```
**-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
*-*-*-*-*-*-*-*-*-*-*-*-*
**
**
** PROJECT NAME       : APR-DRG Inpatient and SNF cost estimates
     **
** PROGRAM NAME       : Req711.sas
                                 **
** PROGRAM LOCATION   : \\svm3cifs\SASGrid_Data\SAS
Data\yonsu\Req711\pgm                      **
** AUTHOR             : Yonsu Kim
                **
**
**
** ----------------------------------------------------------------------
--------------------- **
**
**
** REQUESTER          : Michael Tu
                **
** PURPOSE            : Estimate cost of Inpatient and SNF cases
                      **
** OUTPUT             : \\svm3cifs\SASGrid_Data\SAS
Data\yonsu\Req711\excel                    **
**
**
**-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
*-*-*-*-*-*-*-*-*-*-*-*-*
**
**
** Please use APRDRG and APRDRG_WT in HOA Repository to estimate costs
of Inpatient and     **
** SNF cases.
**
** Please use code sample below for filenames and references.
**
** Please multiply APRDRG_WT by Califormia Medi-cal reimbursement base
rate ($7,936)        **
** to obtain costs.
**
** Create summary by APR-DRG, APR-DRG description, year of service and
product line and     **
** segment.
**
**
**
**-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
*-*-*-*-*-*-*-*-*-*-*-*-*/;
```

```
libname clm "\\svm3cifs\sasgrid_data\SAS Data\shared\production\claim"
access=readonly;
libname enc "\\svm3cifs\sasgrid_data\SAS
Data\shared\production\encounter" access=readonly;
libname qnxt "\\svm3cifs\sasgrid_data\SAS Data\shared\production\qnxt"
access=readonly;
libname m "\\svm3cifs\sasgrid_data\SAS
Data\shared\production\membership" access=readonly;
libname excel "\\SVM3CIFS\SASGRID_DATA\SAS DATA\yonsu\REQ711\excel";


DATA ENC;
SET ENC.ENC_CASE_INPSNF;
 RUN;


DATA CLM;
SET CLM.CLM_CASE_INPSNF;
 RENAME BP_CODE=BP;
 RUN;


 DATA QNXT;
SET QNXT.QNXT_CASE_INPSNF;
 RENAME BP_CODE=BP;
 RUN;

/* 1. CREATING TABS THAT HAS 3 COLUMNS (CLM, ENC & QNXT) */
/** 1.1 JOINING MEMMO AND ENC **/
/*PROC CONTENTS DATA=ENC;RUN;/*462419*/*/;



%MACRO CALL(TABLE);

 PROC SQL;
 CREATE TABLE &TABLE._2 AS SELECT
    A.CIN_NO,
      A.CASE_ID,
    A.ADM_MONTH,
    A.ADM_DT,
      A.DIS_DT,
    A.APRDRG,
    A.APRDRG_WT,
      A.SRV_CAT,
    A.PRODUCT_CODE,
    B.BP,
    B.SEGMENT,
    CASE
       WHEN ADM_DT BETWEEN '31DEC2012'D AND '01JAN2012'D THEN '2012'
       WHEN ADM_DT BETWEEN '31DEC2013'D AND '01JAN2013'D THEN '2013'
       WHEN ADM_DT BETWEEN '31DEC2014'D AND '01JAN2014'D THEN '2014'
```

```
        WHEN ADM_DT BETWEEN '31DEC2015'D AND '01JAN2015'D THEN '2015'
        WHEN ADM_DT BETWEEN '31DEC2016'D AND '01JAN2016'D THEN '2016'
        WHEN ADM_DT BETWEEN '31DEC2017'D AND '01JAN2017'D THEN '2017'
        ELSE 'OTHER'
          END AS ADM_YR,
       CASE
          WHEN ADM_DT BETWEEN '30JUN2018'D AND '01JUL2017'D THEN 7936
        WHEN ADM_DT BETWEEN '30JUN2017'D AND '01JUL2016'D THEN 7436
        WHEN ADM_DT BETWEEN '30JUN2016'D AND '01JUL2015'D THEN 7436
        WHEN ADM_DT BETWEEN '30JUN2015'D AND '01JUL2014'D THEN 7373
        WHEN ADM_DT BETWEEN '30JUN2014'D AND '01JUL2013'D THEN 7200
        ELSE 7200
          END AS BASE
 FROM &TABLE AS A
 LEFT JOIN M.MEMMO AS B
      ON (A.CIN_NO=B.CIN_NO AND A.ADM_MONTH=B.YEARMTH AND
A.PRODUCT_CODE=B.PRODUCT_CODE)
ORDER BY CIN_NO, ADM_MONTH;
QUIT;

DATA _&TABLE;
SET &TABLE._2;
COST=APRDRG_WT*BASE;
RUN;

PROC SORT DATA=_&TABLE
DUPOUT=DUPOUT
NODUPKEY;
BY CASE_ID CIN_NO ADM_DT DIS_DT PRODUCT_CODE;
RUN;

%MEND;
%CALL (ENC);
%CALL (CLM);
%CALL (QNXT);



/* AFTER REMOVING DUPLICSTES, OBS. DECREASED FROM 462,419 TO 459,826
IN ENC3,
WHICH IS SAME AS OBS. (459,826) OF THE TABLE WHICH REMOVED DUPLICATES
FROM ENC_CASE_INPSNF (#462,419) */



/** 1.2 BY APRDRG (APRDRG description), ADM_YR (year of service) and
PRODUCT_CODE (product line) and segment
- INCLUDING DUPLICATES - CREATING 12 TABS **/

%MACRO COST(BY, TABLE);

PROC SQL;
```

```
CREATE TABLE COST_&TABLE._&BY AS
      SELECT
           &BY,
              SUM(COST) AS TCOST_&BY format DOLLAR15.2
FROM _&TABLE
GROUP BY &BY;
QUIT;

PROC EXPORT DATA= COST_&TABLE._&BY
      outfile = "\\svm3cifs\SASGrid_Data\SAS
Data\yonsuk\Req711\excel\COST_EST.XLSX"
      DBMS=xlsx REPLACE;
      sheet="&TABLE._&BY";
run;

%MEND;

%COST (ADM_YR, ENC);
%COST (ADM_YR, CLM);
%COST (ADM_YR, QNXT);
%COST (PRODUCT_CODE, ENC);
%COST (PRODUCT_CODE, CLM);
%COST (PRODUCT_CODE, QNXT);
%COST (SEGMENT, ENC);
%COST (SEGMENT, CLM);
%COST (SEGMENT, QNXT);
%COST (APRDRG, ENC);
%COST (APRDRG, CLM);
%COST (APRDRG, QNXT);

/* END */


/** 1.3 Joining total cost for three tables (CLM, ENC & QNXT) */

PROC SQL ;
  CREATE TABLE SEGMENT AS SELECT
     C.SEGMENT,
       C.TCOST_SEGMENT AS CLM,
       D.TCOST_SEGMENT AS ENC,
       E.TCOST_SEGMENT AS QNXT
  FROM COST_CLM_SEGMENT AS C
  FULL JOIN COST_ENC_SEGMENT AS D
  ON (C.SEGMENT=D.SEGMENT)
  FULL JOIN COST_QNXT_SEGMENT AS E
  ON (C.SEGMENT=E.SEGMENT)
  ORDER BY SEGMENT;
  QUIT;

  PROC SQL ;
  CREATE TABLE ADM_YR AS SELECT
     C.ADM_YR,
```

```
      C.TCOST_ADM_YR AS CLM,
      D.TCOST_ADM_YR AS ENC,
      E.TCOST_ADM_YR AS QNXT
  FROM COST_CLM_ADM_YR AS C
  FULL JOIN COST_ENC_ADM_YR AS D
  ON (C.ADM_YR=D.ADM_YR)
  FULL JOIN COST_QNXT_ADM_YR AS E
  ON (C.ADM_YR=E.ADM_YR)
  ORDER BY ADM_YR;
  QUIT;

  PROC SQL ;
  CREATE TABLE PRODUCT_CODE AS SELECT
     D.PRODUCT_CODE,
      C.TCOST_PRODUCT_CODE AS CLM,
      D.TCOST_PRODUCT_CODE AS ENC,
      E.TCOST_PRODUCT_CODE AS QNXT
  FROM COST_CLM_PRODUCT_CODE AS C
  FULL JOIN COST_ENC_PRODUCT_CODE AS D
  ON (C.PRODUCT_CODE=D.PRODUCT_CODE)
  FULL JOIN COST_QNXT_PRODUCT_CODE AS E
  ON (D.PRODUCT_CODE=E.PRODUCT_CODE)
  ORDER BY PRODUCT_CODE;
  QUIT;

  /** 1.4 Interleaving 'APRDRG' - Because # OF APRDRG ID varies with
tables **/

  DATA APRDRG_IL;
       SET COST_CLM_APRDRG (KEEP=APRDRG) COST_ENC_APRDRG (KEEP=APRDRG)
COST_QNXT_APRDRG (KEEP=APRDRG);
        BY APRDRG;
        RUN;

  PROC SORT DATA=APRDRG_IL
  DUPOUT = DUPOUT
  NODUPKEY;
  BY APRDRG;
  RUN;

  PROC SQL ;
  CREATE TABLE APRDRG AS SELECT
     G.APRDRG AS APRDRG,
      C.TCOST_APRDRG AS CLM,
      D.TCOST_APRDRG AS ENC,
      E.TCOST_APRDRG AS QNXT
  FROM APRDRG_IL AS G
  FULL JOIN COST_CLM_APRDRG AS C
  ON (G.APRDRG=C.APRDRG)
  FULL JOIN COST_ENC_APRDRG AS D
  ON (G.APRDRG=D.APRDRG)
  FULL JOIN COST_QNXT_APRDRG AS E
```

```
  ON (G.APRDRG=E.APRDRG)
  ORDER BY APRDRG;
  QUIT;


  %MACRO OUT (BY);
  PROC EXPORT DATA= &BY
     OUTFILE = "\\SVM3CIFS\SASGRID_DATA\SAS
DATA\YONSUK\REQ711\EXCEL\COST4_EST.XLSX"
     DBMS=XLSX REPLACE;
     SHEET="&BY";
   RUN;

   %MEND;
   %OUT (ADM_YR);
   %OUT (APRDRG);
   %OUT (PRODUCT_CODE);
   %OUT (SEGMENT);

/*END OF ESTIMATEING BY TABLE THAT HAS 3 COLUMN(CLN, ENC, AND QNXT) */




/* 2. SUM (TOTAL COST) OF CLM, ENC AND QNXT AFTER REMOVING DUPLICATES
-
  ONLY 1 COLUMN FOR TOTAL COST */;


DATA APRDRG_SUM;
   SET _CLM _ENC _QNXT;
RUN;

PROC CONTENTS DATA=APRDRG_SUM; RUN; /*979,660 */

PROC SORT DATA=APRDRG_SUM
DUPOUT=DUPOUT
NODUPKEY;
BY CIN_NO ADM_DT DIS_DT SRV_CAT;
RUN;                                  /* Desceased to 928,007 by 51,653
*/


/** 2.1 Estimate total cost (By APRDRG, ADM_YR, PRODUCT LINE, SEGMENT)
**/

%MACRO COST5(BY);

PROC SQL;
CREATE TABLE COST5_&BY AS
      SELECT
```

```
            &By,
              SUM(COST) AS TCOST_&BY format DOLLAR15.2
FROM APRDRG_SUM
GROUP BY &BY;
QUIT;

PROC EXPORT DATA= COST5_&BY
      outfile = "\\svm3cifs\SASGrid_Data\SAS
Data\yonsuk\Req711\excel\COST5_EST.XLSX"
      DBMS=xlsx REPLACE;
      sheet="&BY";
run;

%MEND;

%COST5 (ADM_YR);
%COST5 (PRODUCT_CODE);
%COST5 (SEGMENT);
%COST5 (APRDRG);


/** 2.2 Get APRDRG description and add it to the inital table **/

PROC IMPORT DATAFILE="\\SVM3CIFS\SASGRID_DATA\SAS
DATA\YONSUK\REQ711\APRDRG_DESC.XLSX"
       DBMS=XLSX
       OUT=APRDRG_DESC;
         SHEET='APRDRG_DESC';
       GETNAMES=YES;
RUN;

DATA APRDRG_DESC;
  SET APRDRG_DESC;
  APRDRG_NAME= PUT (DRG_DESCRIPTION,$APRDRG_NM.);
  RUN;


PROC SORT DATA=APRDRG_DESC
  DUPOUT=DUPOUT
  NODUPKEY;
  BY APRDRG;
 RUN;



PROC SQL ;
   CREATE TABLE COST5_APRDRG_2 AS SELECT *
     FROM COST5_APRDRG AS A
     LEFT JOIN APRDRG_DESC AS B
       ON (A.APRDRG=B.APRDRG)
     ORDER BY APRDRG;
QUIT;
```

```
PROC EXPORT DATA= COST5_APRDRG_2
     outfile = "\\svm3cifs\SASGrid_Data\SAS
Data\yonsuk\Req711\excel\APRDRG_ONLY.XLSX"
     DBMS=xlsx REPLACE;
     sheet="APRDRG";
run;



/* 3. COST ESTIMATES BY APR-DRG WEIGHT AND YEAR - BY THE ADDITIONAL
REQUST FROM DR. Miller
      - FROM 2  */

PROC SQL;
     CREATE TABLE APRDRG_WT AS SELECT
     ADM_YR,
      APRDRG,
      APRDRG_WT,
      SUM(COST) AS TCOST
     FROM APRDRG_SUM
     GROUP BY ADM_YR, APRDRG, APRDRG_WT;
  QUIT;


PROC SQL ;                                  /* ADDING APR-DRG DESCRIPTION
*/
   CREATE TABLE APRDRG_WT_2 AS SELECT *
     FROM APRDRG_WT AS A
     LEFT JOIN APRDRG_DESC AS B
       ON (A.APRDRG=B.APRDRG)
     ORDER BY APRDRG;
QUIT;

PROC SORT DATA=APRDRG_WT_2;
BY ADM_YR APRDRG APRDRG_WT; RUN;

PROC MEANS DATA=APRDRG_WT_2;
VAR APRDRG_WT; RUN;



/* BELOW TO BE USED TO ESTIMATE COSTS BASED ON THE FIXED RATE ($7,936)
   IT GOES TO THE BEGINNING OF 2 TO BE CONCATENAED */

%MACRO CALL(TABLE);

 PROC SQL;
 CREATE TABLE &TABLE._2 AS SELECT
```

```
    A.CIN_NO,
      A.CASE_ID,
    A.ADM_MONTH,
    A.ADM_DT,
      A.DIS_DT,
    A.APRDRG,
    A.APRDRG_WT,
      A.SRV_CAT,
    A.PRODUCT_CODE,
    B.BP,
    B.SEGMENT,
    CASE
       WHEN ADM_DT BETWEEN '31DEC2012'D AND '01JAN2012'D THEN '2012'
       WHEN ADM_DT BETWEEN '31DEC2013'D AND '01JAN2013'D THEN '2013'
       WHEN ADM_DT BETWEEN '31DEC2014'D AND '01JAN2014'D THEN '2014'
       WHEN ADM_DT BETWEEN '31DEC2015'D AND '01JAN2015'D THEN '2015'
       WHEN ADM_DT BETWEEN '31DEC2016'D AND '01JAN2016'D THEN '2016'
       WHEN ADM_DT BETWEEN '31DEC2017'D AND '01JAN2017'D THEN '2017'
       ELSE 'OTHER'
         END AS ADM_YR,
     CASE
        WHEN ADM_DT BETWEEN '30JUN2018'D AND '01JUL2011'D THEN 7936
       ELSE 7936
         END AS BASE
 FROM &TABLE AS A
 LEFT JOIN M.MEMMO AS B
      ON (A.CIN_NO=B.CIN_NO AND A.ADM_MONTH=B.YEARMTH AND
A.PRODUCT_CODE=B.PRODUCT_CODE)
ORDER BY CIN_NO, ADM_MONTH;
QUIT;

DATA _&TABLE;
SET &TABLE._2;
COST=APRDRG_WT*BASE;
RUN;

PROC SORT DATA=_&TABLE
DUPOUT=DUPOUT
NODUPKEY;
BY CASE_ID CIN_NO ADM_DT DIS_DT PRODUCT_CODE;
RUN;

%MEND;
%CALL (ENC);
%CALL (CLM);
%CALL (QNXT);


/* END */
```

```
DATA TEST; SET ENC CLM QNXT; RUN;
DATA TEST13; SET TEST;
IF ADM_DT =< '31DEC2013'D AND ADM_DT >='01JAN2013'D;
RUN;

PROC SORT DATA=TEST13
DUPOUT=DUPOUT
NODUPKEY;
BY APRDRG APRDRG_WT;
RUN;

PROC CONTENTS DATA=TEST13; RUN; /* #1,084 */


PROC SORT DATA=TEST13
DUPOUT=DUPOUT
NODUPKEY;
BY APRDRG;
RUN;




PROC SORT DATA=TEST;
BY APRDRG APRDRG_WT;
RUN;

PROC MEANS DATA=TEST; /* 0.88*/
VAR APRDRG_WT; RUN;


PROC MEANS DATA=TEST13; /*2016=1.88*/
VAR APRDRG_WT; RUN;

PROC MEANS DATA=CLM.CLM_CASE_INPSNF; /*0.90 */
VAR APRDRG_WT; RUN;

PROC MEANS DATA=ENC.ENC_CASE_INPSNF; /*0.82 */
VAR APRDRG_WT; RUN;

PROC MEANS DATA=QNXT.QNXT_CASE_INPSNF; /*1.072 */
VAR APRDRG_WT; RUN;


PROC FREQ DATA=APRDRG_SUM;
TABLES APRDRG*APRDRG_WT/ NOPERCENT;
RUN;
```