PHARMO
Institute for Drug Outcomes Research

Patient Journey toolbox

Berber Snoeijer

Jun-2013

Creating patient journeys based on outpatient pharmacy data using an extended macro toolbox

Content of presentation

- PHARMO outpatient database
- Patient Journey requirements
- Macro Toolbox development
- Tips & Trics

The PHARMO outpatient pharmacy database

- Millions of records
- Covers about 20% of Dutch population

- Information on:
  - What is delivered to the patient
  - When it is delivered
  - By whom it is prescribed
  - For how long is it prescribed
- Each delivery is one row

- Linked to other databases like physician data, hospitalization, lab data etc.

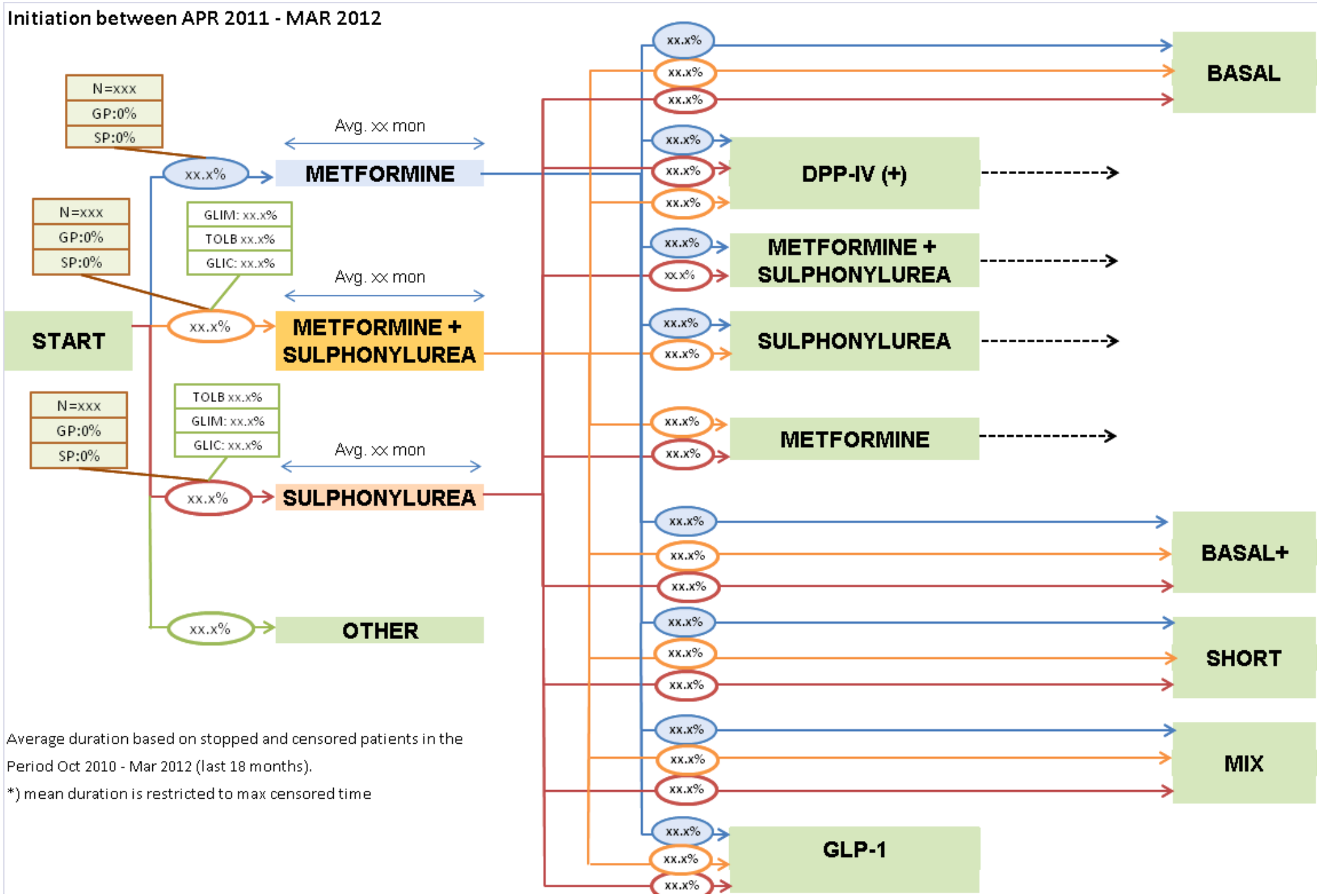
The patient treatment journey

- Overview of treatment changes during the patients treatment life cycle

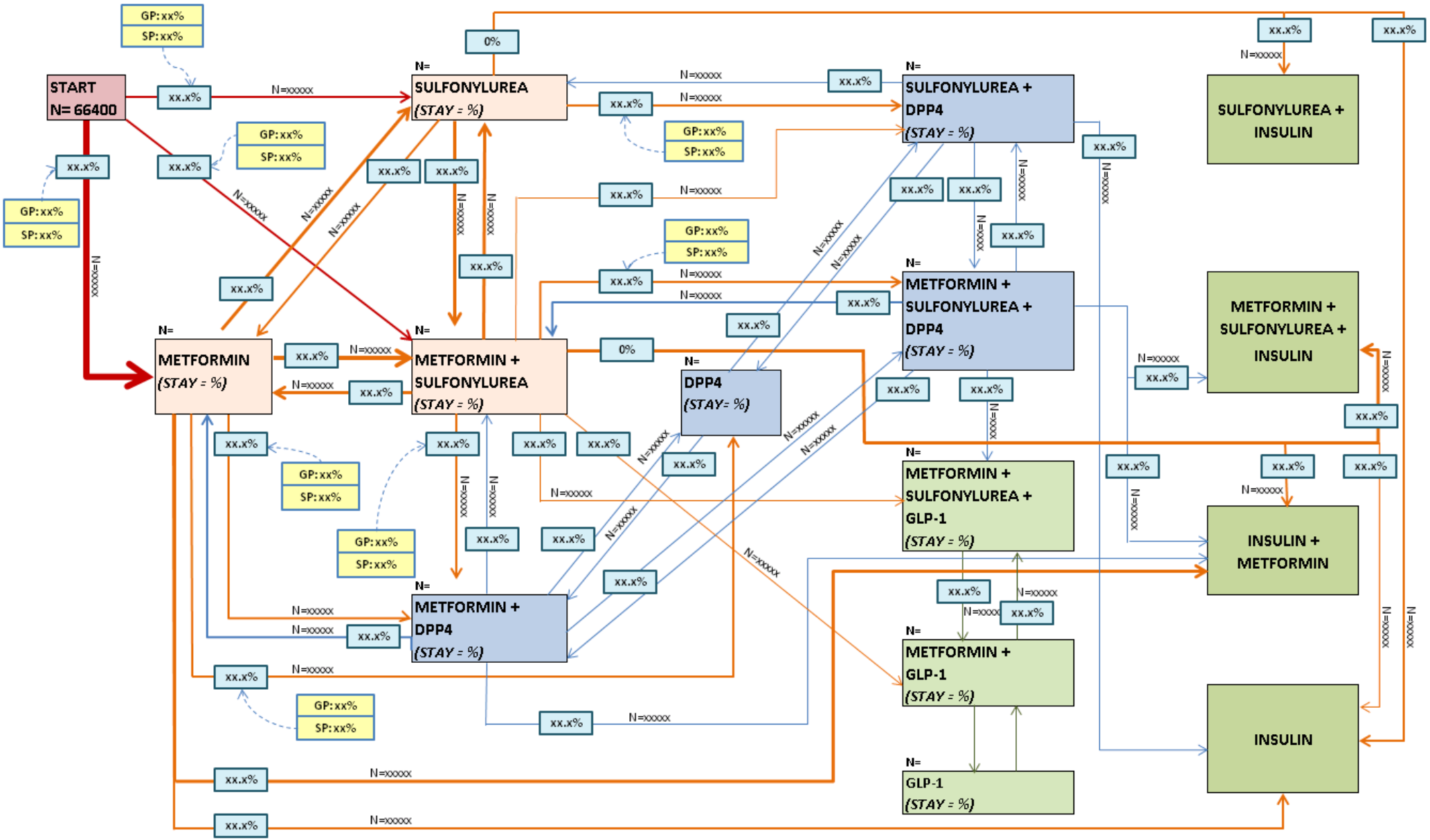
- Basic characteristics:
  - Step in therapy line
  - Start of each step
  - Who initiated the new step (GP or Specialist)
  - Drug/Brand on which the patient was started

The patient journey

- Interesting for pharmaceutical companies
  - New drugs are often registered for a specific patient group or therapy step.
  - Much more collaboration between pharmaceutical company, insurance company, physicians, pharmacists etc.
- More information needed
  - How many patients in which therapy line?
    - Overall for market potential
    - Of the specific product for market share
    - Characteristics of the patients using their product
  - How often do patients switch from therapy, product, prescriber etc..


Initiation between APR 2011 - MAR 2012
START
N=xxx
GP:0%
SP:0%
xx.x%
METFORMINE
Avg. xx mon
GLIM: xx.x%
TOLB xx.x%
GLIC: xx.x%
METFORMINE + SULPHONYLUREA
TOLB xx.x%
GLIM: xx.x%
GLIC: xx.x%
SULPHONYLUREA
OTHER
BASAL
DPP-IV (+)
METFORMINE + SULPHONYLUREA
SULPHONYLUREA
METFORMINE
BASAL+
SHORT
MIX
GLP-1
Average duration based on stopped and censored patients in the Period Oct 2010 - Mar 2012 (last 18 months).
*) mean duration is restricted to max censored time

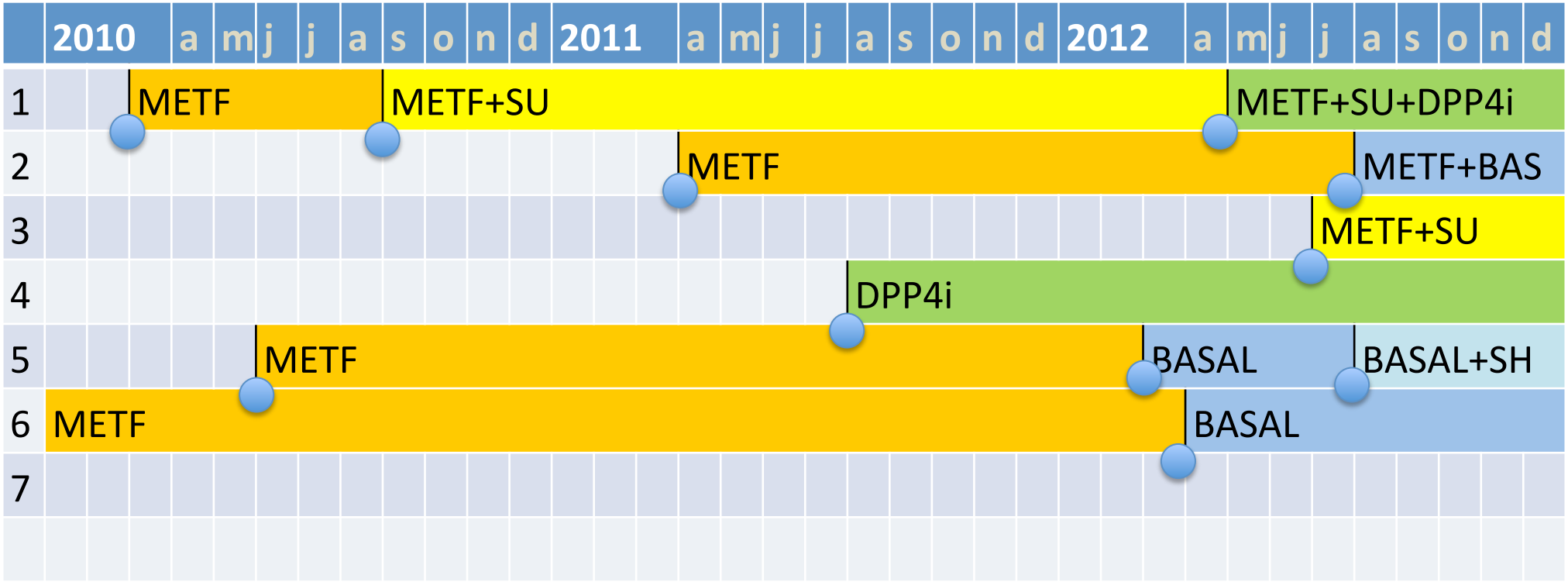
OAD patient flow: November 2011 - October 2012



From millions of records to 1 picture

1. Select patient data that is likely to be complete
2. Add brand and treatment information
3. Transform to 1 row per month per drug
4. Define start and stop of individual drug usage
5. Define which treatment the combination of used drugs belongs to
6. Correct for time lapses
7. Correct for periods in which old treatment is still taken but new treatment is also started (should not be viewed as a separate treatment)
8. Add stop periods in case the period between treatments is too long
9. Make a treatment journey per patient

Examples of individual patient journeys



From millions of records to 1 picture

9. Define per patient the flow characteristics and exceptions:
   - Stopper, start in db, step number, change this year etc.
10. Define starters in the specific therapeutic area
    - N, % initatiated by specialist, % start per brand, % per age category, and optional variables
11. Define switch, add on, drop etc
    - N, % initatiated by specialist, % start per brand, % per age category, and optional variables
12. Correct for unsure period (last 8 months)
13. Extrapolate N to total N in Netherlands

Macro Toolbox

- Independent macro's for each task
  - Each step has a well defined start and end state of the data.
  - All macro variables that are not defined within the macro are defined in the macro statement.
  - No re-programming of similar codes.
    - If similar code is used elsewhere a macro is made of it

- Flexible macro's
  - Use macro variables in case the value/variable is not sure. Give it standard the most likely value.

```
%MACRO MyMacro(Subject=subject,DS=InBase);
```

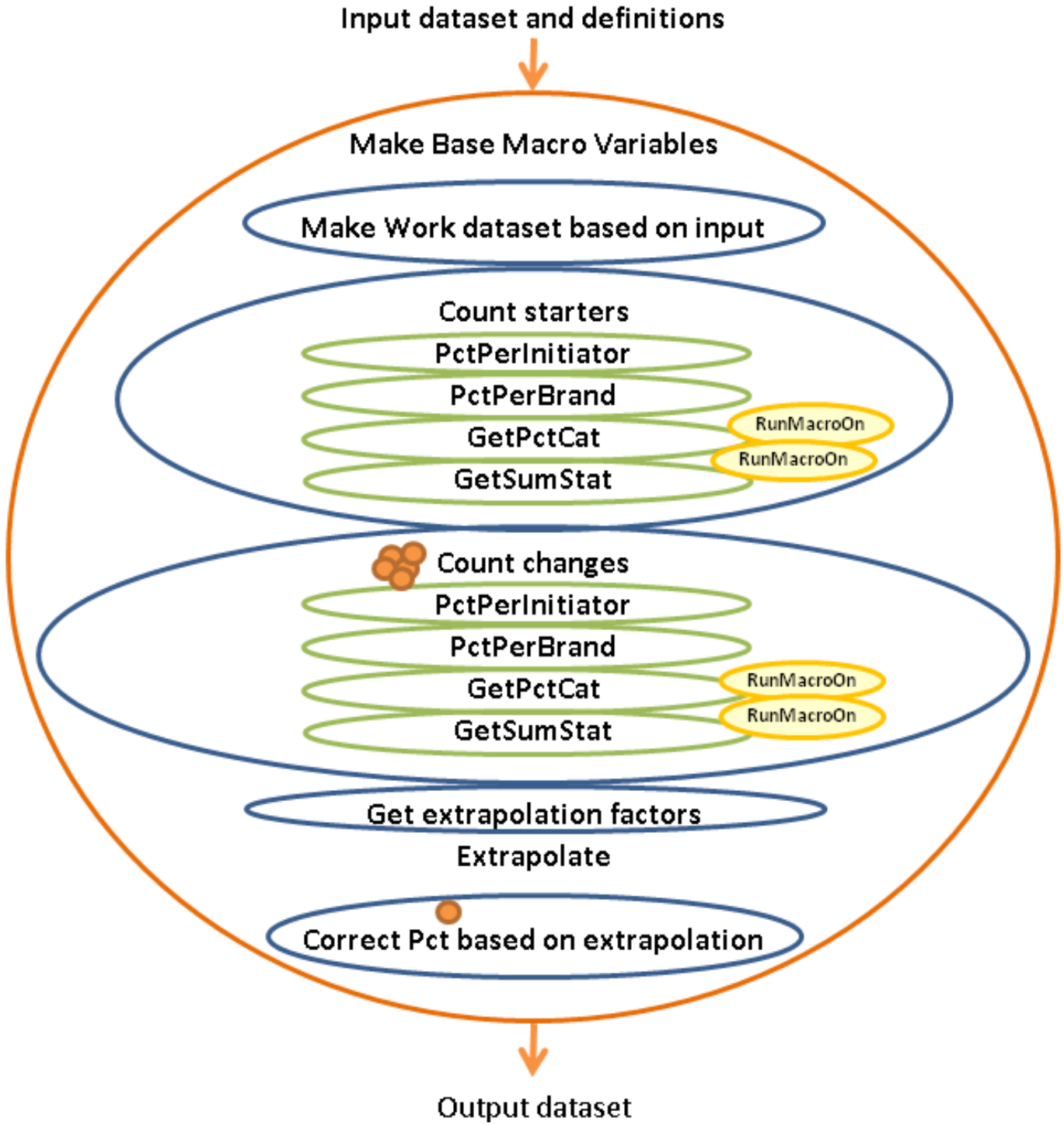
  - If the data can tell itself, we do not define it in the macro input.

Advantages of the macro toolbox

- Keep overview over the process
  - Programming in steps
  - Testing and debugging in steps
  - Possibility to use in between datasets for other tasks
- Different people can work on the same process at the same time
- Much easier to adjust and change after release
- No program repetition of the same actions!!
  - So no errors when changes are made at one place and accidently not at the other place.

Different macro types

- As a step in process
  - Invoked only once (like 1 SAS program)
- Extended function macro
  - Variation in input variables
- Child macro
  - More specific version of an extended function macro
- Small functions
  - Like GetSQLString to add comma's to a by string


Input dataset and definitions
Make Base Macro Variables
Make Work dataset based on input
Count starters
PctPerInitiator
PctPerBrand
GetPctCat
GetSumStat
RunMacroOn
RunMacroOn
Count changes
PctPerInitiator
PctPerBrand
GetPctCat
GetSumStat
RunMacroOn
RunMacroOn
Get extrapolation factors
Extrapolate
Correct Pct based on extrapolation
Output dataset

Tips & Trics

- Add Macro levels to your macro's
  - Level 1 for each level in the process
  - Level 2: invoked from level 1 macro's
  - Level 3: invoked from level 2 macro's
  - Level 4: small help macro's

  - Identify each macro that you start by it's level
  - Add underscores to subdatasets according to it's level
  - Delete work datasets at the end of the macro

```
%IF &debug=N %THEN %DO;
      PROC DATASETS lib=work;
            delete __:;
      RUN; QUIT;
%END;
```

All datasets whose name start with 2 underscores will be deleted.

Tips & Trics

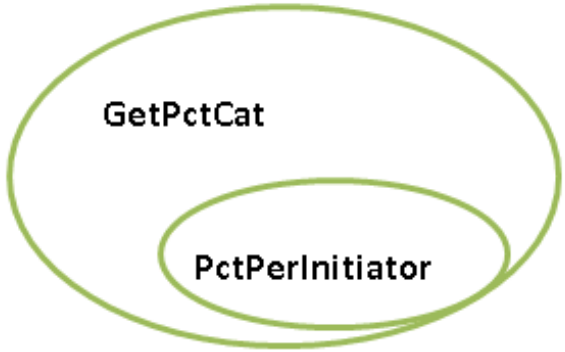
- Start directly with defining a variable as a macro variable when you are doubting that it should be in a later stage.
  - It is less work to make a macro variable silent then to change a fixed macro into a macro variable.

```
%LET subject=subject;
PROC SORT;
        BY &subject;
RUN;
```

```
%MACRO DoMe(subject=subject);
        PROC SORT;
                BY &subject;
        RUN;
%MEND;
```

Tips & trics

- Make child macro's if the mother is used more often for one particular subtask




```
%MACRO PctPerInitiator(inds=,outds=,WhOrder=,byvar=,byvarsql=,debug=Y);
/* Level 3 macro (invocation of other level 3 macro) */
      %GetPctCat(inds=&inds,outds=&outds,WhOrder=&WhOrder,byvar=&byvar,
                  byvarsql=&byvarsql,debug=&debug,var=indi,
                    WhereVar=indi NOT IN (., 99) AND change NE "DROP",prefix=I);
%MEND;
```

Tips & Trics

- A macro to run macro's based on a list of variables:

```
%MACRO RunMacroOn(string=,Costring=,dlm=|,runsubmacro=);
        /* Version 27/05/2013 */
        %LET Mystring=&string;
        %LET MyCo=&Costring;

        %DO %WHILE (%LENGTH(&MyString) >0 );
                /* scan first var in selection */
                %LET runOnMe=%SCAN(&MyString,1,&dlm);
                %LET CoOnMe=%SCAN(&MyCo,1,&dlm);
                /* Remove first var from selection string */
                %IF %INDEX(&mystring,&dlm) = 0 %THEN %LET mystring=;
                        %ELSE %LET mystring=%SUBSTR(&MyString,%INDEX(&mystring,&dlm)+1);
                %IF %INDEX(&myCo,&dlm) = 0 %THEN %LET myCo=;
                        %ELSE %LET myCo=%SUBSTR(&MyCo,%INDEX(&myCo,&dlm)+1);
                /* run macro on selection */
                %UNQUOTE(&runsubmacro);
        %END;
%MEND;

%RunMacroOn(string=&addCat,CoString=&AddPrefix,dlm=|,runsubmacro=%NRSTR(%getpctcat(inds=&inds,
outds=__Out&CoOnMe,WhOrder=order=1,byvar=&boxmed,byvarsql=&boxmed,var=&RunOnMe,WhereVar=,
prefix=&CoOnMe,debug=Y));
```

Tips & Trics

- Divide complex steps into a number of simpler ones
  - A complex process is difficult to debug, difficult to follow and difficult to adapt.
  - Define the action and make a macro of it.
    - (At the end of this step: I should have…)
  - Only focus on the step you are programming

- Make your code as generic as possible
  - If it pops into your mind that a certain data variation is likely and adapting your code to account for that is only taking you a few minutes: adapt your code.

Tips & Trics

- Create 1 final macro invoking all separate steps.
- All global options for variables and actions that are optional should be in it.
- Use predefined values for standard settings.

```
%MACRO PctOfChg(Inds=,
                StartMon=&RepStart,
                boxmed=medi,subject=subject,
                correctforLastM=8,
                MaxStep=3,
                PeriodDuur=12,
                AddCat=, AddPrefix=,
                AddCont=,AddDuurOnStep=N,
                debug=Y);

%PctOfChg(inds=insglp5,debug=N);
```

Questions?