

Constructing Analytic Datasets

Data Analytic Core

Fall 2015

Building Denominator, Numerator,
and Rate Files
from Medicare Data

Denominator and Numerator

- For creating rates two types of files are needed
- The denominator file:
 - a patient level file of all Benes meeting eligibility or other cohort criteria
- The numerator file:
 - Claims meeting the definition for the event or measure
 - Summarized to the patient level
 - contains a record for each Bene with variables for numerator events

Building A Denominator File

- Starts with the Medicare Beneficiary enrollment file
- Create flags for filtering and checking the data, such as
 - Fee-for-service (FFS) identifiers
 - Part A and B eligible, non-HMO
 - Age Group
 - Dual Eligible
- Flags used to determine who stays in the cohort and who drops out and why

Building A Denominator File

- Create a variable that defines the cohort to use as the population count (N Benes)
Year_End_Pop =
(FFS and nonHMO and ValidHRR and YrEndAge6599)
- Using SAS data views:
 - Avoid taking up disk space (thru replication of data)
 - Allow access to all Benes
 - Use flags to get desired cohort.

Building A Denominator File: Sample Code

```
data out.bene_util_denom2013 / view=out.bene_util_denom2013;
  *dataviews work best with physical file reference not libref;
  set "/general_data/views/denom2013";
  ageyrend = 2013-year(sdob);
  ageygrp5 = 0*(0<=ageyrend<=64) +
            1*(64<ageyrend<=69) +
            2*(69<ageyrend<=74) +
            3*(74<ageyrend<=79) +~
            4*(79<ageyrend<=84) +
            5*(84<ageyrend<=99) +
            6*(99<ageyrend);
  ageyrend6599 = (1<=ageygrp5<=5);
  elig_int=(endmo-startmo)+1;
  . . .
  do i = startmo to endmo;
    abmos=abmos + (en{i} in ('3','C'));      *a and b;
    hmomos=hmomos + (hm{i} in ('1','2','A','B','C')); *any hmo;
  end;
  fullab=(abmos=elig_int);
  yrendpop=(ageyrend6599 and fullab and hmomos=0);
```

Building A Denominator File: Sample Output

Checking flags for Year-End POP

yrendpop	ageyrend6599	fullab	anyhmo	us50	COUNT	PERCENT
0	0	0	0	0	133434	0.242
	0	0	0	1	1321005	2.392
	0	0	1	0	14603	0.026
	0	0	1	1	149010	0.270
	0	1	0	0	25165	0.046
	0	1	0	1	5689722	10.302
	0	1	1	0	121481	0.220
	0	1	1	1	1895487	3.432
	1	0	0	0	576136	1.043
	1	0	0	1	3353606	6.072
	1	0	1	0	6591	0.012
	1	0	1	1	210264	0.381
	1	1	0	0	136897	0.248
	1	1	1	0	413566	0.749
	1	1	1	1	13305566	24.092
-----					-----	-----
0					27352533	49.527
1	1	1	0	1	27875034	50.473
					=====	=====
					55227567	100.000

Building A Numerator File

- Uses any of the Medicare claims data: e.g. MedPAR, Carrier (Part B), or Outpatient
- Apply definition of event or measure, keep claims that meet the criteria
- Apply standard filters by claim type:
 - selecting only Acute Care/Critical Access Hospitals from MEDPAR
 - Drop invalid claims or multiple submittals from Part B
 - Keep only the final bill and exclude total charge record from Outpatient

Building A Numerator File

- Event or measure definitions supplied by PI or from an Analytic Memo
- Events use billing and service codes such as DRGs, diagnosis, procedure, CPT/HCPC, or revenue center
- Flag or count each event/measure
- Keep the minimum set of variables needed and only the claims having the event
 - to reduce the file size
 - use data views if possible (run times can be long if accessing the data multiple times)
- A numerator file can have multiple events/measures per Bene

Building A Numerator File

- Summarize the file to the Bene level
- Sum the flags or counts
 - reset as needed if counting as only 1 event per Bene
- Examples of events/counts:
 - Number of physician visits
 - Hospital stays by diagnosis
 - Total payments or allowed charges
 - Percent of Benes with an emergency department visit

Building A Numerator File: Sample Code

```
*apply filters to get short stay, acute care/critical access
  hospital claims
;
data outmed.allssmedpar2013 /view=outmed.allssmedpar2013;
  *dataviews work best with physical file reference not libref;
  set  "/general_data/views/medpar2013";
  if sdschrgdt^=. and sadmsndt^=. then
    losdays = sdschrgdt-sadmsndt;
  if losdays = 0 then losdays = 1;  *same-day discharge = 1 day;

*ACH/CAH filter, no SNFs;
*non-Special Unit, length of stay <=365;
prov34=substr(prvnumgrp,3,2);
prov3=substr(prvnumgrp,3,1);
if (prov3='0' or prov34='13') and SSLSSNF^='N'
  and spclunit=' ' and losdays<=365;
```

Building A Numerator File: Sample Code

```
*count each admission;
```

```
anyhosp_cnt=1;
```

```
*hospital payments/reimbursements, use pmt_amt;
```

```
hosp_payments=pmt_amt;
```

```
*check primary diagnosis for diabetes;
```

```
diabpdx_cnt=(substr(dgns_cd1, 1, 3)='250');
```

```
*flag if ER pmt, flag any ER visit w/Diabetes dx;
```

```
erhosp_cnt=(er_amt>0);
```

```
erdiabhosp_cnt=(erhosp_cnt>0 and diabpdx_cnt>0);
```

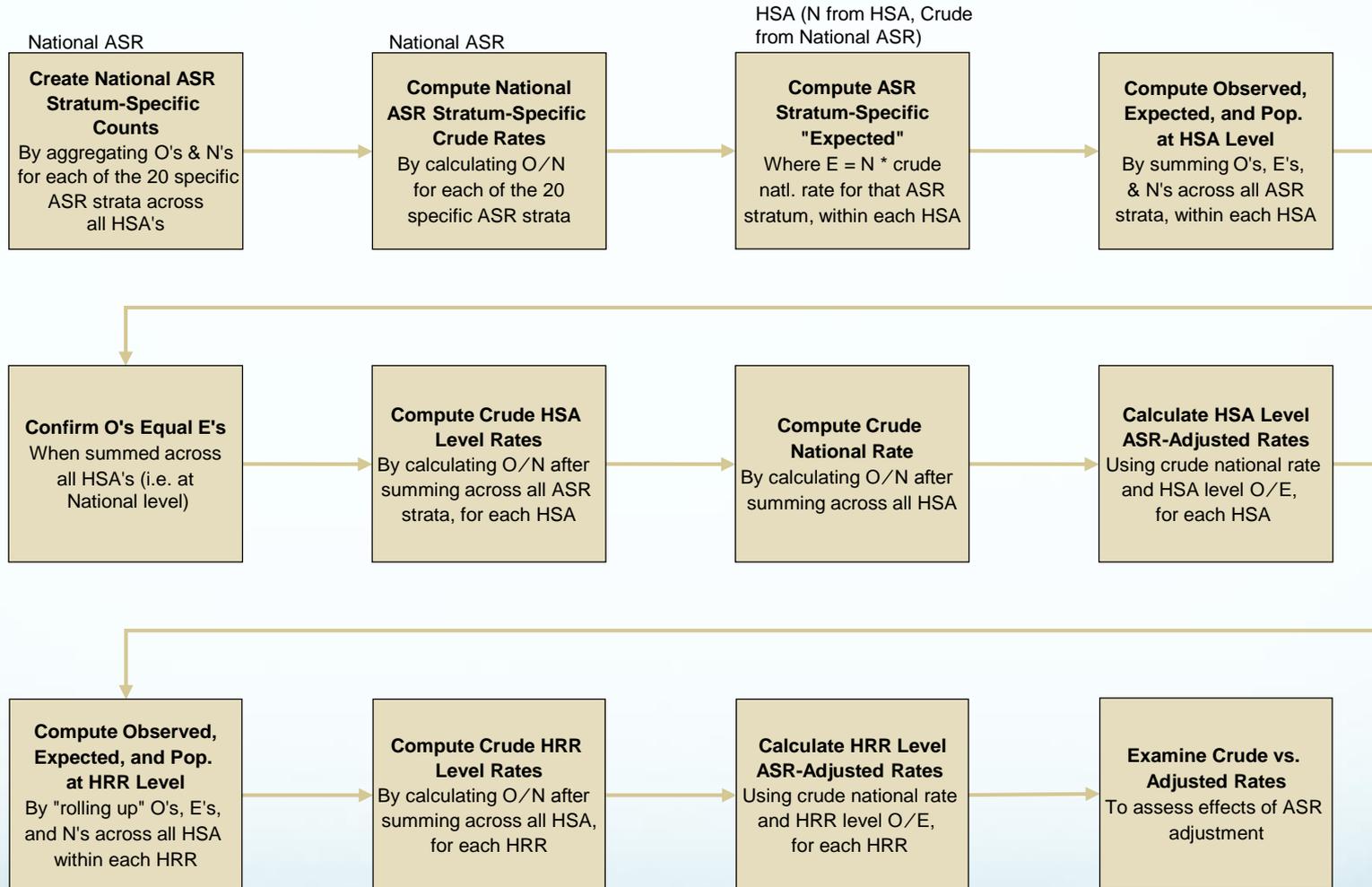
```
*Hip Fracture (femoral neck);
```

```
hip=('820  ' <= dgns_cd1 <= '82099');
```

Generating Rates

- Link the denominator and numerator by Bene ID, keeping all records in the denominator
- There are macros available to produce indirect adjusted rates and keep track of Bene counts to use for suppression
- A final output for rates summarizes the data by geography (e.g. HRR, HSA, State)
- Apply suppression, CMS rule is required
 - CMS rule: suppress if fewer than 11 Benes in a cell (e.g. observed or population counts)
 - TDI rule: apply statistical suppression where expected values are less than 26

The Dartmouth Atlas of Health Care: Rate Calculations for HSAs & HRRs



ASR = AGE (5 levels), SEX (2 levels), RACE (2 levels)

Crude Rate = $\frac{\text{Event Count (Numerator or Observed, O's)}}{\text{Population Count (Denominator or N's)}}$ Adjusted Rate = $\frac{\text{Event Count (O's)}}{\text{Expected Count (E's)}} \times \text{National Crude}$

Generating Rates

- A general use set of rate macros are located in `/general_resources/rate_macros`
- `rate_run.sas`
 - Main driver for indirect adjusted rates
- `rate_confint.sas`
 - Calculates standard error and confidence limits
- `suppress_rates.sas`
 - Applies CMS rule (required) and TDI rule (optional)

Export Rate Data

- Only suppressed data is exported
- One file per event or many events per file
- One row per geography
- For each event variables should include:
 - Population (N)
 - Observed (O)
 - Expected (E)
 - O/E ratio
 - Crude rate
 - Adjusted rate
 - Confidence limits

Export Rate Data

- Data can be converted from SAS datasets using StatTransfer, SAS's proc export or SAS/ODS
- Common data types
 - Stata
 - Excel
 - Access

Export Rate Data: Sample Code

```
proc export
```

```
    data=some.geo_level_rates
```

```
    outfile="geo_level_rates.csv"
```

```
    dbms=csv                /*comma separated value*/
```

```
    replace
```

```
    ;
```

```
run;
```