SAS DATA steps

Uses of the DATA step

- Building and modifying data sets
  - Reading data
  - Creating and labeling variables
    - Calculate new variables
    - Recode variables
    - Conditional calculations
    - Variable labels
    - Formatting
  - Subsetting and merging data

Documentation

- Base SAS – SAS 9.2 Language Reference: Concepts – DATA Step Concepts – DATA Step Processing

- Base SAS – SAS 9.2 Language Reference: Dictionary – Dictionary of Language Elements

The simplest DATA step

- Copying a data set (and making it the default)

```
data newcopy;
   set oldcopy;
   run;
```

- Defaults

```
 data datan;
    set _last_;
    run;
```

Creating a new variable

- With an *assignment* statement

```
data new;
   Set y.employee;
   raiseperyear = (salary-salbegin)/(jobtime/12);
   run;

proc means data=new;
   var raiseperyear;
   class gender;
   run;
```

DATA Step Processing

- Think of PROC steps as being applied to *columns* of data
  - Statement order usually does *not* matter
- Think of DATA steps as a sequence of statements applied to *rows* of data
  - Statement order usually *does* matter
  - A DATA step *loops* through the statements as it *sweeps* through the data set

Program Data Vector (PDV)

- At the core of DATA step processing is the PDV, an area of memory that builds and processing the data values for a single observation
  - Compile phase: statements are read and the PDV is defined
  - Execute phase: the loop and the sweep
    - Each loop begins with an empty PDV
  - Implicit `output` and `return`

Boolean computation

```
data boolean;
    set y.employees;
    old = bdate < '01Jan1960'd;
    run;

proc freq;
    tables old*jobcat /chisq;
    run;
```

Random numbers

```
data randomid;
    set y.employees;
    id = ceil(ranuni(-1)*100000); * 5-digit id ;
    run;

proc print;
    var id gender jobcat salary;
    run;
```

Pure Simulation

```
data sim;
   do i=1 to 25;
     x = rannor(-1) + 5); *Normal(5,1) distribution;
     output;
     end;
   run;
```

- `output` no longer at the end
- `do` loops within a *single* PDV

Dates and durations

```
data age;
    set y.employees;
    age = floor((‘21Sep2010’d - bdate)/365.25);
    run;

proc corr data=age;
    var age salary salbegin;
    run;
```

More dates

```
data age2;
   set y.employees;
   today = mdy(9, 21, 2010);
   days = today - bdate;
   years = days/365.25;
   age=floor(years);
   run;

proc plot data=age2;
   plot salary*age; * plot y by x;
   run;
```

Conditional Assignment

- IF-THEN-ELSE

Formats

- Built-in
  - Fixed
  - Comma
  - Dollar
  - Date
- User-written
  - Two steps
    - Create the set of value labels
    - Use them

Fixed formats

- proc freq data=y.employee;
  - tables gender*minority / norow nopercent;
  - format minority f4.2;
  - run;