****

**CHIA Medical Expenditure Trends**

**MA APCD Programming Code**

**(December 2016 Edition)**

As described in the Medical Expenditure Trends [technical appendix](http://www.chiamass.gov/assets/Uploads/enrollment/Medical-Expenditures-Trends-Technical-Appendix-12-2016.pdf), CHIA’s Medical Expenditure Trends reporting monitors medical claims spending in Massachusetts over time, using payers’ Massachusetts All-Payer Claims Database (MA APCD) submissions where possible. Spending is reported for unique Massachusetts residents with private, primary medical health insurance in 12 of the Commonwealth’s largest commercial payers.

Provided below are the methodology and programming code used by CHIA to create the MA APCD membership and spending reports for Medical Expenditure Trends. The methodology and code have been discussed with and reviewed by the top 12 commercial payers (and affiliated entities) but are subject to further development in order to arrive at the most accurate payer-paid and patient-paid costs for each payer. On a case-by-case basis, CHIA has used supplemental data from payers to replace MA APCD financial amounts.

1. **Methodology**

Enrollment is determined if the fifteenth day of the month (i.e., Snapshot Date) is within the Product Enrollment Start Date (ME041) and Product Enrollment End Date (ME042) of a member record (i.e., ME041 ≤ Snapshot Date ≤ ME042). A member record with NULL values for either ME041 or ME042 is considered actively enrolled. Members are distinguished using data element HashCarrierSpecificUniqueMemberID (ME107) by OrgID (ME001).

MA APCD membership reports do not count duplicate member records. A unique member record is selected according to the following criteria, in order of significance:

Subset by the following:

1. Massachusetts resident (ME016=MA)
2. Medical coverage (ME018=1)
3. Primary insurance[[1]](#footnote-1) (ME028=1)
4. Behavioral health benefit flag (ME051=1)

De-duplicate by the following order:

1. Coverage type[[2]](#footnote-2) (ME029)
	1. ME029=(ASO, ASW) for Self-Insured
	2. ME029=(UND) for Fully-Insured
	3. ME029=(STN, OTH) for Other
2. Last activity date (ME056)
3. Member Eligibility ID (Derived-ME05)

Medical claims with calendar year 2014 dates of service are linked to this defined member population using the HashCarrierSpecificUniqueMemberID (ME107 and MC137) by OrgID (ME001 and MC001). See programming code below for more details.

1. **Programming Code[[3]](#footnote-3)**

MA APCD data extraction and aggregation are implemented in SAS, using SQL pass-through to access data in Netezza. The code below may not be inclusive of all data manipulations performed on the final output dataset.

1. **SAS/SQL Code**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 SAS/SQL CODE TO EXTRACT AGGREGATE MEDICAL CLAIMS (MC) DATA WITH MERGED MEMBER ELIGIBILITY

(ME) DETAILS FOR MEDICAL EXPENDITURE TRENDS REPORTING

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*--------------- BEGIN ACTION ITEMS -------------------------------------------

 1. Initialize settings and variables

 2. Perform additional action items below prior to running program

--------------- END ACTION ITEMS ---------------------------------------------;

option compress=binary;

\* Initialize global variables;

%let DataDir= C:\MedicalExpenditureTrends\output \*Output directory;

%let OutFile= MC\_ME\_CY14; \*Name of output SAS dataset;

%let OrgIdList= (290,291,295,296,300,301,302,312,3505,3735,4962,7041,7655,7789,8026,8647,10353,10441,10444,10647,10920,10926,10929,11474,11726,12122,12226,10632); \*List of OrgIDs to include in output dataset;

%let ME\_SYM= 201412; \*Submission Year Month of Member Eligibility (ME) file;

%let FirstSnapshot= '20140115'; \*First snapshot date, format: 'YYYYMMDD';

%let LastSnapshot= '20141215'; \*Last snapshot date, format: 'YYYYMMDD';

%let nMonths= 1; \*# of months to increment snapshot dates, Default=1 (e.g. nMonths=1 means monthly, nMonths=3 means quarterly);

\* Fixed global variables;

libname out "&DataDir.";

%let IncurredStart= %substr(&FirstSnapshot,2,6); \*Used for subsetting claims by IncurredDate;

%let IncurredEnd= %substr(&LastSnapshot,2,6); \*Used for subsetting claims by IncurredDate;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin: Netezza summary data extract

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*--------------- BEGIN ACTION ITEMS -------------------------------------------

 Perform the checks below prior to running code

 1. Check that source data is coming from correct data source

--------------- END ACTION ITEMS ---------------------------------------------;

**proc** **sql**;

 connect to odbc (dsn=odbc\_database);

 create table out.&OutFile. as

 select \* from connection to odbc (

 /\* Determine time span between first snapshot date and last snapshot date \*/

 with dts as (select (months\_between(&LastSnapshot., &FirstSnapshot.)+**1**)/&nMonths. as nTimeSpan

 ),

 /\* Determine list of snapshot dates \*/

 dt as (select add\_months(cast(&FirstSnapshot. as datetime), &nMonths.\*idx) as ss

 from (select &FirstSnapshot. :: date start\_date) starting\_date

 cross join \_v\_vector\_idx

 cross join dts

 where idx < nTimeSpan

 )

 /\* MC Fields \*/

 select mc.OrgID as OrgID\_mc

 , VersionIndicator

 , CapitatedEncounterFlag

 , case when MC.Standardized\_MemberStateorProvince='MA' then **1** else **0** end as MAResident\_mc

 , mc.InsuranceTypeCodeProduct as InsuranceTypeCodeProduct\_mc

 , ClaimStatus

 , mc.IncurredDate as mos /\* mos = Month of Service \*/

 , ClaimLineType

 , DeniedFlag

 , case when PrincipalDiagnosisCleaned between '290' and '31699' then **1** else **0** end as BHprimdx /\* Approximate identification of claims with behavioral health as a primary diagnosis. This inadvertently includes Smoking (305.1) \*/

 , GlobalPaymentFlag

 , PaymentArrangementTypeCleaned

 , TypeOfClaimCleaned

 , HighestVersionDenied

 , HighestVersionIndicator

 /\* ME Fields \*/

 , me.OrgID as OrgID\_me /\* NULL will indicate non-merge \*/

 , me.PrimaryInsuranceIndicator

 , me.MedicalCoverage

 , me.CoverageType

 , me.InsuranceTypeCodeProduct as InsuranceTypeCodeProduct\_me

 , me.SpecialCoverage\_CCflag

 , me.MarketCategoryCode

 , me.MAResident as MAResident\_me

 , me.PrescriptionDrugCoverage

 , me.DentalCoverage

 , me.BehavioralHealthBenefitFlag

 /\* Calculated Fields \*/

 , count(\*) as n\_ClaimLines

 , sum(cast(Quantity as bigint)) as Quantity

 , sum(PatientTotalOutOfPocketAmount) as PatientTotalOutOfPocketAmount

 , sum(ChargeAmountCleaned) as ChargeAmountCleaned

 , sum(PaidAmountCleaned) as PaidAmountCleaned

 , sum(PrepaidAmountCleaned) as PrepaidAmountCleaned

 , sum(CopayAmountCleaned) as CopayAmountCleaned

 , sum(CoinsuranceAmount) as CoinsuranceAmount

 , sum(DeductibleAmount) as DeductibleAmount

 , sum(ExcludedExpensesCleaned) as ExcludedExpensesCleaned

 , sum(CoordinationOfBenefitsTPLLiabilityAmount) as CoordinationOfBenefitsTPLLiabilityAmount

 , sum(OtherInsurancePaidAmountCleaned) as OtherInsurancePaidAmountCleaned

 , sum(MedicarePaidAmountCleaned) as MedicarePaidAmountCleaned

 , sum(AllowedAmountCleaned) as AllowedAmountCleaned

 , sum(NonCoveredAmountCleaned) as NonCoveredAmountCleaned

 , sum(WithholdAmount) as WithholdAmount

 , sum(case when PrincipalDiagnosisCleaned between '290' and '31699' then **1** else **0** end) as nclms\_BHprimdx /\* Approximate identification of claims with behavioral health as a primary diagnosis. This inadvertently includes Smoking (305.1) \*/

 FROM APCD\_Release\_4\_0\_MedicalClaims AS mc

 left join

 (select \*

 from (

 select MemberLinkEID

 , me.OrgID

 , InsuranceTypeCodeProduct

 , MedicalCoverage

 , PrimaryInsuranceIndicator

 , CoverageType

 , MarketCategoryCode

 , cast(ProductIDNumber\_Linking\_ID as char(**20**)) as ProductIDNumber\_Linking\_ID

 , HashCarrierSpecificUniqueMemberID

 , PrescriptionDrugCoverage

 , DentalCoverage

 , BehavioralHealthBenefitFlag

 , prd.ProductName

 , prd.ProductLineOfBusinessModel

 , case when SpecialCoverage in ('CC') then **1** else **0** end SpecialCoverage\_CCflag

 , case when Standardized\_MemberStateorProvince ='MA' then **1** else **0** end as MAResident

 , floor(months\_between(ss,ad.DT)/**12**) as Age

 , case when Standardized\_MemberStateorProvince ='MA' then Standardized\_MemberCounty else 'Other' end as MemberCounty

 , MemberGenderCleaned

 , date\_part('year',ss)\***100** + date\_part('month',ss) as mon /\* Snapshot Year and Month \*/

 , case when lead(me.MemberEligibilityID) over (partition by me.OrgID, HashCarrierSpecificUniqueMemberID, ss

 order by case when Standardized\_MemberStateorProvince='MA' then **1** else **0** end

 , case when MedicalCoverage='1' then **1** else **0** end

 , case when PrimaryInsuranceIndicator in ('1','3','4') then **1** else **0** end

 , case when CoverageType in ('ASO', 'ASW') then **1**

 when CoverageType='UND' then **0**

 else -**1** end

 , LastActivityDate

 , me.MemberEligibilityID)

 is null then **1** else **0**

 end AS to\_keep\_v1

 FROM APCD\_Release\_4\_0\_MemberEligibility AS me

 inner join dt

 on ss between ProductEnrollmentStartDate and isnull(cast(ProductEnrollmentEndDate as datetime), cast('99991231' as datetime))

 left join APCD\_Release\_4\_0\_Product as prd

 on me.OrgID=prd.OrgID and me.ProductIDNumber\_Linking\_ID=prd.LinkingProductID and prd.LinkingProductDelegate=**1**

 left join DATE\_TABLE AS ad

 on me.MemberDateofBirth = ad.DT\_YYYYMMDD

 where me.SubmissionYearMonth=&ME\_SYM.

 ) me

 where to\_keep\_v1=**1**

 ) me

 on me.OrgID=mc.OrgID

 and me.HashCarrierSpecificUniqueMemberID=mc.HashCarrierSpecificUniqueMemberIDCleaned

 and me.mon=mc.IncurredDate

 where mc.IncurredDate between &IncurredStart. and &IncurredEnd.

 and mc.OrgID in &OrgIdList.

 group by mc.OrgID

 , VersionIndicator

 , CapitatedEncounterFlag

 , MAResident\_mc

 , mc.InsuranceTypeCodeProduct

 , ClaimStatus

 , mc.IncurredDate

 , ClaimLineType

 , DeniedFlag

 , BHprimdx

 , GlobalPaymentFlag

 , PaymentArrangementTypeCleaned

 , TypeOfClaimCleaned

 , PatientTotalOutOfPocketAmount

 , HighestVersionDenied

 , HighestVersionIndicator

 /\* ME fields \*/

 , me.OrgID

 , me.PrimaryInsuranceIndicator

 , me.MedicalCoverage

 , me.CoverageType

 , me.InsuranceTypeCodeProduct

 , me.SpecialCoverage\_CCflag

 , me.MarketCategoryCode

 , me.MAResident

 , me.PrescriptionDrugCoverage

 , me.DentalCoverage

 , me.BehavioralHealthBenefitFlag

 order by **1**,**2**,**3**,**4**

 );

 disconnect from odbc;

 **quit**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

End: Netezza summary data extract

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin: Export summary data extract

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Include additional field(s) \*/

**proc** **sql**;

 create table out.&OutFile.\_mod as

 select \*

 , case when OrgID\_me=**.** then **0** else **1** end as Ind\_MergedClaim

 from out.&OutFile.

 ;

 **quit**;

/\* Export to txt-file \*/

**proc** **sql** noprint;

 select '"'||trim(name)||'"'

 into :vars

 separated by "'|'"

 from dictionary.columns

 where upcase(libname)=upcase("out") and

 upcase(memname)=upcase("&OutFile.\_mod")

 ;

 **quit**;

filename outfile "&DataDir.\&OutFile..txt";

**data** \_null\_;

 file outfile delimiter='|' DSD DROPOVER lrecl=**32767**;

 if \_n\_ = **1** then do;

 put &vars.;

 end;

 set out.&OutFile.\_mod;

 put (\_all\_) (+**0**);

 **run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

End: Export summary data extract

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. Include ME028=(3, 4) if the payer stores primary insurance as such in its system. [↑](#footnote-ref-1)
2. This selection hierarchy is to ensure that self-insured plans are given preference over fully-insured plans for members with concurrent enrollment in both self-insured and fully-insured coverage. [↑](#footnote-ref-2)
3. CHIA is providing this methodology and programming code as a convenience. It has been prepared for informational purposes only and is based on information believed to be reliable.  The methodology and programming code are subject to change without notice. CHIA does not provide any guarantee or opinion on its accuracy.  CHIA disclaims any liability for the improper or incorrect use of the information contained herein. [↑](#footnote-ref-3)