_supply (drop=_name_)
prefix=days_supply;
by member_id;
var days_supply;
run;

data both;
merge fill_dates days_supply;
by member_id;
format start_dt end_dt mmdyy10. ;
start_dt=fill_dt1;
end_dt='31DEC2017'd;
run;

```

```

/*Step 2*/
data pdc_med_copd;
set both;
array daydummy(365) day1-day365;
serv_days=end_dt-start_dt+1;
array filldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
    do i=1 to dim(filldates) while (filldates(i) ne .);
        if filldates(i)<=first_dt+ii-1<=filldates(i)+days_supply(i)-1
            then daydummy(ii)=1;
        end;
    end;
drop i ii ;
dayscovered=sum(of daydummy[*]);
label dayscovered='Total Days Covered';
p_dayscovered=dayscovered/serv_days;
label p_dayscovered='Proportion of Days Covered';
drop fill_dt1-fill_dt365 days_supply1-days_supply365;
run;

data pdc_med_copd_1;
set pdc_med_copd;
bene_id=member_id;
pdc_med_copd = p_dayscovered;
sd_med_copd = serv_days;
dc_med_copd = dayscovered;
keep bene_id pdc_med_copd sd_med_copd dc_med_copd;
run;

proc sort data=pdc_med_copd_1; by bene_id; run;
proc sort data= finalcohortw3; by bene_id; run;

data finalcohortw4;
merge finalcohortw3 (in=in1) pdc_med_copd_1;
by bene_id;
if in1;
run;

/*****
*PDC for Antiretroviral Medications (PDC-ARV)*;
*****/
/*Inclusion criteria
1) Antiretroviral Medication Users
2) 18 years and older
3) At least 2 unique medications on 2 unique dates
4) First prescription occurred at least 91 days before the end of the year */

/* Determine inclusion outcomes*/

data medsarv1; set _UPLDS.arv2017_vxx; ndc1=ndc*1; drop ndc; run;
proc sort data=medsarv1; by ndc1; run;

data medsarv1_ex1;

```

```

set medsarv1;
if Abacavir_Flag = 'X' then label1=1; else label1=.;
if Atazanavir_Flag = 'X' then label2=2; else label2=.;
if Darunavir_Flag = 'X' then label3=3; else label3=.;
if Delavirdine_Flag = 'X' then label4=4; else label4=.;
if Didanosine_Flag = 'X' then label5=5; else label5=.;
if Dolutegravir_Flag = 'X' then label6=6; else label6=.;
if Efavirenz_Flag = 'X' then label7=7; else label7=.;
if Elvitegravir_Flag = 'X' then label8=8; else label8=.;
if Emtricitabine_Flag = 'X' then label9=9; else label9=.;
if Enfuvirtide_Flag = 'X' then label10=10; else label10=.;
if Etravirine_Flag = 'X' then label11=11; else label11=.;
if Fosamprenavir_Flag = 'X' then label12=12; else label12=.;
if Indinavir_Flag = 'X' then label13=13; else label13=.;
if Lamivudine_Flag = 'X' then label14=14; else label14=.;
if Lopinavir_Flag = 'X' then label15=15; else label15=.;
if Maraviroc_Flag = 'X' then label16=16; else label16=.;
if Nelfinavir_Flag = 'X' then label17=17; else label17=.;
if Nevirapine_Flag = 'X' then label18=18; else label18=.;
if Raltegravir_Flag = 'X' then label19=19; else label19=.;
if Rilpivirine_Flag = 'X' then label20=20; else label20=.;
if Ritonavir_Flag = 'X' then label21=21; else label21=.;
if Saquinavir_Flag = 'X' then label22=22; else label22=.;
if Stavudine_Flag = 'X' then label23=23; else label23=.;
if Tenofovir_Flag = 'X' then label24=24; else label24=.;
if Tipranavir_Flag = 'X' then label25=25; else label25=.;
if Zidovudine_Flag = 'X' then label26=26; else label26=.;

keep ndc1 label1-label26;
run;

proc sort data=medsarv1_ex1; by ndc1; run;

proc transpose data=medsarv1_ex1 out=medsarv1_ex2;
  by ndc1;
run;

data medsarv1_ex3;
set medsarv1_ex2;
labels=col1;
drop _name_ col1;
if labels ne .;
run;

proc sql;
create table wantz as
select aa.*, medsarv1_ex3.labels
from aa, medsarv1_ex3
where aa.ndc1 = medsarv1_ex3.ndc1;
quit;

data wantz1;
set wantz;
med_arv=1;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM med_arv;
run;

```

```

proc sort data=wantz1 out=wantz2 nodupkey; by bene_id labels SRVC_DT
descending DAYS_SUPLY_NUM;

proc sort data=wantz2 out=d nodupkey; by bene_id labels SRVC_DT;

*record ARV medication use;
proc sort nodupkey data=wantz1 out=da; by bene_id; run;
data das; set da; keep bene_id med_arv; run;

*91 days;
proc sort data=wantz1 out=d1 nodupkey; by bene_id SRVC_DT ;
run;

data d3;
set d1;
by bene_id;
if first.bene_id then output;
run;

data d4;
set d3;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_med_arv;
duration_med_arv=1;
run;

* at least 2 unique medications on 2 unique dates;

proc freq data=d noprint;
table bene_id*labels/out=m1; run;

data m2;
set m1;
if count>=2;
count_med_arv=count;
keep bene_id count_med_arv labels;
run;

proc freq data=m2 noprint;
table bene_id/out=m3; run;

data m4;
set m3;
if count>=2;
keep bene_id;
run;

proc sort data= m4; by bene_id;run;
proc sort data= m2; by bene_id descending count_med_arv;run;

data m5;
merge m2(in= in1) m4 (in= in2);
by bene_id;
if in1 and in2;
number_med_arv=1;
run;

```



```

proc sort data=m5 out=m6 nodupkey; by bene_id ;
run;

data m7;
set m6;
drop labels count_med_arv;
run;

proc sort data= d4 nodupkey; by bene_id; run;
proc sort data= m7 nodupkey; by bene_id; run;
proc sort data= das nodupkey; by bene_id; run;
proc sort data= finalcohortw4; by bene_id; run;

* record inclusion outcomes;
data aaa;
merge finalcohortw4 (in=in1) das m7 d4 ;
by bene_id;
if in1;
run;

data finalcohortw5;
set aaa;
if med_arv=. then med_arv=0;
if number_med_arv=. then number_med_arv=0;
if duration_med_arv=. then duration_med_arv=0;
if age_BOY>=18 and med_arv=1 and number_med_arv=1 and duration_med_arv=1 then
include_med_arv=1; else include_med_arv=0;
run;

/*Calculate PDC (Measure Assessment)*/

data medsarv_cx; set _UPLDS.arv2017_vxx; ndc1=ndc*1; drop ndc; run;

data medsarv_cx1;
set medsarv_cx;
if Abacavir_Flag ='X' then label1=1; else label1=.;
if Atazanavir_Flag ='X' then label2=2; else label2=.;
if Darunavir_Flag ='X' then label3=3; else label3=.;
if Delavirdine_Flag ='X' then label4=4; else label4=.;
if Didanosine_Flag ='X' then label5=5; else label5=.;
if Dolutegravir_Flag ='X' then label6=6; else label6=.;
if Efavirenz_Flag ='X' then label7=7; else label7=.;
if Elvitegravir_Flag ='X' then label8=8; else label8=.;
if Emtricitabine_Flag ='X' then label9=9; else label9=.;
if Enfuvirtide_Flag ='X' then label10=10; else label10=.;
if Etravirine_Flag ='X' then label11=11; else label11=.;
if Fosamprenavir_Flag ='X' then label12=12; else label12=.;
if Indinavir_Flag ='X' then label13=13; else label13=.;
if Lamivudine_Flag ='X' then label14=14; else label14=.;
if Lopinavir_Flag ='X' then label15=15; else label15=.;
if Maraviroc_Flag ='X' then label16=16; else label16=.;
if Nelfinavir_Flag ='X' then label17=17; else label17=.;
if Nevirapine_Flag ='X' then label18=18; else label18=.;
if Raltegravir_Flag ='X' then label19=19; else label19=.;
if Rilpivirine_Flag ='X' then label20=20; else label20=.;
if Ritonavir_Flag ='X' then label21=21; else label21=.;
if Saquinavir_Flag ='X' then label22=22; else label22=.;

```

```

if Stavudine_Flag = 'X' then label23=23; else label23=.;
if Tenofovir_Flag = 'X' then label24=24; else label24=.;
if Tipranavir_Flag = 'X' then label25=25; else label25=.;
if Zidovudine_Flag = 'X' then label26=26; else label26=.;
keep ndc1 label1-label26;
run;

proc sort data=medsarv_cx1; by ndc1; run;

proc transpose data=medsarv_cx1 out=medsarv_cx2;
  by ndc1;
run;

data medsarv_cx3;
set medsarv_cx2;
category=coll;
drop _name_ coll;
if category ne .;
run;

proc sql;
create table wantq as
select aa.*, medsarv_cx3.category
from aa, medsarv_cx3
where aa.ndc1 = medsarv_cx3.ndc1;
quit;

data wantq1;
set wantq;
keep bene_id ndc1 category SRVC_DT DAYS_SUPLY_NUM;
run;

proc sort data=wantq1; by bene_id category SRVC_DT ; run;

data wantq2;
do until (last.category);
  set wantq1 ;
  by bene_id category;
  SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;
  output ;
end;
format SRVC_END DATE9. ;
run;

data wantq3;
set wantq2;
by bene_id category ;
format adjStart adjEnd lastdate mmdyy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd   = SRVC_DT + DAYS_SUPLY_NUM -1;
  if first.category then lastdate = .;
  else do ;
** If overlap then adjust start/end dates, else keep as is;
  if adjStart <= lastdate then do;
    adjStart = lastdate + 1;

```

```

        adjEnd    = adjStart + DAYS_SUPLY_NUM -1 ;
        end;
    end;
** Save last ending date;
lastdate = adjEnd;
run;

*rename Variables;
data wantq4;
set wantq3;
keep bene_id category adjStart DAYS_SUPLY_NUM;
run;

data wantq5;
set wantq4;
SRVC_DT = adjStart;
format SRVC_DT mmddyy10.;
keep bene_id category SRVC_DT DAYS_SUPLY_NUM;
run;

data mcon2;
set wantq5;
format SRVC_DT END_DT END_DT1 mmddyy10.;
END_DT = SRVC_DT+DAYS_SUPLY_NUM-1;
END_DT1='31DEC2017'd;
run;

data mcon2x;
set mcon2;
if END_DT >= END_DT1 then END_DT2=END_DT1; else END_DT2= END_DT;
format END_DT2 mmddyy10.;
if SRVC_DT >= END_DT1 then delete;
keep bene_id category srvc_dt end_dt2 END_DT1;
run;

data mcon3;
set mcon2x;
keep bene_id category srvc_dt end_dt2;
run;

proc sort data=mcon3; by bene_id category srvc_dt ;

*record treatment period;
data arv_treatp;
set mcon2x ;
arvtreat_p = END_DT1 - srvc_dt +1;
run;

proc sort data=arv_treatp out=arv_treatp1; by bene_id descending arvtreat_p;
run;

data arv_treatp2;
set arv_treatp1;
by bene_id;
if first.bene_id then output;
keep bene_id arvtreat_p;
run;

```

```

data have;
set
  mcon3(keep = bene_id category srvc_dt in=insrvc_dt rename=(srvc_dt=date))
  mcon3(keep = bene_id category end_dt2 in=inend_dt2 rename=(end_dt2=date))
;
if insrvc_dt then marker = 1;
           else marker = -1;
run;

proc sort data = have;
  by bene_id category Date;
run;

*define accumulator;
data havel;
  set have;
  by bene_id category ;
  if first.category then cumulator = 0;
  cumulator + marker;
run;

*remove overlapping days;
data have2;
  set havel;
  by bene_id category ;

  if first.category or lag(cumulator) = 0 then
  do;
    retain srvc_dt end_dt2;
    format srvc_dt mmddyy10. end_dt2 mmddyy10.;
    srvc_dt = Date;
    end_dt2 = Date;
  end;

  srvc_dt = srvc_dt >< Date;
  end_dt2 = end_dt2 <> Date;

  if cumulator = 0 then output;
  keep bene_id category srvc_dt end_dt2;
run;

data want;
  set have2;
  do PrescribedDate= srvc_dt to end_dt2;
    output;
  end;
  format PrescribedDate mmddyy10.;
run;

proc summary data=want nway;
  class bene_id PrescribedDate;
  var srvc_dt;
  output out=summary (drop=_: ) n=category;
run;

```

```

data tab;
  set summary;
  overlap=(category>1);
run;

proc tabulate data=tab out=tab1;
  class bene_id;
  var overlap;
  tables bene_id, overlap*(sum);
run;

/*Calculate PDC (Measure Assessment)*/

proc sort data=tab1; by bene_id; run;
proc sort data=arv_treatp2; by bene_id; run;

data tab2;
merge tab1 arv_treatp2;
by bene_id;
run;

data pdc_med_arv_1;
set tab2;
p_dayscovered=overlap_Sum/arvtreat_p;
pdc_med_arv = p_dayscovered;
sd_med_arv = arvtreat_p;
dc_med_arv = overlap_Sum;
keep bene_id pdc_med_arv sd_med_arv dc_med_arv;
run;

proc sort data=pdc_med_arv_1; by bene_id; run;
proc sort data= finalcohortw5; by bene_id; run;

data finalcohortw6;
merge finalcohortw5 (in=in1) pdc_med_arv_1;
by bene_id;
if in1;
run;

/*****
*Adherence to Non-Infused Disease-Modifying Agents Used to treat Multiple
Sclerosis (PDC-MS)*;
*****/

/*Inclusion criteria
1) Disease Modifying Medication Users (MS-A)
2) 18 years and older
3) At least 2 medications on 2 unique dates
4) First prescription occurred at least 91 days before the end of the year
5) At least 56 days of supply
6) Exclusion of infused disease-modifying medication (MS-B)*/

/* Determine inclusion outcomes*/

data medms1; set _UPLDS.med_ms; ndc1=ndc*1; drop ndc; run;
proc sort data=medms1; by ndc1; run;

```

```

data medms2;
set medms1;
rename 'MSPDC-B Flag'n = MSPDC_B_Flag;
run;

data medms3;
set medms2;
if MSPDC_B_Flag = 'X' then label=1; else label=0;
run;

data medmsa;
set medms3;
if label=0;
run;

proc sort data=medmsa; by ndc1; run;

data d;
merge aa (in=in1) medmsa (in=in2);
by ndc1;
if in1 and in2;
keep bene_id SRVC_DT med_msa DAYS_SUPPLY_NUM;
med_msa=1;
run;

* Two medications on different dates;
proc sort data=d out=d ; by bene_id ;
run;

proc sort nodupkey data=d out=da; by bene_id; run;

proc sort data=d out=d1 nodupkey; by bene_id SRVC_DT ;
run;

proc freq data=d1 noprint;
table bene_id/out=d2; run;

data d2a;
set d2;
if count>=2;
count_med_msa = count;
keep bene_id number_med_msa count_med_msa;
number_med_msa=1;
run;

*91 days;
data d3;
set d1;
by bene_id;
if first.bene_id then output;
run;

data d4;
set d3;
if '31DEC2017'd-srvc_dt>90;
keep bene_id duration_med_msa;
duration_med_msa=1;

```

```

run;

*Greater than 56 days of supply;
data d7 ;
  set d1;
  by bene_id;
  if first.bene_id then ttlsupply_msa =0;
  ttlsupply_msa + DAYS_SUPLY_NUM;
  if last.bene_id;
run;

data d8;
set d7;
if ttlsupply_msa > 56 then supply_msa=1; else supply_msa=0;
keep bene_id supply_msa ttlsupply_msa;
run;

*exclude Infused Disease-Modifying used to treat MS (MS-B);

data medmsb;
set medms3;
if label=1;
run;

proc sort data=medmsb; by ndc1; run;

data dmsb;
merge aa (in=in1) medmsb (in=in2);
by ndc1;
if in1 and in2;
keep bene_id med_msb;
med_msb=1;
run;

proc sort data=dmsb out=dmsb1 nodupkey; by bene_id; run;

proc sort data=d8 nodupkey; by bene_id; run;
proc sort data=d2a nodupkey; by bene_id; run;
proc sort data=d4 nodupkey; by bene_id; run;

data damsas; set da; drop SRVC_DT DAYS_SUPLY_NUM; run;
proc sort data=damsas nodupkey; by bene_id; run;
proc sort data= finalcohortw6 ; by bene_id; run;

data aaa;
merge finalcohortw6 (in=in1) damsas d2a d4 d8 dmsb1;
by bene_id;
if in1;
run;

data finalcohortw7;
set aaa;
if med_msa=. then med_msa=0;
if number_med_msa=. then number_med_msa=0;
if duration_med_msa=. then duration_med_msa=0;
if supply_msa=. then supply_msa=0;

```

```

if med_msb=. then med_msb=0;
if age_BOY>=18 and med_msa=1 and number_med_msa=1 and duration_med_msa=1 and
supply_msa=1 and med_msb=0 then include_msa=1;
else include_msa=0;
run;

/*Calculate PDC (Measure Assessment)*/

data medsms_cx; set _UPLDS.ms2017_vxx; ndc1=ndc*1; drop ndc; run;

data medsms_cx1;
set medsms_cx;
if Beta_interferon_Flag ='X' then label1=1; else label1=.;
if Glatiramer_Flag ='X' then label2=2; else label2=.;
if Fingolimid_Flag ='X' then label3=3; else label3=.;
if Teriflunomide_Flag ='X' then label4=4; else label4=.;
if Dimethyl_fumerate_Flag ='X' then label5=5; else label5=.;
if Peginterferon_beta_la_Flag ='X' then label6=6; else label6=.;
if Daclizumab_Flag ='X' then label7=7; else label7=.;

keep ndc1 label1-label7;
run;

proc sort data=medsms_cx1; by ndc1; run;

proc transpose data=medsms_cx1 out=medsms_cx2;
by ndc1;
run;

data medsms_cx3;
set medsms_cx2;
labels=coll;
drop _name_ coll;
if labels ne .;
run;

proc sql;
create table want as
select aa.*, medsms_cx3.labels
from aa, medsms_cx3
where aa.ndc1 = medsms_cx3.ndc1;
quit;

data want1;
set want;
keep bene_id ndc1 labels SRVC_DT DAYS_SUPLY_NUM;
run;

proc sort data=want1; by bene_id labels SRVC_DT ; run;

data want2;
do until (last.labels);
set want1 ;
by bene_id labels;
SRVC_END=SRVC_DT+DAYS_SUPLY_NUM;
output ;

```



```

    end;
    format SRVC_END DATE9. ;
run;

data want3;
set want2;
by bene_id labels ;
format adjStart adjEnd lastdate mmddyy10.;
retain lastdate;
adjStart = SRVC_DT;
adjEnd   = SRVC_DT + DAYS_SUPLY_NUM -1;
    if first.labels then lastdate = .;
    else do ;
** If overlap then adjust start/end dates, else keep as is;
    if adjStart <= lastdate then do;
        adjStart = lastdate + 1;
        adjEnd   = adjStart + DAYS_SUPLY_NUM -1 ;
    end;
    end;
** Save last ending date;
lastdate = adjEnd;
run;

proc sort data=want3 out=want4 nodupkey; by bene_id adjStart descending
DAYS_SUPLY_NUM;
run;

proc sort data=want4 out=d_ms nodupkey;
by bene_id adjStart;
run;

data aa_d1; set d_ms;
member_id=bene_id;
fill_dt=adjStart;
format fill_dt DDMMYY10. ;
days_supply=DAYS_SUPLY_NUM;
run;

data aa_d2; set aa_d1;
keep member_id fill_dt days_supply; run;

data claims_med_msa; set aa_d2;
run;

/*Step 1*/

proc sort data=claims_med_msa; by member_id fill_dt;
run;

proc transpose data=claims_med_msa out=fill_dates (drop=_name_)
prefix=fill_dt;
by member_id;
var fill_dt;
run;

proc transpose data=claims_med_msa out=days_supply (drop=_name_)
prefix=days_supply;

```

```

by member_id;
var days_supply;
run;

data both;
merge fill_dates days_supply;
by member_id;
format start_dt end_dt mmddyy10. ;
start_dt=fill_dt1;
end_dt='31DEC2017'd;
run;

/*Step 2*/
data pdc_med_msa;
set both;
array daydummy(365) day1-day365;
serv_days=end_dt-start_dt+1;
array filldates(*) fill_dt1-fill_dt365;
array days_supply(*) days_supply1-days_supply365;
do ii=1 to 365; daydummy(ii)=0; end;
format first_dt mmddyy10.;
first_dt='01JAN2017'd;
do ii=1 to 365;
    do i=1 to dim(filldates) while (filldates(i) ne .);
        if filldates(i)<=first_dt+ii-1<=filldates(i)+days_supply(i)-1
            then daydummy(ii)=1;
    end;
end;
drop i ii ;
dayscovered=sum(of daydummy[*]);
label dayscovered='Total Days Covered';
p_dayscovered=dayscovered/serv_days;
label p_dayscovered='Proportion of Days Covered';
drop fill_dt1-fill_dt365 days_supply1-days_supply365;
run;

data pdc_med_msa_1;
set pdc_med_msa;
bene_id=member_id;
pdc_med_msa = p_dayscovered;
sd_med_msa = serv_days;
dc_med_msa = dayscovered;
keep bene_id pdc_med_msa sd_med_msa dc_med_msa;
run;

proc sort data=pdc_med_msa_1; by bene_id; run;
proc sort data= finalcohortw7; by bene_id; run;

data finalcohortw8;
merge finalcohortw7 (in=in1) pdc_med_msa_1;
by bene_id;
if in1;
run;

```

```

/*****/
*Cholesterol Management in Coronary Artery Disease (CMC)*;
/*****/

/*Inclusion criteria
1) Patients with a diagnosis of coronary disease
2) 18 to 85 years old*/

/* Measure Assessment
1) At least one prescription of HMG-CoA reductase inhibitor */

/*Determine inclusion outcomes*/

data dem; set star2017unr_finalcohortz_p1; keep bene_id; run;
data dem1; set star.DIAG_2017; bene_id=hicno; drop hicno; run;
proc sort data=dem out=dems; by bene_id; run;
proc sort data=dem1 out=dem1s; by bene_id; run;

data dem2;
merge dems (in=in1) dem1s (in=in2);
by bene_id;
if in1;
run;

data dem3;
set dem2;
dxcode=upcase(diag);
dx_3=substr(dxcode,1,3);
dx_4=substr(dxcode,1,4);
dx_5=substr(dxcode,1,5);
dx_6=substr(dxcode,1,6);

if
dx_4='I214' | dx_4='I200' | dx_4='I201' | dx_4='I208' |
dx_4='I209' | dx_5='I2109' | dx_5='I2111' | dx_5='I2119' | dx_5='I2129' |
dx_4='I213' | dx_4='I240' | dx_4='I241' | dx_4='I248' | dx_5='I2510' | dx_4='I252' |
dx_4='I255' | dx_5='I2581' | dx_5='I2589' | dx_4='I259' | dx_4='Z951'
| dx_5='Z9861' |
dx_5='I2101' | dx_5='I2102' | dx_5='I2121' | dx_4='I214' | dx_4='I220'
| dx_4='I221' |
dx_4='I222' | dx_4='I228' | dx_4='I229' | dx_4='I230' | dx_4='I231' | dx_4='I232' |
dx_4='I233' | dx_4='I234' | dx_4='I235' | dx_4='I236' | dx_4='I237' | dx_4='I238' |
dx_4='I249' |
dx_6='I25110' | dx_6='I25111' | dx_6='I25118' | dx_6='I25119' | dx_6='I25700'
| dx_6='I25701' |
dx_6='I25708' | dx_6='I25709' | dx_6='I25710' | dx_6='I25711' | dx_6='I25718'
| dx_6='I25719' |
dx_6='I25720' | dx_6='I25721' | dx_6='I25728' | dx_6='I25729' | dx_6='I25730'
| dx_6='I25731' |
dx_6='I25738' | dx_6='I25739' | dx_6='I25750' | dx_6='I25751' | dx_6='I25758'
| dx_6='I25759' |
dx_6='I25760' | dx_6='I25761' | dx_6='I25768' | dx_6='I25769' | dx_6='I25790'
| dx_6='I25791' |
dx_6='I25798' | dx_6='I25799' |

```

```

dx_5='41071' | dx_4='4111' | dx_4='4131' | dx_4='4139' |
dx_5='41001' |dx_5='41011' |dx_5='41031' |dx_5='41021' |dx_5='41041' |
dx_5='41051'|dx_5='41061'|dx_5='41081'|
dx_5='41091' |dx_5='41181' |dx_4='4110'|dx_5='41189' |
dx_5='41401' |dx_4='4292' |dx_3='412'|
dx_4='4148' |dx_4='4149' |dx_5='V4581' |dx_5='V4582'|
dx_5='42979' |dx_5='42971' |
dx_4='4295' |dx_4='4296'|
dx_5='41405' |
dx_5='41402'|
dx_5='41404' |dx_5='41403' |
dx_5='41406'|
dx_5='41407'

then diagnosis_cmc=1; else diagnosis_cmc=0;
run;

data dem4; set dem3; keep bene_id diagnosis_cmc; if diagnosis_cmc=1; run;
proc sort data=dem4 nodupkey out=dem4s; by bene_id; run;

/*Determine Measure Assessment*/

data medcmcstatins1; set _UPLDS.med_cmcstatins; ndc1=ndc*1; drop ndc; run;
proc sort data=medcmcstatins1; by ndc1; run;

data dcmcstatins;
merge aa (in=in1) medcmcstatins1 (in=in2);
by ndc1;
if in1 and in2;
keep bene_id med_cmcstatins;
med_cmcstatins=1;
run;

proc sort data=dcmcstatins out=dcmcstatins1 nodupkey; by bene_id; run;

proc sort data= finalcohortw8; by bene_id; run;
proc sort data=dcmcstatins1; by bene_id; run;
proc sort data=dem4s; by bene_id; run;

data aaa;
merge finalcohortw8 (in=in1) dem4s dcmcstatins1;
by bene_id;
if in1;
run;

*record outcomes for inclusion and measure assessment;

data finalcohortw9;
set aaa;
if diagnosis_cmc=. then diagnosis_cmc=0;
if med_cmcstatins=. then med_cmcstatins=0;
if diagnosis_cmc=1 and age<86 and age>17 then include_cmc=1; else
include_cmc=0;
run;

```

```

/*****
*Use of Benzodiazepine Sedative Hypnotics in the Elderly (BSH)*;
*****/

/*Inclusion criteria
1) 65 years of age and older*/

/* Measure Assessment
1) At least two prescriptions for any benzodiazepine sedative hypnotic for 90
cumulative days */

/*Determine inclusion outcomes*/

data medbsh1; set _UPLDS.med_bsh; ndc1=ndc*1; drop ndc; run;
proc sort data=medbsh1; by ndc1; run;

data d;
merge aa (in=in1) medbsh1 (in=in2);
by ndc1;
if in1 and in2;
keep bene_id SRVC_DT med_bsh DAYS_SUPLY_NUM;
med_bsh=1;
run;

* two medications on different dates;
proc sort data=d out=d ; by bene_id ;
run;

proc sort nodupkey data=d out=da; by bene_id; run;

proc sort data=d out=d1 nodupkey; by bene_id SRVC_DT ;
run;

proc freq data=d1 noprint;
table bene_id/out=d2; run;

data d2a;
set d2;
if count>=2;
count_med_bsh = count;
keep bene_id number_med_bsh count_med_bsh;
number_med_bsh=1;
run;

*greater than 90 days of cumulative supply (measure assessment);

proc means data=d1 noprint;
class bene_id ;
var DAYS_SUPLY_NUM;
output out=d4 sum= ;
run;

data d4a;
set d4;
if DAYS_SUPLY_NUM>90;
if bene_id ne '';

```

```

ttlsupply_bsh = DAYS_SUPLY_NUM;
keep bene_id ttlsupply_bsh supply_bsh;
supply_bsh = 1;
run;

proc sort data=d4a out=d4as nodupkey; by bene_id; run;
proc sort data=d2a out=d2as nodupkey; by bene_id; run;
proc sort data=da ; by bene_id; run;
proc sort data= finalcohortw9 ; by bene_id; run;

data aaa;
merge finalcohortw9 (in=in1) da d2as d4as;
by bene_id;
if in1;
run;

data finalcohortw10;
set aaa;
if med_bsh=. then med_bsh=0;
if number_med_bsh=. then number_med_bsh=0;
if supply_bsh=. then supply_bsh=0;
if age_BOY>=65 then include_bsh=1; else include_bsh=0;
if med_bsh=1 and number_med_bsh=1 and supply_bsh=1 then use_bsh=1; else
use_bsh=0;
drop DAYS_SUPLY_NUM SRVC_DT;
run;

data star2017unr_finalcohortw_p1;
set finalcohortw10;
run;

```