Chapter 5
Writing Loops in the DATA Step

Arthur Li

Implicit and Explicit Loops
Implicit Loops

- The DATA step works like a loop – an implicit loop
- It repetitively executes statements
    - reads data values
    - creates observations in the PDV one at a time
- Each loop is called an iteration
- Suppose you have the following dataset that contains patient IDs for a clinical trial

Patient:

	ID
1	M2390
2	F2390
3	F2340
4	M1240

- You would like to assign each patient with either a drug or a placebo (50% chance of either/or)

Implicit Loops

❖ The **RANUNI** function

RANUNI (SEED)

❑ It generates a number ~ Uniform(0, 1)
e.g. 0.13567, 0.34567, 0.56789, etc

❑ SEED is a nonnegative integer

❑ The RANUNI function generates a stream of numbers based on SEED

❑ When SEED is set to 0, the generated number cannot be reproduced

❑ when SEED is a non-zero number, the generated number can be produced

Implicit Loops

```
data ex5_1 (drop=rannum);
    set patient;
    rannum = ranuni(2);
    if rannum > 0.5 then group = 'D';
    else group = 'P';
run;
```

Patient:

	ID
1	M2390
2	F2390
3	F2340
4	M1240

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0		.	

1st iteration:

- _N_ ← 1
- _ERROR_ ← 0
- The rest of variables are set to *missing*

Implicit Loops

```
data ex5_1 (drop=rannum);
    set patient;
    rannum = ranuni(2);
    if rannum > 0.5 then group = 'D';
    else group = 'P';
run;
```

Patient:

	ID
1	M2390
2	F2390
3	F2340
4	M1240

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	M2390	.	

1st iteration:

- The SET statement copies the 1st observation → PDV

Implicit Loops

```
data ex5_1 (drop=rannum);
    set patient;
    rannum = ranuni(2);
    if rannum > 0.5 then group = 'D';
    else group = 'P';
run;
```

Patient:

	ID
1	M2390
2	F2390
3	F2340
4	M1240

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	M2390	0.36993	

1st iteration:

❖ RANNUM is generated

Implicit Loops

```
data ex5_1 (drop=rannum);
    set patient;
    rannum = ranuni(2);
    if rannum > 0.5 then group = 'D';
    else group = 'P';
run;
```

Patient:

	ID
1	M2390
2	F2390
3	F2340
4	M1240

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	M2390	0.36993	P

1st iteration:

❖ GROUP ← 'P' since RANNUM is *not* > 0.5

Implicit Loops

```
data ex5_1 (drop=rannum);
    set patient;
    rannum = ranuni(2);
    if rannum > 0.5 then group = 'D';
    else group = 'P';
run;
```

Patient:

	ID
1	M2390
2	F2390
3	F2340
4	M1240

Ex5_1:

	ID	GROUP
1	M2390	P

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	M2390	0.36993	P

1st iteration:

- The implicit OUTPUT statement writes the variables marked with (K) to the final dataset
- SAS returns to the beginning of the DATA step

Implicit Loops

```
data ex5_1 (drop=rannum);
     set patient;
     rannum = ranuni(2);
     if rannum > 0.5 then group = 'D';
     else group = 'P';
run;
```

Patient:

	ID
1	M2390
2	F2390
3	F2340
4	M1240

Ex5_1:

	ID	GROUP
1	M2390	P

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
2	0	M2390	.	

2nd iteration:

❖_N_ ↑2

❖ID is retained since ID is from input dataset

❖GROUP and RANNUM are set to *missing*

Implicit Loops

```
data ex5_1 (drop=rannum);
    set patient;
    rannum = ranuni(2);
    if rannum > 0.5 then group = 'D';
    else group = 'P';
run;
```

Patient:

	ID
1	M2390
2	F2390
3	F2340
4	M1240

Ex5_1:

	ID	GROUP
1	M2390	P

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
2	0	M2390	.	

2nd iteration:

❖The SET statement copies the 2nd observation → PDV

Skip a few iterations….

Implicit Loops

```
data ex5_1 (drop=rannum);
    set patient;
    rannum = ranuni(2);
    if rannum > 0.5 then group = 'D';
    else group = 'P';
run;
```

Patient:

	ID
1	M2390
2	F2390
3	F2340
4	M1240

Ex5_1:

	ID	GROUP
1	M2390	P
2	F2390	D
3	F2340	D
4	M1240	D

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
4	0	M1240	0.51880	D

The end of 4th iteration:

❖The implicit OUTPUT statement writes the variables marked with K to final dataset

❖SAS returns to the beginning of the DATA step

Implicit Loops

```
data ex5_1 (drop=rannum);
     set patient;
     rannum = ranuni(2);
     if rannum > 0.5 then group = 'D';
     else group = 'P';
run;
```

Patient:

	ID
1	M2390
2	F2390
3	F2340
4	M1240

Ex5_1:

	ID	GROUP
1	M2390	P
2	F2390	D
3	F2340	D
4	M1240	D

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
5	0	M1240	.	

5th iteration:

❖_N_ ↑5

❖ID is retained

❖GROUP and RANNUM are set to *missing*

Implicit Loops

```
data ex5_1 (drop=rannum);
     set patient;
     rannum = ranuni(2);
     if rannum > 0.5 then group = 'D';
     else group = 'P';
run;
```

End-of-file marker

Patient:

	ID
1	M2390
2	F2390
3	F2340
4	M1240

Ex5_1:

	ID	GROUP
1	M2390	P
2	F2390	D
3	F2340	D
4	M1240	D

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
5	0	M1240	.	

5th iteration:

- SAS reaches the end-of-file-marker, which means that there are no more observations to read
- The execution phase is completed, goes to next DATA/PROC step

Explicit Loops

- Suppose you don’t have a dataset containing the patient IDs

- You are asked to assign four patients, ‘M2390’, ‘F2390’, ‘F2340’, ‘M1240’, with a 50% chance of receiving either the drug or the placebo

- You can create the ID and assign each ID to a group in the DATA step at the same time. For example

Explicit Loops

Assigning IDs in the DATA step

```
data ex5_2(drop = rannum);
    id = 'M2390';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'F2390';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'F2340';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'M1240';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;
run;
```

Explicit Loops

4 explicit OUTPUT statements

```
data ex5_2(drop = rannum);
    id = 'M2390';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'F2390';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'F2340';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'M1240';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;
run;
```

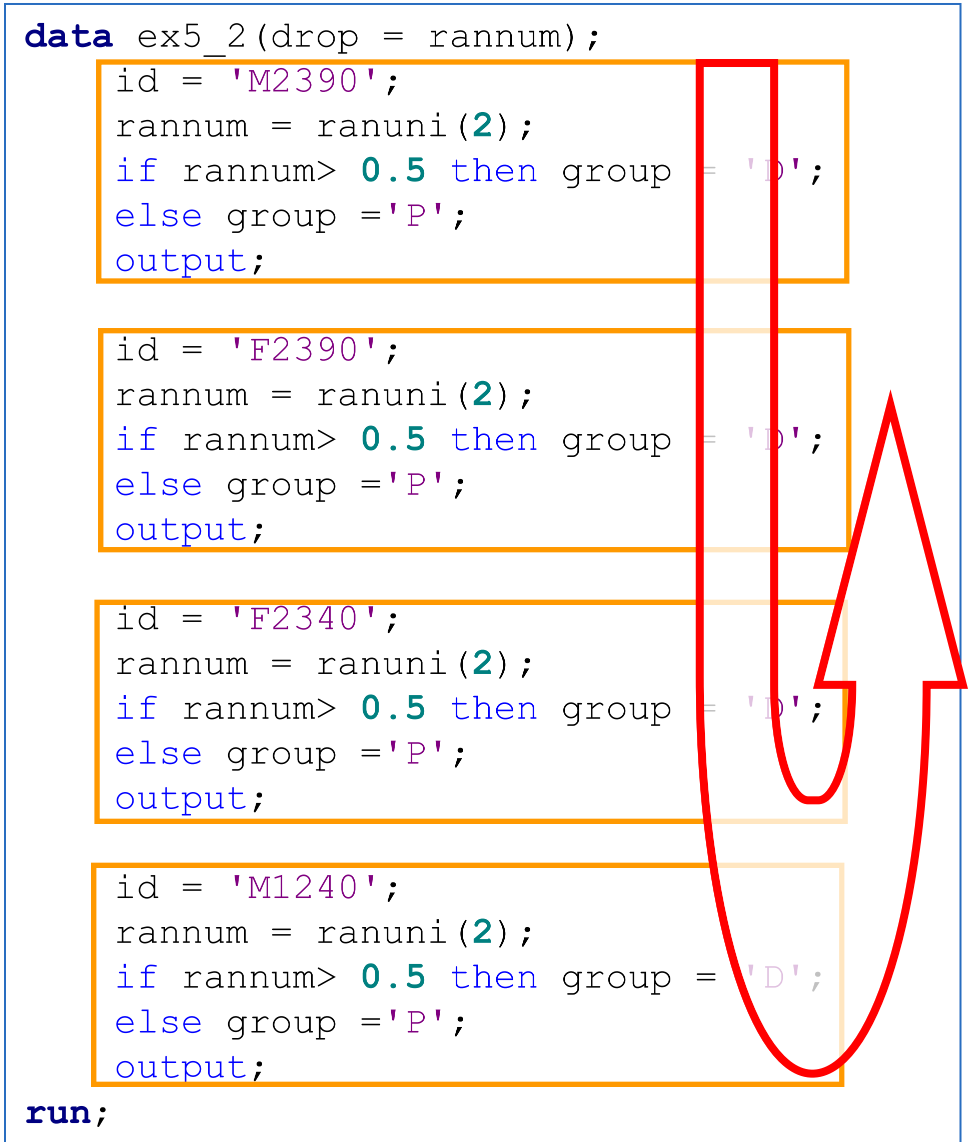
Explicit Loops

4 almost identical blocks

```
data ex5_2(drop = rannum);
    id = 'M2390';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'F2390';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'F2340';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'M1240';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;
run;
```

Explicit Loops

4 almost identical blocks



```
data ex5_2(drop = rannum);
    id = 'M2390';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'F2390';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'F2340';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'M1240';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;
run;
```

- Put identical codes in a loop
- Loop along the IDs
- Reduce amount of coding

Explicit Loops

- General form for an iterative DO loop:

```
DO  index-variable = value1, value2, …, valuen;
     SAS statements
END;
```

  - INDEX-VARIABLE: contains the value of the current iteration
  - The loop will execute along VALUE1 through VALUEN
  - The VALUES can be either character or numeric

Explicit Loops

DO index-variable = value1, value2, …, valuen;
    SAS statements
END;

```
data ex5_2(drop = rannum);
    id = 'M2390';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'F2390';
    ...

    id = 'F2340';
    ...

    id = 'M1240';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;
run;
```

❖INDEX-VARIABLE: **ID**

Explicit Loops

DO index-variable = value1, value2, …, valuen;
SAS statements
END;

```
data ex5_2(drop = rannum);
    id = 'M2390';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'F2390';
    ...

    id = 'F2340';
    ...

    id = 'M1240';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;
run;
```

- INDEX-VARIABLE: **ID**
- VALUE1 – VALUEN: **'M2390’, 'F2390’, 'F2340’, 'M1240'**

Explicit Loops

DO index-variable = value1, value2, …, valuen;
    SAS statements
END;

```
data ex5_2(drop = rannum);
    id = 'M2390';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'F2390';
    ...

    id = 'F2340';
    ...

    id = 'M1240';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;
run;
```

❖INDEX-VARIABLE: **ID**

❖VALUE1 – VALUEN:
'M2390’, 'F2390’, 'F2340’, 'M1240

❖SAS STATEMENTS:

```
rannum = ranuni(2);
if rannum> 0.5 then group = 'D';
else group ='P';
output;
```

Explicit Loops

DO index-variable = value1, value2, …, valuen;
    SAS statements
END;

```
data ex5_2(drop = rannum);
    id = 'M2390';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;

    id = 'F2390';
    ...

    id = 'F2340';
    ...

    id = 'M1240';
    rannum = ranuni(2);
    if rannum> 0.5 then group = 'D';
    else group ='P';
    output;
run;
```

```
data ex5_3(drop = rannum);
    do id = 'M2390', 'F2390',
            'F2340', 'M1240';
       rannum = ranuni(2);
       if rannum> 0.5 then group = 'D';
       else group ='P';
       output;
    end;
run;
```

Explicit Loops

- Usually we use the iterative DO loop and loop along a sequence of integers

```
DO index-variable = start TO stop <BY increment>;
    SAS statements
END;
```

- The loop will execute from the START to the STOP value

Explicit Loops

- Usually we use the iterative DO loop and loop along a sequence of integers

```
DO index-variable = start TO stop <BY increment>;
    SAS statements
END;
```

- The optional BY clause specifies an increment between START and END
- The default value for INCREMENT is 1

Explicit Loops

- Usually we use the iterative DO loop and loop along a sequence of integers

```
DO index-variable = start TO stop <BY increment>;
   SAS statements
END;
```

- START, STOP, and INCREMENT
  - ❑ Numbers
  - ❑ Variables
  - ❑ SAS expressions
- These values are set upon entry into the DO loop and cannot be modified during the processing of the DO loop

Explicit Loops

- Usually we use the iterative DO loop and loop along a sequence of integers

```
DO index-variable = start TO stop <BY increment>;
    SAS statements
END;
```

- INDEX-VARIABLE can be changed within the loop

Explicit Loops

DO index-variable = start TO stop <BY increment>;
SAS statements
END;

- Suppose you are using a sequence of numbers, say 1 to 4, as patient IDs

```
data ex5_4(drop = rannum);
    do id = 1 to 4;
        rannum = ranuni(2);
        if rannum> 0.5 then group = 'D';
        else group ='P';
        output;
    end;
run;
```

- INDEX-VARIABLE: ID
- START: 1
- STOP: 4
- INCREMENT: 1

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	.	.	

- ❖Since we didn’t read an input dataset, there will be only one iteration for the DATA step
- ❖_N_ will be 1 for the entire execution phase

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	1	.	

1st Iteration of DO loop:

- ID ← 1

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	1	0.36993	

1st Iteration of DO loop:

- RANNUM is generated

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	1	0.36993	P

1st Iteration of DO loop:

❖GROUP ← 'P' since RANNUM is *not* > 0.5

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P

PDV:

_N_	D	_ERROR_	D	ID	K	RANNUM	D	GROUP	K
1		0		1		0.36993		P	

1st Iteration of DO loop:

- ❖ The OUTPUT statement instructs SAS to write observations to the output dataset

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	1	0.36993	P

1st Iteration of DO loop:

- SAS reaches the end of DO loop

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	2	0.36993	P

2nd Iteration of DO loop:

❖ID ↑ 2; since 2 ≤ 4, the 2nd iteration continues

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	2	0.94018	P

2nd Iteration of DO loop:

❖RANNUM is generated

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P

PDV:

_N_	D	_ERROR_	D	ID	K	RANNUM	D	GROUP	K
1		0		2		0.94018		D	

2nd Iteration of DO loop:

❖GROUP ← 'D' since RANNUM > 0.5

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P
2	2	D

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	2	0.94018	D

2nd Iteration of DO loop:

- The OUTPUT statement instructs SAS to write observations to the output dataset

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
 ➡    output;
   end;
run;
```

	ID	GROUP
1	1	P
2	2	D

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	2	0.94018	D

- ❖Let’s skip two iterations

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P
2	2	D
3	3	D
4	4	D

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	4	0.51880	D

4th Iteration of DO loop:

- SAS reaches the end of the DO loop of the 4th iteration

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P
2	2	D
3	3	D
4	4	D

PDV:

_N_	D	_ERROR_	D	ID	K	RANNUM	D	GROUP	K
1		0		5		0.51880		D	

5th iteration of DO loop:

- ❖ID ↑5; since 5 is > 4, the loop ends

Explicit Loops

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group = 'D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P
2	2	D
3	3	D
4	4	D

PDV:

_N_	D	_ERROR_	D	ID	K	RANNUM	D	GROUP	K
1		0		5		0.51880		D	

- ❖There will be no implicit OUTPUT statement
- ❖Since we didn’t read an input dataset, the DATA step execution ends

Explicit Loops

- Using an iterative DO loop requires specifying the *number* of iterations for the DO loop.

- Sometimes you will need to execute statements repetitively until a condition is met.

- In this situation, you need to use either the DO WHILE or DO UNTIL statements.

Explicit Loops

```
DO WHILE (expression);
SAS statements
END;
```

- EXPRESSION is evaluated at the top of the DO loop
- The DO loop will not execute if the EXPRESSION is false

Explicit Loops

DO WHILE (expression);
SAS statements
END;

Iterative DO loop:

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

DO WHILE loop:

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

PDV:

_N_	D	_ERROR_	D	ID	K	RANNUM	D	GROUP	K
1		0		0		.			

At the beginning of the execution phase:

❖_N_ ← 1, _ERROR_ ← 0

❖ID ← 0 because of the SUM statement

❖The rest of the variables are set to *missing*

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	0	.	

1st iteration of the DO WHILE loop:

- Since ID < 4, loop continues

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
  ➡   id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

PDV:

_N_	D	_ERROR_	D	ID	K	RANNUM	D	GROUP	K
1		0		1		.			

1st iteration of the DO WHILE loop:

- ID ← 1

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	1	0.36993	

1st iteration of the DO WHILE loop:

- RANNUM is generated

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	1	0.36993	P

1st iteration of the DO WHILE loop:

- GROUP ← ‘P’

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
  ➔   output;
   end;
run;
```

	ID	GROUP
1	1	P

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	1	0.36993	P

1st iteration of the DO WHILE loop:

- The OUTPUT statement instructs SAS to write observations to the output dataset

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
➡  end;
run;
```

	ID	GROUP
1	1	P

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	1	0.36993	P

1st iteration of the DO WHILE loop:

- SAS reaches the end of DO loop

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	1	0.36993	P

2nd iteration of the DO WHILE loop:

- ❖Since ID < 4, the loop continues

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
  ➡   id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P

PDV:

_N_	D	_ERROR_	D	ID	K	RANNUM	D	GROUP	K
1		0		2		0.36993		P	

2nd iteration of the DO WHILE loop:

❖ID ← 2

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	2	0.36993	P

- Let’s skip a few iterations

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P
2	2	D
3	3	D
4	4	D

PDV:

_N_	D	_ERROR_	D	ID	K	RANNUM	D	GROUP	K
1		0		4		0.51880		D	

At the end of the 4th iteration:

❖Here’s the contents of the PDV at the end of the 4th loop

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P
2	2	D
3	3	D
4	4	D

PDV:

_N_	D	_ERROR_	D	ID	K	RANNUM	D	GROUP	K
1		0		4		0.51880		D	

5th iteration:

- ❖Now ID is not < 4, loop stops

Explicit Loops

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

	ID	GROUP
1	1	P
2	2	D
3	3	D
4	4	D

PDV:

_N_ D	_ERROR_ D	ID K	RANNUM D	GROUP K
1	0	4	0.51880	D

5th iteration:

- The execution phase ends

Explicit Loops

```
DO UNTIL (expression);
SAS statements
END;
```

- Unlike DO WHILE loops, the DO UNTIL loop evaluates the condition at the **end** of the loop
- The DO UNTIL loop will not continue for another iteration if the EXPRESSION is evaluated to be **TRUE** at the end of the current loop
- That means the DO UNTIL loop always executes at least once

Explicit Loops

DO UNTIL (expression);
SAS statements
END;

Iterative DO loop:

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

DO WHILE loop:

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

DO UNTIL loop:

```
data ex5_6(drop=rannum);
   do until (id >=4);
      id +1;
      rannum = ranuni(2);
      if rannum > 0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

Explicit Loops

DO UNTIL (expression);
SAS statements
END;

Iterative DO loop:

```
data ex5_4(drop = rannum);
   do id = 1 to 4;
      rannum = ranuni(2);
      if rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

Will not continue if the EXPRESSION is false

DO WHILE loop:

```
data ex5_5(drop=rannum);
   do while (id <4);
      id + 1;
      rannum = ranuni(2);
      if  rannum>0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

Will not continue for another iteration if the EXPRESSION is true

DO UNTIL loop:

```
data ex5_6(drop=rannum);
   do until (id >=4);
      id +1;
      rannum = ranuni(2);
      if rannum > 0.5 then group='D';
      else group ='P';
      output;
   end;
run;
```

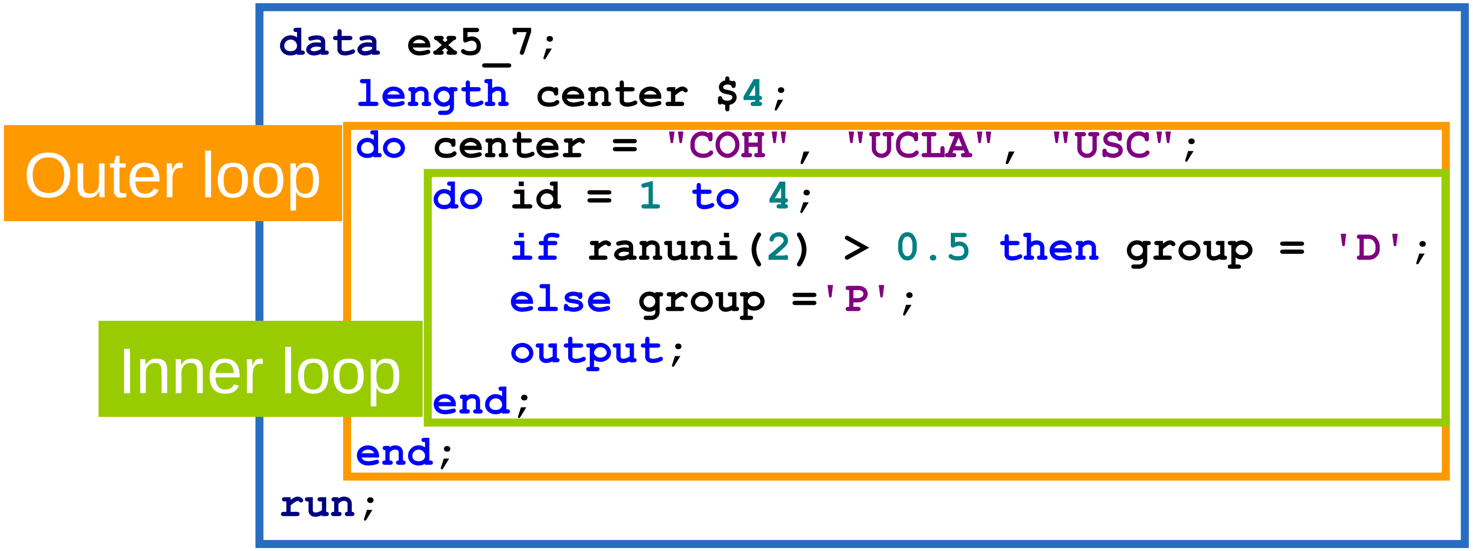
Nested Loops

- Suppose that you would like to assign 12 patients with either a drug or a placebo
- These 12 subjects are from 3 cancer centers (“COH”, “UCLA”, and “USC”) with 4 subjects per center

```
data ex5_7;
   length center $4;
   do center = "COH", "UCLA", "USC";
      do id = 1 to 4;
         if ranuni(2) > 0.5 then group = 'D';
         else group ='P';
         output;
      end;
   end;
run;
```

Nested Loops

- Suppose that you would like to assign 12 patients with either a drug or a placebo
- These 12 subjects are from 3 cancer centers (“COH”, “UCLA”, and “USC”) with 4 subjects per center



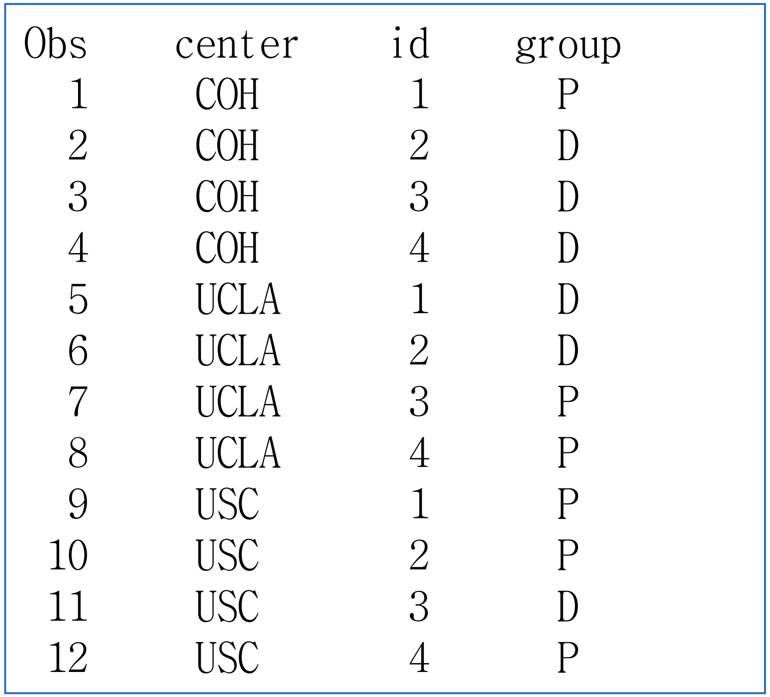
Outer loop

Inner loop

```
data ex5_7;
   length center $4;
   do center = "COH", "UCLA", "USC";
      do id = 1 to 4;
         if ranuni(2) > 0.5 then group = 'D';
         else group ='P';
         output;
      end;
   end;
run;
```

Nested Loops

- Suppose that you would like to assign 12 patients with either a drug or a placebo
- These 12 subjects are from 3 cancer centers (“COH”, “UCLA”, and “USC”) with 4 subjects per center



```
Obs    center    id    group
  1     COH       1      P
  2     COH       2      D
  3     COH       3      D
  4     COH       4      D
  5     UCLA      1      D
  6     UCLA      2      D
  7     UCLA      3      P
  8     UCLA      4      P
  9     USC       1      P
 10     USC       2      P
 11     USC       3      D
 12     USC       4      P
```

Combining Implicit and Explicit Loops

- In previous program all the observations were created from one DATA step since we didn't read any input data
- Suppose the values for CENTER is stored in a SAS dataset
- For each center, you need to assign 4 patients with either a drug or a placebo

	CENTER
1	COH
2	UCLA
3	USC

```
data trial7;
    set cancer_center;
    do id = 1 to 4;
        if  ranuni(2)> 0.5 then group = 'D';
        else group ='P';
        output;
    end;
run;
```

Combining Implicit and Explicit Loops

- In previous program all the observations were created from one DATA step since we didn't read any input data
- Suppose the values for CENTER is stored in a SAS dataset
- For each center, you need to assign 4 patients with either a drug or a placebo

	CENTER
1	COH
2	UCLA
3	USC

DATA step: implicit loop

```
data trial7;
    set cancer_center;
    do id = 1 to 4;
        if  ranuni(2)> 0.5 then group = 'D';
        else group ='P';
        output;
    end;
run;
```

explicit loop