

```

proc printto;run;

%pgm_begin_end(b)

options validvarname=V7;

*proc import datafile='\\pixley\HOA\TonyY\readmission\z_HEDIS 2015
Volume 2 VSD - 2014-07-01.xlsx' dbms=xlsx out=vs2015 replace;
* sheet = 'Volume 2 Value Sets to Codes';
* This data needs to be updated for the run in Oct 2016 ;
* File location \\plano\hoa\HEDIS\HEDIS 2017\HEDIS Tech Specs
2017\HEDIS 2017 Tech Specs\NCQA;
proc import datafile='\\pixley\HOA\TonyY\readmission\HEDIS value
sets\z_HEDIS 2017 Volume 2 VSD 2016-07-01.xlsx' dbms=xlsx out=vs2017
replace;
    sheet = 'Volume 2 Value Sets to Codes';
run;

data fmt;
    set vs2017;
    length fmtname $16.;
    if value_set_name = 'Chemotherapy' then fmtname = '$chemo';
    if value_set_name = 'Rehabilitation' then fmtname = '$rehab';
    if find(value_set_name, 'Transplant') then do;
        if code_system in ('ICD9CM','ICD10CM') then fmtname =
'$transplant_dx';
        else if code_system in ('ICD9PCS','ICD10PCS') then fmtname =
'$transplant_px';
        else fmtname = '$transplant_ot';
    end;
    if value_set_name = 'Potentially Planned Procedures' then fmtname =
'$plan';
    if value_set_name = 'Acute Condition' then fmtname = '$acute';
    if value_set_name in ('Pregnancy', 'Perinatal Conditions') then
fmtname = '$pregnancy';
run;

proc sql;
    create table fmt2 as
    select code as start, '+' as label, fmtname
    from fmt where fmtname is not null order by 3,1;
quit;
proc format cntlin=fmt2;
run;

/*****
/* DEFINE TRANSPLANTS USING CPT/HCPCS/REV codes*/
*****/

*273 236;
data foo;

```

```

    set qnxt.qnxt_detail_inpsnf clm.clm_detail_inpsnf
    clm11.clm_detail_inpsnf enc.enc_detail_inpsnf enc11.enc_detail_inpsnf;
    where put(strip(proc_cd), $transplant_ot.) = '+' or
    put(strip(rev_cd), $transplant_ot.) = '+';
run;

```

```

proc sql;
    create table bar as
    select distinct cl_id as start, '$clid' as fmtname, '+' as label
    from foo;
quit;

```

```

proc format cntlin=bar;
run;

```

```

data bar;
    set qnxt.qnxt_hdr_inpsnf clm.clm_hdr_inpsnf clm11.clm_hdr_inpsnf
    enc.enc_hdr_inpsnf enc11.enc_hdr_inpsnf;
    where put(strip(cl_id), $clid.) = '+';
run;

```

```

proc sql;
    create table baz as
    select distinct case_id as start, '$caseid' as fmtname, '+' as label
    from bar where case_id is not null;
quit;

```

```

proc format cntlin=baz;
run;

```

*Hospital admissions since 2012, define various flags;

```

data inp_case;
    set qnxt.qnxt_case_inpsnf(in=a)
        clm.clm_case_inpsnf(in=b) clm11.clm_case_inpsnf(in=bb)
        enc.enc_case_inpsnf(in=c) enc11.enc_case_inpsnf(in=cc);

    where srv_cat in: ('01','02','03') and dis_dt >= &beg_mon_dt. and
    adm_dt <= dis_dt;
    if a then tag = '1qnt';
    if b or bb then tag = '2clm';
    if c or cc then tag = '3enc';
    if (a or b or bb) and provider =: 'H' then hosp_id =
    substr(provider, 2) + 0;    *define hosp_id used for claims;

```

```

    keep CIN_NO MEMBER_NO CASE_ID ADM_DT DIS_DT case_dx: case_pr: tag
    severity aprdrg dis_status hosp_id PAID_AMT_CASE ADJ_BILL_AMT_CASE
    icd_flag;
run;

```

```

data inp_case;
    set inp_case;
    array dx(*) CASE_DX;;

```

```

array px(*) CASE PR;;
do i=1 to dim(px);
  if put(px(i), $transplant_px.) = '+' then transplant = 'Y';
  *first only;
  if put(px(i), $plan.) = '+' then plan = 'Y';
  *first only;
end;
do i=1 to dim(dx);
  if put(dx(i), $transplant_dx.) = '+' then transplant = 'Y';
  *first only;
end;

if put(strip(case_id), $caseid.) = '+' then transplant = 'Y';
*first only;
if put(case_dx1, $chemo.) = '+' then chemo = 'Y';
*first only;
if put(case_dx1, $rehab.) = '+' then rehab = 'Y';
*first only;
if put(case_dx1, $acute.) = '+' then acute = 'Y';
*first only;
if put(case_dx1, $pregnancy.) = '+' then pregnancy0 = 'Y'; *all;
if dis_status in ('40','41','42','20') then expired0 = 'Y'; *all;
drop dis_status i;
run;

proc sort data=inp_case;
  by cin_no adm_dt dis_dt tag;
run;

proc sort data=inp_case nodupkey;
  by cin_no adm_dt dis_dt;
run;

* Identify acute to acute transfers. If the discharge date is on the
same day or 1 day before the next admission date,
they are considered transfers. All transfers share the same ID
variable;
data tmp;
  set inp_case;
  by cin_no;
  retain first last;
  FORMAT first last MMDDYY10.;

  rownum = _N_;
  if first.cin_no then do;
    first = adm_dt;
    last = dis_dt;
    id = 1;
  end;
  else if adm_dt <= last+1 then
    last = max(last, dis_dt);
  else do;

```

```

        first = adm_dt;
        last = dis_dt;
        id + 1;
    end;
    drop adm_dt dis_dt;
run;

*For transfers, use the diagnosis codes from the first admission (per
Na's instruction). Use the last admission for the consolidated admit
and discharge dates;
*Acute to acute transfer is finished after this step;
proc sql;
    create table inp_case2 as
    select *, max(pregnancy0) as pregnancy, max(expired0) as expired,
            max(first) as adm_dt format mmddyy10., max(last) as
dis_dt format mmddyy10., (count(*) > 1) as transfer
    from tmp group by cin_no, id having rownum = min(rownum)
    order by rownum;

    select count(*), count(distinct cin_no), count(distinct member_no),
count(distinct case_id) from inp_case2;
quit;

*define PREV - gap between current admission date and last discharge
date;
data inp_case2;
    set inp_case2;
    by cin_no;
    prev = adm dt - lag(dis dt);
    if first.cin_no then prev = .;
    yearmth = put(dis_dt, yymmnn6.);
    if adm_dt = dis_dt then same_day = 'Y';
run;

*define NEXT - gap between current discharge date and next admission
date;
*define planned visits (step 5 in HEDIS PCR definition);
data inp_case2 (drop=pregnancy0 expired0 first last transplant plan
chemo rehab acute rownum plan_stay_next)
    plan (keep=cin_no id plan_stay_next);
    set inp_case2;
    by cin_no;
    if eof1=0 then set inp_case2(firstobs=2 keep=prev
rename=(prev=next)) end=eof1;
    if last.cin_no then next = .;
    output inp_case2;

    if not first.cin_no and (chemo = 'Y' or rehab = 'Y' or transplant =
'Y' or (plan = 'Y' and acute ~= 'Y')) and prev <= 30 then do;
        id = id - 1;
        plan_stay_next = 'Y';
        output plan;
    end;
end;
quit;

```

```

    end;
run;

proc import
datafile=/'\\pixley\HOA\TonyY\readmission\hosp_name_011516.xlsx'*/
'\\pixley\HOA\TonyY\readmission\hospital name mapping
file\hosp_name_011516.xlsx' dbms=xlsx out=hosp_name replace;
run;

proc sql;
    create table enc as
    select distinct a.case_id, a.dos_1, b.hospital_id, b.hospital_name
    from

(select * from enc.enc_hdr_inpsnf(keep=case_id dos_1 bill:) union
select * from enc11.enc_hdr_inpsnf(keep=case_id dos_1 bill:)) a

    left join hosp_name b
    on a.bill_prov = b.bill_prov and a.bill_prov_fname =
b.bill_prov_fname and a.bill_prov_npi = b.bill_prov_npi
    where a.case_id in (select case_id from inp_case2)
    order by 1;

    select count(*),count(distinct case_id) from enc;
quit;

proc sql;
    create table foo as
    select distinct case_id, hospital_id, hospital_name from enc
    group by case_id having count(*) = 1 or (count(*) =
count(hospital_id) and count(distinct hospital_id) = 1) or
count(distinct hospital_id) = 0;

    create table bar as
    select distinct case_id, hospital_id, hospital_name from enc
    where case_id not in (select case_id from foo)
    group by case_id having max(hospital_id) in
('KP', 'KP2', 'IPACLM', 'IPAENC') and hospital_id = min(hospital_id);

    create table baz as
    select * from enc
    where case_id not in (select case_id from foo union select case_id
from bar) and hospital_id is not null
    order by case_id, dos_1;

    select count(*),count(distinct case_id) from foo;
    select count(*),count(distinct case_id) from bar;
    select count(*),count(distinct case_id) from baz;
quit;

proc sort data=baz nodupkey;
    by case_id;

```

```

run;

data enc_hosp_name;
  set foo bar baz;
  by case_id;
  hosp_name_source = 'enc';
  drop dos_1;
run;

*Transpose the MEMMO file horizontally, to get a Yes/No flag per
yearmth;
proc sql;
  create table m as
  select distinct cin_no, yearmth, mem from out.memmo /*edit libname*/
  where cin_no in (select distinct cin_no from inp_case2) and bp is
not null;
quit;

proc transpose data=m out=m2(drop=_NAME_ _LABEL_) prefix=ym;
  by cin_no;
  var mem;
  id yearmth;
run;

data m2;
  retain CIN_NO &retain_string.;
  set m2;
run;

proc sql;
  create table inp_case3 as
  select a.*, b.*, c.hospital_name as hosp_name, d.*, f.dob, m.*,
m2.*, p.*
  from inp_case2 a
  left join enc_hosp_name b      on a.case_id = b.case_id
  left join mpd.lac_hospital c  on a.hosp_id = c.hospital_id
  left join plan d              on a.cin_no = d.cin_no and a.id
= d.id
  left join dmr.members f      on a.member_no = f.mhc_member_no
  left join out.memmo(keep=cin_no mhc_member_no yearmth product_code
bp mem /*spd*/ dual SEGMENT DHS site_no aid_code first_name last_name
gender where=(bp is not null)) m
      on m.cin_no = a.cin_no and m.mhc_member_no = a.member_no and
m.yearmth = a.yearmth
  left join m2                  on m.cin_no = m2.cin_no
  left join out.ppg(keep=site_no PPG_CODE PPG_name: ppg2:) p on
m.site_no = p.site_no /* Update May 31, 2016 */
  order by a.cin_no, a.id;

  /*select count(*), count(distinct cin_no), count(distinct
member_no), count(distinct case_id) from inp_case3;*/
quit;

```

```

proc sql;
    select count(*), count(distinct cin_no), count(distinct
member_no), count(distinct case_id) from inp_case3;
quit;

/*
Continuous enrollment:
MediCal (product_code 10) - no gaps between dis_dt-120 and dis_dt+30
SNP (product_code 70) - One 45-day gap (i.e. one month gap in yearmth
granularity) allowed between dis_dt-365 and dis_dt, no gaps between
dis_dt and dis_dt+30
Other LOBs (40, 60, 80, 90) added in Sept 2015

Additional exclusions:
1) [DELETED] Only keep stays with discharge date before December 1st
of each year [DELETED]
2) Exclude age under 21 in MediCal, under 18 in SNP
Finally, define denominator and numerator
*/

* breaking up foo into separate parts;

data foo;
    set inp_case3;

    hospital_id = coalescec(hospital_id, compress(hosp_id));
    hospital_name = coalescec(hospital_name, hosp_name);

    if hospital_id = '45' then hospital_name = 'COAST PLAZA HOSPITAL';
    if hospital_id = '84' then hospital_name = 'WESTERN MEDICAL CENTER
SANTA ANA';
    if hospital_id = '160' then hospital_name = 'PROVIDENCE TARZANA
MEDICAL CENTER';
    if hospital_id = '221' then hospital_name = 'BAKERSFIELD MEMORIAL
HOSPITAL';
    if hospital_id = '580' then hospital_name = 'PROVODENCE LITTLE
COMPANY OF MARY-SAN PEDRO';
    if hospital_id = '621' then hospital_name = 'PROMISE HOSP OF ELA-
SUBURBAN MEDICAL CENTER';
    if hospital_id = '625' then hospital_name = 'MONTCLAIR HOSPITAL
MEDICAL CENTER';
    if hospital_id = '969' then hospital_name = 'COASTAL COMMUNITIES
HOSPITAL';
    if hospital_id = '1044' then hospital_name = 'ST LUKES MEDICAL
CENTER';
    if hospital_id = '1405' then hospital_name = 'SHERMAN OAKS HOSP &
HLTH CTR';
    if hospital_id = '1426' then hospital_name = 'HUNTINGTON BEACH
HOSPITAL';

```

```

    if hospital_id = '1636' then hospital_name = 'TRI-CITY MEDICAL
CENTER';
    if hospital_id = '1643' then hospital_name = 'ALHAMBRA HOSPITAL
MEDICAL CTR';
    *if hospital_id = '1716' then hospital_name = 'WESTERN MED CTR-SANTA
ANA';

    if hospital_id = '207' then hospital_name = 'RONALD REAGAN UCLA
MEDICAL CENTER';
    if hospital_id = '483' then hospital_name = 'MEMORIAL HOSPITAL';

    hospital_name = strip(hospital_name);
    hosp_name_source = coalescec(hosp_name_source, 'clm');

/*****/

    dob = datepart(dob);
    format age_gp $10.;
    age_at_disch = int((dis_dt - dob) / 365.25);

    if dob > adm_dt then age_gp = 'NEG';
    else if age_at_disch <= 5 then age_gp = '0-5';
    else if age_at_disch <= 17 then age_gp = '06-17';
    else if age_at_disch <= 45 then age_gp = '18-45';
    else if age_at_disch <= 64 then age_gp = '46-64';
    else if age_at_disch <= 74 then age_gp = '65-74';
    else if age_at_disch <= 84 then age_gp = '75-84';
    else if age_at_disch <= 90 then age_gp = '85-90';
    else if age_at_disch <= 96 then age_gp = '91-96';
    else if age_at_disch <= 102 then age_gp = '97-102';
    else age_gp = 'above 103';
run;

* 9/19/2016 --- ;
* Sep 2016 KPI run - Array is set from Jan 2011 to Sep 2016. This
works for looking back 1 year or 120 days and check the gap.
* For those who discharged in Sep 2016, 30 days after the discharge
date is some times in Oct, which exceeds the range of array;
* We can look at up to 30 days after discharge date when dis_dt is in
a previous month.
* If the KPI run month/year is the same as dis_dt month/year, we
cannot look at 30 days after discharge date simply because the data is
not available.
* In other words, we can look at gaps up to August for those people
who have dis_dt in Sep --> set "To" below to 8 (August) using j-1;

%let t_day=%sysfunc(intnx(day,%sysfunc(today()),0),mmddyy10.);
%put &t_day;

data foo2;
    length ppg2 $7;
    set foo ;

```



```

format t_dayn mmddyy10.;

t_dayn=%sysfunc(inputn(&t_day.,mmddyy10.));

*array ym(70) YM201101--YM201610;
array ym(&array_len.) YM&beg_mon1.--YM&mth0.; /*Data for KPI
running month included in MEMMO */

ppg2 = coalescec(ppg2, "(Blank)");

/*****

*Enrollment;
if mem = 1 and product_code in ('10','40','60','70','80','90') then
do;
  if product_code = '10' then from = dis_dt - 120;
  else from = dis_dt - 365;
  *doesn't matter for 80 and 90;
  to = dis_dt + 30;

  i = (year(from) - %substr(&beg_mon1.,1,4)) * 12 + month(from);
  d = (year(dis_dt) - %substr(&beg_mon1.,1,4)) * 12 + month(dis_dt);
  j = (year(to) - %substr(&beg_mon1.,1,4)) * 12 +
month(to);

  gap1a = 0;
  gap1b = 0;
  do k=i to d-1;
    gap1a + (ym(k) = .);
  end;

  /* 9/19/2016 - if discharge month and year are the same as the
run-date month and year, set j-1;
This will reset "to" to Aug ("to" was originally set to 9 as
Sep). This reset prevents "j" to go beyond the array range */
  if month(dis_dt)=month(t_dayn) and year(dis_dt)=year(t_dayn) then
do;
  do k=d to j-1;
    gap1b + (ym(k) = .);
  end;
end;
  if month(dis_dt)^=month(t_dayn) and year(dis_dt)^=year(t_dayn)
then do;
  do k=d to j;
    gap1b + (ym(k) = .);
    /*if ym(k) = . then gap1b = 1 + gap1b;*/
  end;
end;

```

```

        if gap1a + gap1b = 0 or (product_code in ('40','60','70') and
gap1a = 1 and gap1b = 0) or product_code in ('80','90') then
enrollment = 'Y'; *no restriction for 80 and 90 for now;
        end;

/*drop MHC_MEMBER_NO ym: from to i d j k gap: mem enrollment hosp_id
hosp_name case_pr: dob;*/
run;

data foo3; set foo2;
    if (month(dis_dt) = 12 and day(dis_dt) > 1) then bad_dates = 'Y';
    if (product_code = '10' and age_at_disch < 21) or (product_code in
('40','60','70') and age_at_disch < 18) then underage = 'Y';          *no
restriction for 80 and 90 for now;
    if enrollment = '' then bad_enrollment = 'Y';

    *denom = (cat(pregnancy, expired, plan_stay_next, underage,
bad_enrollment, same_day, bad_dates) = '');          *HEDIS definition;
    denom = (cat(pregnancy, expired, plan_stay_next, underage,
bad_enrollment, same_day) = '');          *No
restriction on December;
    readmit7 = (denom = 1 and 0 <= next <= 7);
    readmit14 = (denom = 1 and 0 <= next <= 14);
    readmit = (denom = 1 and 0 <= next <= 30);

    drop MHC_MEMBER_NO ym: from to i d j k t_dayn gap: mem enrollment
hosp_id hosp_name case_pr: dob;
run;

data readm_detail(rename=(dis_dt_c=dis_dt adm_dt_c=adm_dt));
    /*retain CIN_NO MEMBER_NO CASE_ID id tag adm_dt dis_dt yearmonth prev
next
age_at_disch age_gp GENDER PRODUCT_CODE BP spd dual DHS PPG_CODE
/*PPG_DESCRIPTION*/ /*ppg_name1 ppg_name2 ppg_name3 ppg2
/*ppg_desc_combined ppg_combined */*/
pregnancy expired plan_stay_next bad_dates underage bad_enrollment
same_day denom readmit readmit2
CASE_DX1-CASE_DX21 site_no hospital_id hospital_name TANF MCE; */

/* 9/21/2016 - There are variables that are not included in the retain
below are included in the Altaf's template table.
Just include them and make it clear that they are in. Readmit2 is not
included in the tempalte, so exclude it from retain */
    retain CIN_NO MEMBER_NO CASE_ID id tag adm_dt dis_dt yearmonth prev
next
age_at_disch age_gp GENDER PRODUCT_CODE BP /*spd*/ SEGMENT dual DHS
PPG_CODE /*PPG_DESCRIPTION*/ ppg_name1 ppg_name2 ppg_name3 ppg2
/*ppg_desc_combined ppg_combined */
pregnancy expired plan_stay_next bad_dates underage bad_enrollment
same_day denom readmit readmit14 readmit7 /*readmit2*/

```

```
CASE_DX1-CASE_DX21 site_no hospital_id hospital_name /*TANF MCE*/  
ADJ_BILL_AMT_CASE PAID_AMT_CASE ICD_FLAG APRDRG SEVERITY TRANSFER  
HOSP_NAME_SOURCE AID_CODE  
FIRST_NAME LAST_NAME;
```

```
length CASE_ID $32. MEMBER_NO $16.;  
set foo3;  
*readmit2 = (lag(readmit) = 1);  
array datevars dis_dt adm_dt;  
length dis_dt_c adm_dt_c $10;  
array datevars_c dis_dt_c adm_dt_c;  
do i=1 to 2;  
    datevars_c{i}=put(datevars{i},mmddy10.);  
end;  
drop dis_dt adm_dt i;  
run;
```

```
proc sql;  
create table out.readm_detail as  
select a.*  
    ,case when b.cin_no is null then 0 else 1 end as found_in_memmo  
from readm_detail a left join out.memmo b  
on a.cin_no = b.cin_no and a.product_code=b.product_code and a.yearmth  
= b.yearmth;  
quit;
```

```
proc datasets lib=work memtype=data kill; run;
```

```
%pgm_begin_end(e)
```