Программирование на языке SAS

Осень 2016 – ВМК МГУ

Лекция 2. Основы.

Павел Гребенников,
Pavel.Grebennikov@sas.com
Pavel.Grebennikov@statcloud.ru

Библиотеки SAS

«Ссылка» на директорию
В «UNIX»
В WINDOWS

```
libname mylib '/folders/myfolders/data';

libname mylib 'C:\Users\SASU\myfolders';
```

«Ссылка» на схему в СУБД
Например, в Oracle
Например, с Teradata

```
libname cargo oracle schema=airports user=myusr1 password=mypwd1
path="mysrv1";
libname mydblib teradata user=myusr1 pw=mypwd1 schema=otheruser;
```

Временная библиотека WORK (живет до конца сессии)
Обращения WORK.AIR и AIR – одно и тоже

Атрибуты переменных в SAS

1. ТИП
   - Числовой / **numeric** [пропущенные значения: **.** *<точка>*]
   - Строковый / **character** [пропущенные значения: **“ ”** *<пустая строка или строка, состоящая целиком из пробелов>*]

     “Это строка” ‘Это тоже’
2. ДЛИНА – кол-во байт
3. МЕТКА / LABEL – описание переменной
4. ФОРМАТ – правило отображения

```
/* дескриптор для набора данных sashelp.air */
proc contents data=sashelp.air;
run;
```

Дата и время

1. ДАТА, числовой, хранится
   кол-во дней с 1 января 1960 г.
2. ВРЕМЯ, числовой, хранится
   кол-во секунд с полуночи
3. ДАТА-ВРЕМЯ, числовой, хранится
   кол-во секунд с полуночи 1 января 1960 г.
4. ФОРМАТЫ:

```
data dates;
    format
        start_date date9.
        start_time time8.
        start_dt datetime17.
    ;
start_date='01jan2015'd;
start_time='18:30't;
start_dt='01jan201518:30:00'dt
;
run;
```

Дата и время без форматов:

Obs	start_date	start_time	start_dt
1	20089	66600	1735756200

Дата и время с форматами:

Obs	start_date	start_time	start_dt
1	01JAN2015	18:30:00	01JAN15:18:30:00

Шаг DATA

- Обработку данных нужно было делать ещё до того, как был принят стандарт SQL (wiki: в 1986 году первый стандарт языка SQL был принят ANSI, SAS развивается с 1966г, с 1976 как коммерческий продукт).
- Шаг DATA дополняет SQL и наоборот (SQL – работа с множеством наблюдений, шаг DATA – работа с единичными наблюдениями). Чем пользоваться для конкретных целей – нужно думать.
- Шаг DATA очень часто требует существенно меньше ресурсов по сравнению с SQL (важно если данных много), но нужно изучить логику его работы, которая бывает нетривиальной.
- Позволяет создавать и манипулировать наборами данных SAS (включая запись, чтение наборов данных, чтение необработанных данных, изменение структуры, создание агрегатов, объединение таблиц, фильтрацию наблюдений, можно использовать условные переходы, циклы, вызов пользовательских и встроенных функций, …).
- Мы посмотрим малую часть того, что может шаг DATA. Справка по DATA STEP:

  - **SAS(R) 9.4 Language Reference: Concepts, Second Edition -> DATA Step Concepts**
  - **SAS(R) 9.4 Language Reference: Concepts, Second Edition -> Dictionary of Language Elements -> SAS Data Set Options**

Шаг DATA

Самая простая операция –
создание копии набора данных:

```
data work.air; /* новый набор данных */
     set sashelp.air; /* исходный набор данных */
run;
```

1. фаза компиляции
2. фаза выполнения

см. Overview of DATA Step Processing: Flow of Action

Шаг DATA: фаза компиляции

i. проверяется синтаксис

ii. создается:

a. *входной буфер*

когда входной файл не в формате SAS

b. *дескриптор выходного файла*

описание/атрибуты выходного набора и переменных в нем.

c. ***PVD*** *(**p**rogram **d**ata **v**ector)*

область оперативной памяти, куда считываются значения переменных из входного набора или буфера и где происходит обработка. Обрабатывается одно наблюдение за раз.

```
data work.air;
/* будет создан буфер */
   infile 'c:\air.csv' <...>;
   <...>
run;

/* дескриптор для набора
данных sashelp.air */
proc contents data=sashelp.air;
run;
```

DATE	AIR	_ERROR_	_N_
.	.	.	.

см. Overview of DATA Step Processing: Flow of Action

Шаг DATA: фаза выполнения

```
data work.air; /* новый набор данных */
    set sashelp.air; /* исходный набор данных */
run; /* неявный вывод, оператор output; */
```

work.air

DATE	AIR
JAN49	112
FEB49	118
MAR49	132

PDV

DATE	AIR	_ERROR_	_N_
MAR49	132	0	3

sashelp.air

DATE	AIR
JAN49	112
FEB49	118
MAR49	132
APR49	129

Когда этот «цикл» закончится?

см. Overview of DATA Step Processing: Flow of Action

Выбор наблюдений по условию

```
data work.air;
     set sashelp.air;
     where air > 132;
run;
```

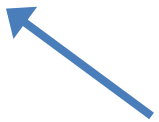
1. Выражение в условии обязано возвратить число:
   0 или . = FALSE
   всё остальное = TRUE
2. Операторы min (**><**) и max (**<>**):
   Например, если A<B, то A><B вернёт значение A
3. `where` умеет работать ***только с теми*** переменными, которые ***есть во входном наборе*** данных.
4. при использовании оператора `where` в ***PDV попадают не все наблюдения***, а только те, которые удовлетворяют условию.

Выбор наблюдений по условию

Symbol	Mnemonic Equivalent
=	EQ
^=	NE
~=	NE
^	NOT
~	NOT

Symbol	Mnemonic Equivalent
>	GT
<	LT
>=	GE
<=	LE
	IN

Symbol	Mnemonic Equivalent
&	AND
|	OR
!	OR
¦	OR



Вхождение в список: **STATE IN ('NY','NJ','PA')**

```
where air > 132 and
(date < '01JAN1965'd
      or
date >= '01JAN1975'd);
```

```
where air gt 132 &
(date lt '01JAN1965'd
      |
date ge '01JAN1975'd);
```

Операторы условного перехода

```
if logic_expr then do;
    …; …; …;
end;
else do;
    …; …; …;
end;
```

```
select <(select-expr)>;
when (when-expr-1) …;
<when (when-expr-2) …; >
<when (when-expr-n) …; >
<otherwise …; >
end;
```

do; …; …; end; - операторные скобки

Действия с переменными на шаге DATA

```
data employees;
     set employee_list;
     if COUNTRY='RU' then bonus=100;
     else bonus=0;
run;
```

PDV содержит:

- все переменные из входного набора
- все переменные, встречающиеся между **`data;`** и **`run;`**

Атрибуты переменных:

- из входного набора
- из data step
- если и там и там, то первое место в коде (***сегодня, чуть позже***)

Как создать новую переменную

1. **Явно:** операторы *length, format*
2. **Неявно**: с помощью присваивания

атрибуты определяются значением, которое присваивается.

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format
4	COUNTRY	Char	2	
5	bonus	Num	8	
6	flag	Char	11	
1	num_children	Num	6	
3	salary	Num	8	BEST12.
2	state	Char	32	

```
data employees;
    length num_chldrn 6 state $32;
    format salary best12.;
    set employee_list;
    if COUNTRY='RU' then bonus=100;
    else bonus=0;
    flag=" processed ";
run;
```

```
NOTE: Variable state is uninitialized.
NOTE: Variable salary is uninitialized.
NOTE: Variable COUNTRY is
uninitialized.
```

Как создать новую переменную

3. А если справа от оператора присваивания стоит **функция**? Тогда тип переменной и её длина зависят от функции (***см. help***). Например:

« …

Details

In a DATA step, if the UPCASE function returns a value to a variable that has not previously been assigned a length, then that variable is given the length of the argument

… »

ФУНКЦИИ

Функции – это подпрограммы, которые возвращают одно значение

function-name (*argument-1*<, ...*argument-n*>)

function-name (***OF*** *шаблон-названия*)

Применяться на шаге DATA и PROC (например, в условиях-фильтрах)

Примеры:

SUM(*argument*,*argument*,...) – сумма аргументов (пропущенные значения игнор.). Sum и оператор «+» могут работать по разному.

UPCASE(*argument*) – перевод символьной строки в верхний регистр

Функции можно применять друг к другу без создания промежуточных переменных

```
data temp;
   set in_tab;
   a1=b+c;
   a2=sum(b,c);
run;
```

in_tab

B (Num)	C (Num)
.	1

** Справка: SAS(R) 9.4 Functions and CALL Routines: Reference*

Функции преобразования типов

PUT(*source*, *format*.) :

число в строку с использованием формата

INPUT(*source*,*informat*.) :

строка в число с использованием формата

```
data test;
   num=input("123.45", best12.);
   str=put(num, best6.);
   len_str=length(str);

   format date date9.;
   date=input("21JUN2015", date9.);
   dt_str=put(date, ddmmyy10.);
   len_dt=length(dt_str);
run;
```

Функции для даты и времени

DATE	Текущая дата в формате SAS
DATEPART	Извлекает дату в формате SAS из даты/времени
DATETIME	Текущая дата/время (timestamp)
DAY	Число из даты во внутреннем формате: day ('01jan2013'd) -> 1 ; day (19359) -> 1
DHMS	Вычисляет дату/время (datetime) из даты, часов, минут, секунд.
HMS	Вычисляет время в формате SAS из часов, минут и секунд
HOUR	Возвращает часы из внутреннего формата времени или даты/времени
INTCK	Возвращает количество временных промежутков, укладывающихся на данный интервал (годы, месяцы, ...)
INTNX	Увеличивает дату на данную величину, и возвращает полученное значение в формате SAS
MDY	Сформировать дату во внутреннем формате из месяца, числа, года: Mdy(1,1,2013) -> 19359
MINUTE	Извлечь минуты из внутреннего формата времени или даты/времени
MONTH	Извлечь месяц из внутреннего формата даты
QTR	Вычисляет квартал (года) из значения даты SAS.
TIME	Возвращает текущее время в формате SAS
TIMEPART	Извлекает часть, содержащую время, из временной метки
TODAY	Возвращает текущую дата во внутреннем формате даты
WEEK	Номер недели из внутреннего формата даты
WEEKDAY	День недели из внутреннего формата даты
YEAR	Год из внутреннего формата даты

Вывод наблюдений в наборы данных

```
data male female;
     set employee_list;
     if GENDER='M' then output male;
     else if GENDER='F' then output female;
     else output male female;
run;
```

Циклы DO

Вывести в набор данных целые цифры от 1 до 5 (включительно)

1.
```
data test;
n=0;
   do until(n>=5);
      n+1;
         output;
   end;
run;
```

По умолчанию новая переменная инициализируется пропущенным значением

В данном случае – то же, что n=n+1;

Условие проверяется в конце цикла

2.
```
data test;
n=0;
   do while(n<5);
      n+1;
         output;
   end;
run;
```

Условие проверяется в начале цикла

3.
```
data test;
   do n=1 to 5;
         output;
   end;
run;
```

А если хочется, можно вообще без циклов (см. оператор go to):

```
data test;
n=1;
  lbl1:
  output;
  n=n+1;
  if n<=5 then go to lbl1;
run;
```

Массивы (шаг DATA)

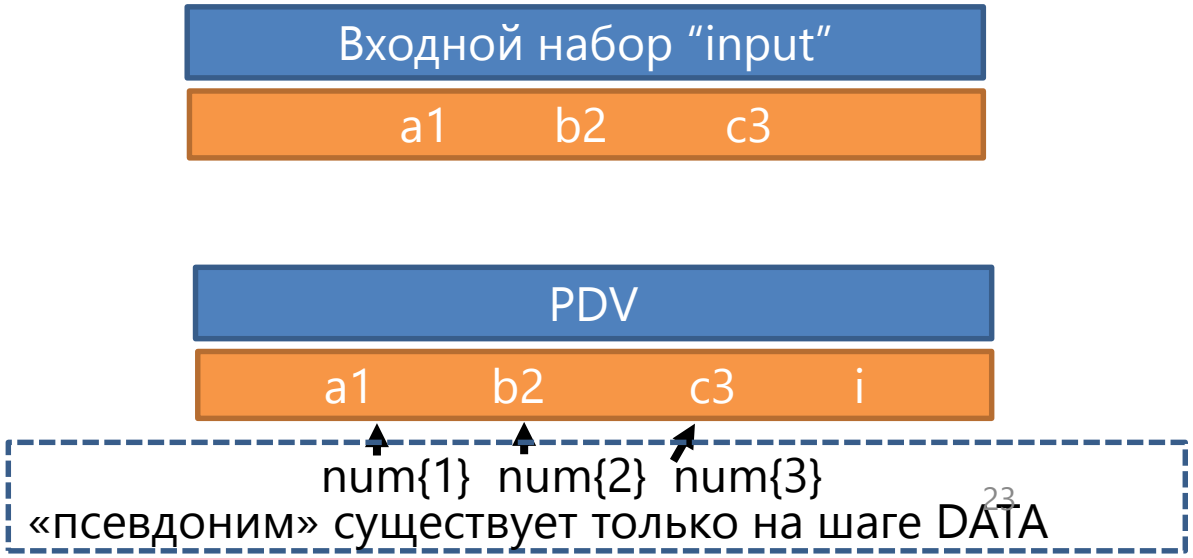
- Удобный способ автоматизировать обработку множества (однотипных) переменных.
- Также можно использовать для создания новых переменных.

Пример:

Есть исходный набор данных с числовыми переменными a1, b2, c3. Нужно к этим переменным прибавить 1.

Создадим «псевдоним с целочисленным индексом» для этих переменных, и используем цикл для повторяющихся действий.

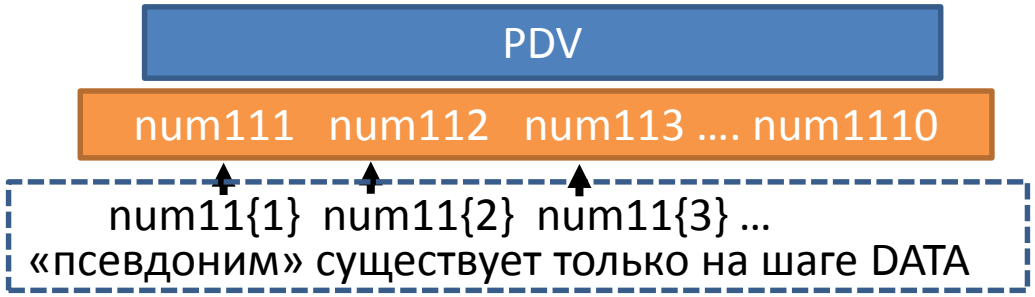
```
data temp1;
  array num {*} a1 b2 c3;
  set input;
  do i=1 to 3;
    num{i}=num{i}+1;
  end;
run;
```



Массивы (шаг DATA)

Создание массива из новых переменных

```
data test;
   array num11{10};
   num11{1}=34;
   num112=12;
run;
```



Число элементов массива,
элементы нумеруются с единицы

Начало названия новой переменной – как у массива, а затем идет число. К переменной можно обращаться:

- по названию, которое создалось автоматически, и
- по псевдониму – названию массива.

	num111	num112	num113	num114	num115
1	34	12	.	.	

Массивы (шаг DATA)

Массивы из символьных переменных.

```
data test;
    array char{6} $ 20;
run;
```

Массив из 6 переменных символьного типа, которые имеют размер 20 байт

```
data test1;
    array char{*} $ 20 var22-var33;
run;
```

Размер массива может быть опущен (заменён символом «*»), если он ясен «из контекста», то есть в определении можно понять, для скольких переменных нужно создать «псевдоним». В названии переменных для хранения данных, на которые «ссылается» массив, может быть применён шаблон «-», который понимает целое число в конце названия переменной как счетчик.

	var22	var23	var24	var25	var26	var27	var28	var29	var30	var31	var32	var33
1												

Переменные могут быть взяты из набора данных, или созданы на шаге DATA

```
data test2;
   array char{*} $ 20 aa bb cc;
   set input;
run;
```

Массивы (шаг DATA)

Врѐменные массивы

Массив можно использовать не только для обращения к переменным в PDV и их создания, но и как средство для выделения области памяти, которая не относится к PDV.

```
data test1;
    array char{10} $ 20 _temporary_;
run;
```

Особенности:

1) Обращение к элементам массива – только через название массива и индекс элемента.
2) В PDV не создаётся никаких переменных для хранения данных.

```
data test1;
    array char{10} $ 20 _temporary_;
    put _all_;
run;
```

```
_ERROR_=0 _N_=1
NOTE: The data set WORK.TEST1 has 1 observations and 0 variables.
NOTE: DATA statement used (Total process time):
      real time          0.00 seconds
      cpu time           0.01 seconds
```

Массивы: особенности

1) Память для массива выделяется на фазе компиляции шага DATA и освобождается при его завершении
2) На шаге DATA нет никаких инструментов для динамического выделения памяти (“malloc”) и для её освобождения (“free”), то есть память нельзя выделить позже, чем начнётся шаг DATA, и освободить раньше, чем он завершится.
3) Размер массива не может быть динамическим. Он должен быть известен на момент компиляции шага DATA.

Массивы: ошибки

1) выход за границы массива

```
62    data test1;
63       array char{10} $ 20 _temporary_;
64       char{30}="ggg";
65    run;

ERROR: Array subscript out of range at line 64 column 4.
_ERROR_=1 _N_=1
NOTE: The SAS System stopped processing this step because of errors.
WARNING: The data set WORK.TEST1 may be incomplete.  When this step was stopped
         there were 0 observations and 0 variables.
WARNING: Data set WORK.TEST1 was not replaced because this step was stopped.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

2) смешивание типов

```
66    data test1;
67       array char{2} aa bb;
68       aa=10;
69       bb="ggg";
70    run;

NOTE: Character values have been converted to numeric
      values at the places given by: (Line):(Column).
      69:7
NOTE: Invalid numeric data, 'ggg' , at line 69 column 7.
aa=10 bb=. _ERROR_=1 _N_=1
NOTE: The data set WORK.TEST1 has 1 observations and 2 variables.
NOTE: DATA statement used (Total process time):
      real time           0.05 seconds
      cpu time            0.00 seconds
```

Переменная bb числового типа была создана с помощью оператора array.