```
*************************************************************************************************************;
*************************************************************************************************************;
* Author                : Melanie Greenland
* Association           : School of Population and Global Health, The University of Western Australia
* Date                  : March 2018
* Version               : v1.0
* Paper title           : Operationalization and validation of a novel method to calculate adherence to polypharmacy
                        with refill data from the Australian Benefits Scheme (PBS) database
* Description           : SAS program for calculating DPPR from refill data
*************************************************************************************************************;
*************************************************************************************************************;


* need to have variables named the following:
      ID - id of patient
      group - user defined drug group for atc code
      atc_code - specific atc code for the supply
      supply_date - date of supply
      quantity - quantity of medication supplied
      end_date - readmission date (this is the final date used in calculations)
      dose - dose of medication per day
;

* there should be no repeat supply_dates by group, atc_code and ID;
* there should be complete fields for each observation - no missing cells e.g. no missing atc codes;


* import dataset, rename variables and remove labels;
data indata1 (drop = adm_date adm_time sep_date sep_time sex age diag1 dod1 ddd units);
      set <<dataset_name>>;
      rename rootnum = ID  atc7 = atc_code totqty = quantity eop = end_date drg_grp = group supp_date = supply_date;
      attrib _all_ label = '';
      dose = 1;
      if atc7 = 'C07AG02' or atc7 = 'C07AB02_T' then dose = 2;
run;


* sort data by patient ID, drug group and supply date;
proc sort data = indata1;
      by ID group supply_date;
run;
```

```
* calculate duration of supply in days, create variable with first supply date for each group by group and patient;
data indata2 start_dates1 (keep = ID group start_date);
      set indata1;
      by ID group ;
      format start_date ddmmyy10.;
      retain start_date;
      duration = quantity/dose;
      if first.group = 1 then start_date = supply_date;
      start_date = start_date;
      output indata2;
      if first.group = 1 then output start_dates1;
run;

* sort start_dates dataset by ID and start_date, want to find if patient had staggered start dates;
proc sort data = start_dates1;
      by ID start_date;
run;


* if a patient has staggered start dates then class a new supply start as starting a new period;
* want number of periods by patient and the start date of each new period;
data start_dates2 (drop = prev_start) start_dates3 (keep = ID period rename = (period = total_periods));
      set start_dates1;
      by ID;
      format prev_start ddmmyy10.;
      retain period prev_start;
      if first.id = 1 then do;
            period = 1;
            prev_start = start_date;
      end;
      if prev_start < start_date then period = period + 1;
      prev_start = start_date;
      output start_dates2;
      if last.id = 1 then output start_dates3;
run;

* merge the start date of each period and the total number of periods together by ID;
data start_dates4;
      merge start_dates2 start_dates3;
      by ID;
run;
```

```
* want to calculate a variable called repeats which will indicate how many times to repeat the supply record (proxy
start date to have clean calulations);
* if a patient has staggered supply starts then we need to create proxy supplies (0 quantity) but start date of new
period so need to create repeats now;
data repeats1 repeats2;
      set start_dates4;
      repeats = total_periods - period;
      output repeats1;
      do repeat_i = 1 to repeats;
            output repeats2;
      end;
run;

* set the two datasets together: one with all actual supplies and dataset with proxy supply dates (repeated supplies
with start_date missing);
data repeats3;
      set repeats1 repeats2;
      if repeat_i >=1 then start_date = .;
      if repeat_i >=1 then period = period + repeat_i;
      if repeat_i = . then repeat_i = 0;
      drop repeats;
run;

* sort by patient ID, period and repeat index;
proc sort data = repeats3;
      by ID period repeat_i;
run;

* retain down the supply date of new period for the new proxy repeat supplies;
data repeats4 (drop = start_date repeat_i);
      set repeats3;
      by ID period ;
      format supply_date ddmmyy10.;
      retain supply_date;
      if first.ID = 1 then supply_date = start_date;
      if start_date ne . then supply_date = start_date;
run;
```

```
* read from dataset repeats4, if the ID, group and supply date match to a record in indata2 then keep and output as
match;
* read from dataset repeats4, if the ID, group and supply date do not match to a record in indata2 then keep and output
as indata2_only;
data match indata2_only (drop = total_periods period);
      if 0 then set repeats4;
      if _n_ = 1 then do;
            declare hash a(dataset:'repeats4');
            a.defineKey('ID', 'group', 'supply_date');
            a.defineData(all: 'yes');
            a.defineDone();
      end;
      set indata2;
      if a.find() = 0 then output match;
      if a.find() ne 0 then output indata2_only;
run;

* read from dataset indata2, if the ID, group and supply date do not match to a record in repeats4 then keep and output
as repeats4_only;
data repeats4_only (keep = id group supply_date total_periods period);
      if 0 then set indata2;
      if _n_ = 1 then do;
            declare hash b(dataset: 'indata2');
            b.defineKey('ID', 'group', 'supply_date');
            b.defineData(all: 'yes');
            b.defineDone();
      end;
      set repeats4;
      if b.find() ne 0 then output repeats4_only;
run;

* set all three datasets together with their key (match, original supply or dummy supply);
data all1;
      set match (in = match) indata2_only (in = d2) repeats4_only (in = d7);
      format source $8.;
      if match then source = 'match';
      if d2    then source = 'original';
      if d7  then source = 'dummy';
      if duration = . AND source = 'dummy' then duration = 0;
run;

* sort patient id, supply date, descending period in order to retain dates for next step;
proc sort data = all1;
      by ID supply_date descending period ;
run;
```

```
* retain down period and total_periods for original supplies;
data all2 (drop = total_periods_retain period_retain start_date);
      set all1;
      by ID period notsorted;
      retain total_periods_retain period_retain;
      if first.ID = 1 then do;
            total_periods_retain = .;
            period_retain = .;
      end;
      if total_periods ne . then total_periods_retain = total_periods;
      total_periods = total_periods_retain;
      if period ne . then period_retain = period;
      period = period_retain;
run;

* sort data;
proc sort data = all2;
      by ID group period supply_date atc_code;
run;

* retain down atc_code, quantity, end_date and dose for dummy supplies;
data all3;
      set all2;
      retain _quantity _end_date _dose _atc_code;
      if quantity ne . then _quantity = quantity;
      else quantity = _quantity;
      if end_date ne . then _end_date = end_date;
      else end_date = _end_date;
      if dose ne . then _dose = dose;
      else dose = _dose;
      if atc_code ne '' then _atc_code = atc_code;
      else atc_code = _atc_code;
      drop _quantity _end_date _dose _atc_code;
run;

* create next_supply_date and next_atc_code variables;
data all4;
      recno = _n_ + 1; /* select next observation */
      set all3 end = last; /* last is the end observation in the dataset */
      if last ne 1 then set all3 (keep = supply_date atc_code rename = (supply_date = next_supply_date atc_code =
next_atc_code)) point = recno;
      else do;
            next_supply_date = .; /* set the last observation of the dataset to missing for the two new variables */
            next_atc_code = '';
      end;
run;
```

```
* need to set the last supply date for each group to end_date and set next_atc_code to missing for last group,
* also flag if next supply is different by atc_code as need to discard previous supplies if changing medication ;
data all5;
      set all4;
      by ID group;
      if last.group = 1 then do;
            next_supply_date = end_date;
            next_atc_code = '';
      end;
      if last.group ne 1 and atc_code ne next_atc_code then next_atc_code_diff = 1;
run;

* calculate time to next supply in days, difference in days between duration of supply and time to next supply,
* possible oversupply after second supply date, possible number of days not on drugs between two dates;
data all6;
      set all5;
      by ID group;
      ttns = next_supply_date - supply_date;
      diff_days = ttns - duration;
      if diff_days < 0 then do;
            if atc_code = next_atc_code then poss_ovs = abs(diff_days);
            if atc_code ne next_atc_code then poss_ovs = 0;
      end;
      if diff_days >= 0 then do;
            poss_ovs = 0;
            poss_dnod = diff_days;
      end;
      if diff_days < 0 then poss_dnod = 0;
run;

* calculate actual oversupply, retain down for future 'gaps' to calculate actual days not on drug ;
data all7;
      set all6;
      by ID group;
      retain act_ovs 0;
      if first.group = 1 then act_ovs = 0;
      act_ovs = act_ovs + poss_ovs - poss_dnod;
      if act_ovs <= 0 then act_dnod = abs(act_ovs); /* set act_dnod to number of days without supply */
      if act_ovs > 0 then act_dnod = 0; /* set act_dnod to 0 if there is oversupply */
      if act_ovs <= 0 then act_ovs = 0; /* finally set act_ovs to 0 if less than 0 */
run;
```

```
* now we're only really interested in days not on drugs to calculate dppr;
* calculate cumulative number of days in period by ID, group and period (sum together time to next supply by period);
* calculate cumulative number of days in period not on drug by ID, group and period (sum together act_dnod by period);
data all8;
      set all7;
      by ID group period;
      retain period_group_dnod period_days;
      if first.period = 1 then do;
            period_group_dnod = 0;
            period_days = 0;
      end;
      period_group_dnod = period_group_dnod + act_dnod;
      period_days = period_days + ttns;
run;

* keep the last observation with cumulative sums on for calculations by ID, group and period;
data periods1 (keep = ID total_periods group period period_group_dnod period_days);
      set all8;
      by ID group period;
      if last.period = 1;
run;

proc sort data = periods1;
      by ID period group;
run;

* calculate cumulative days not on drug by period, number of groups in period, dppr numerator by period, dppr
denominator by period;
data periods2 (keep = ID period total_periods no_groups period_num period_den);
      set periods1;
      by ID period;
      retain period_dnod no_groups;
      if first.period = 1 then do;
            period_dnod = 0;
            no_groups = 0;
      end;
      period_dnod = period_dnod + period_group_dnod;
      no_groups = no_groups + 1;
      if last.period = 1 then do;
            period_num = period_days - period_dnod/no_groups;
            period_den = period_days;
      end;
      if last.period = 1;
run;
```

```
* calculate dppr by ID;
data dppr (keep = ID total_periods no_groups DPPR);
      set periods2;
      by ID;
      format dppr 8.2;
      retain num den ;
      if first.ID = 1 then do;
            num = 0;
            den = 0;
      end;
      num = num + period_num;
      den = den + period_den;
      if last.ID = 1 then dppr = (num/den)*100;
      if last.ID = 1;
run;

* keep only dppr dataset;
proc datasets lib = work nolist;
      save dppr ;
quit;
```