

Fairy-tale of procedure: Proc Sort

Soujanya Konda

Disclaimer!

All opinions expressed in this presentation are the presenter's personal views, and do not reflect the views or opinions of IQVIA or its subsidiary companies' policies and position.

Some images taken here are freely available from the Internet for a diagrammatic representation of the content.

What is Proc Sort

Proc Sort is a procedure which helps to orders the data by the values of one or more character or numeric variables. Sort creates a new dataset or either replaces the original dataset.

Syntax:

```
proc sort data=<name of data>;  
    by <name of variables>;  
run;
```

sashelp.class.sas7bdat dataset is used an example for most of the illustrations in this presentation.

Sorting in SAS:

```
proc sort data=class;  
    by age sex;  
run;
```

SORTSIZE

These are the few options which increases performance and decrease run time.



```
44
45
46 proc sort data=dm out=dm1 sortsize=5gb;
47   by usubjid;
48 run;
```

```
NOTE: There were 257431 observations read from the data set WORK.DM.
NOTE: The data set WORK.DM1 has 257431 observations and 325 variables.
NOTE: Compressing data set WORK.DM1 decreased size by 93.80 percent.
      Compressed is 1996 pages; un-compressed would require 32180 pages.
NOTE: PROCEDURE SORT used (Total process time):
      real time           11.62 seconds
      cpu time            17.23 seconds
```

```
49
50
51 proc sort data=dm out=dm2 sortsize=250mb;
52   by usubjid;
53 run;
```

```
NOTE: There were 257431 observations read from the data set WORK.DM.
NOTE: The data set WORK.DM2 has 257431 observations and 325 variables.
NOTE: Compressing data set WORK.DM2 decreased size by 93.80 percent.
      Compressed is 1996 pages; un-compressed would require 32180 pages.
NOTE: PROCEDURE SORT used (Total process time):
      real time           9.72 seconds
      cpu time            14.83 seconds
```

TAGSORT

Dataset is sorted by specified tags (variables specified in BY Statement).

TAGSORT

Without TAGSORT

```
298 proc sort data=class out=class_1_ ;
299     by usubjid age;
300 run;
```

NOTE: There were 320926 observations read from the data set WORK.CLASS.
NOTE: The data set WORK.CLASS_1_ has 320926 observations and 107 variables.
NOTE: Compressing data set WORK.CLASS_1_ decreased size by 91.47 percent.
Compressed is 1441 pages; un-compressed would require 16891 pages.

NOTE: PROCEDURE SORT used (Total process time):	
real time	7.02 seconds
cpu time	10.46 seconds

With TAGSORT

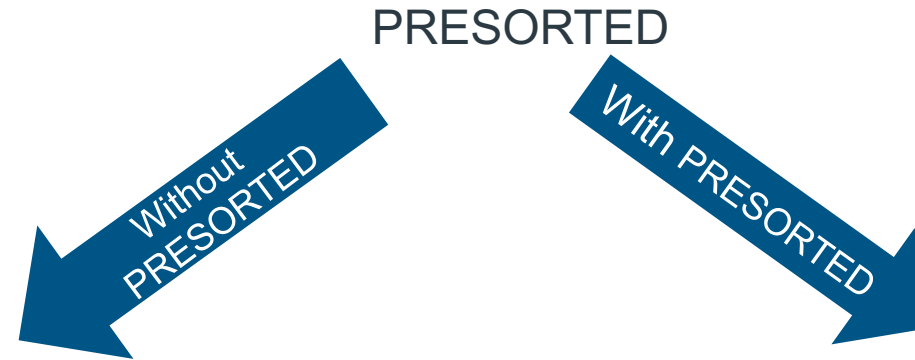
```
295 proc sort data=class out=class_1 tagsort;
296     by usubjid age;
297 run;
```

NOTE: Tag sort reads each observation of the input data set twice.
NOTE: The data set WORK.CLASS_1 has 320926 observations and 107 variables.
NOTE: Compressing data set WORK.CLASS_1 decreased size by 91.47 percent.
Compressed is 1441 pages; un-compressed would require 16891 pages.

NOTE: PROCEDURE SORT used (Total process time):	
real time	3.53 seconds
cpu time	3.55 seconds

PRESORTED

Is a redundant sort - helps to perform a sequential check and ensures input data is properly sorted.



```
304 proc sort data=class out=classw ;
305     by usubjid age;
306 run;
```

NOTE: There were 320926 observations read from the data set WORK.CLASS.
NOTE: The data set WORK.CLASSW has 320926 observations and 107 variables.
NOTE: Compressing data set WORK.CLASSW decreased size by 91.47 percent.
Compressed is 1441 pages; un-compressed would require 16891 pages.
NOTE: PROCEDURE SORT used (Total process time):
real time 6.82 seconds
cpu time 9.95 seconds

```
301 proc sort data=class out=classy presorted ;
302     by usubjid age;
303 run;
```

NOTE: Sort order of input data set has been verified.
NOTE: Input data set is already sorted; it has been copied to the output data set.
NOTE: There were 320926 observations read from the data set WORK.CLASS.
NOTE: The data set WORK.CLASSY has 320926 observations and 107 variables.
NOTE: Compressing data set WORK.CLASSY decreased size by 91.47 percent.
Compressed is 1441 pages; un-compressed would require 16891 pages.
NOTE: PROCEDURE SORT used (Total process time):
real time 3.40 seconds
cpu time 3.41 seconds

Reverse Sort Order

SYNTAX:

```
proc sort data=class;  
  by descending age;  
run;
```

NAME	SEX	AGE	HEIGHT	WEIGHT
Joyce	F	11	51.3	50.5
Thomas	M	11	57.5	85
James	M	12	57.3	83
Jane	F	12	59.8	84.5
John	M	12	59	99.5
Louise	F	12	56.3	77
Robert	M	12	64.8	128
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Jeffrey	M	13	62.5	84
Alfred	M	14	69	112.5
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5
Judy	F	14	64.3	90
Janet	F	15	62.5	112.5
Mary	F	15	66.5	112
Ronald	M	15	67	133
William	M	15	66.5	112
Philip	M	16	72	150



NAME	SEX	AGE	HEIGHT	WEIGHT
Philip	M	16	72	150
Janet	F	15	62.5	112.5
Mary	F	15	66.5	112
Ronald	M	15	67	133
William	M	15	66.5	112
Alfred	M	14	69	112.5
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5
Judy	F	14	64.3	90
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Jeffrey	M	13	62.5	84
James	M	12	57.3	83
Jane	F	12	59.8	84.5
John	M	12	59	99.5
Louise	F	12	56.3	77
Robert	M	12	64.8	128
Joyce	F	11	51.3	50.5
Thomas	M	11	57.5	85

Ascending order of sorting is a default sort order

Sort with different options like format, label, keep, drop and etc

```
proc format;  
  value $SEX  'F'='Female'  
              'M'='Male';  
  
run;  
  
proc sort data=class  
  out=class2 (keep=name age sex  
              height weight  
              rename=(name=subject));  
  label WEIGHT='Weight at Baseline';  
  format sex $sex.;  
  by age;  
  where age > 12;  
run;  
  
proc print data=class2 label; run;
```

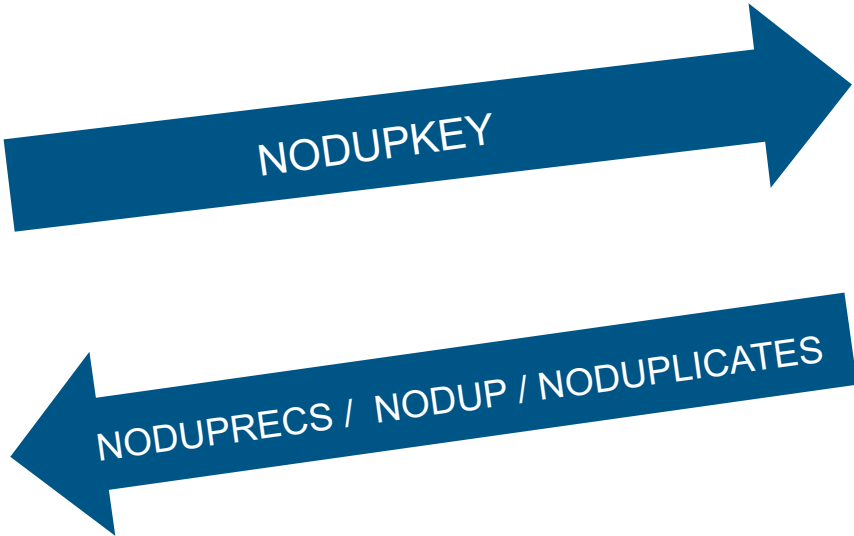
Obs	SUBJECT	SEX	AGE	HEIGHT	Weight at Baseline
1	Alice	Female	13	56.5	84.0
2	Barbara	Female	13	65.3	98.0
3	Jeffrey	Male	13	62.5	84.0
4	Alfred	Male	14	69.0	112.5
5	Carol	Female	14	62.8	102.5
6	Henry	Male	14	63.5	102.5
7	Judy	Female	14	64.3	90.0
8	Janet	Female	15	62.5	112.5
9	Mary	Female	15	66.5	112.0
10	Ronald	Male	15	67.0	133.0
11	William	Male	15	66.5	112.0
12	Philip	Male	16	72.0	150.0

“Where” statement and “Where” option gives the same result.

NODUP and NODUPKEY Options with Proc Sort

```
proc sort data=class
out=class2 noduprec
by age;
run;
```

Obs	NAME	SEX	AGE	HEIGHT	WEIGHT
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84.0
10	John	M	12	59.0	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90.0



```
proc sort data=class
out=class2 nodupkey
by age;
run;
```

Obs	NAME	SEX	AGE	HEIGHT	WEIGHT
1	Joyce	F	11	51.3	50.5
2	James	M	12	57.3	83.0
3	Alice	F	13	56.5	84.0
4	Alfred	M	14	69.0	112.5
5	Janet	F	15	62.5	112.5
6	Philip	M	16	72.0	150.0

NOUNIQUEKEY

Obs	NAME	SEX	AGE	HEIGHT	WEIGHT
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84.0
10	John	M	12	59.0	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90.0
13	Louise	F	12	56.3	77.0
14	Mary	F	15	66.5	112.0
15	Philip	M	16	72.0	150.0
16	Robert	M	12	64.8	128.0
17	Ronald	M	15	67.0	133.0
18	Thomas	M	11	57.5	85.0
19	William	M	15	66.5	112.0
20	Alfred	M	14	69.0	112.5
21	Alice	F	13	56.5	84.0
22	Barbara	F	13	65.3	98.0
23	Carol	F	14	62.8	102.5
24	Henry	M	14	63.5	102.5
25	James	M	12	57.3	83.0

```
proc sort data=class out=class2
  NOUNIQUEKEY uniqueout=obs_deleted;
  by name;
run;
```

Non Unique Observations

Unique Observations

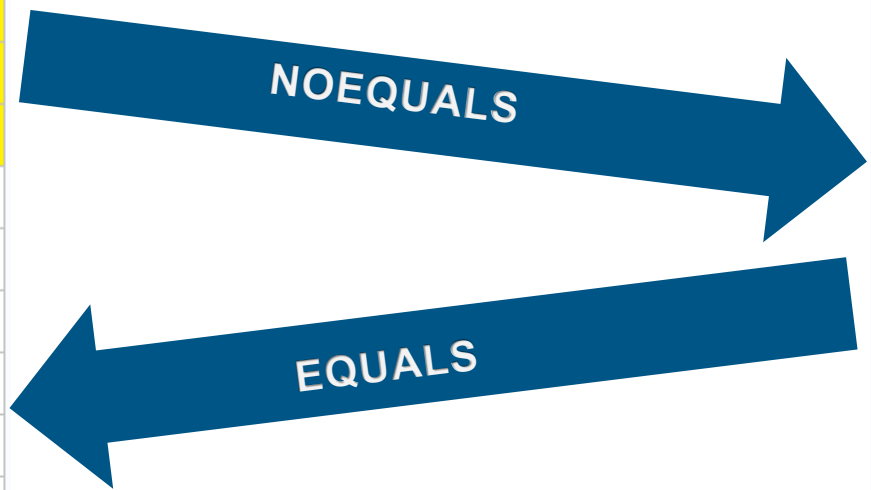
Obs	NAME	SEX	AGE	HEIGHT	WEIGHT
1	Alfred	M	14	69.0	112.5
2	Alfred	M	14	69.0	112.5
3	Alice	F	13	56.5	84.0
4	Alice	F	13	56.5	84.0
5	Barbara	F	13	65.3	98.0
6	Barbara	F	13	65.3	98.0
7	Carol	F	14	62.8	102.5
8	Carol	F	14	62.8	102.5
9	Henry	M	14	63.5	102.5
10	Henry	M	14	63.5	102.5
11	James	M	12	57.3	83.0
12	James	M	12	57.3	83.0

Obs	NAME	SEX	AGE	HEIGHT	WEIGHT
1	Jane	F	12	59.8	84.5
2	Janet	F	15	62.5	112.5
3	Jeffrey	M	13	62.5	84.0
4	John	M	12	59.0	99.5
5	Joyce	F	11	51.3	50.5
6	Judy	F	14	64.3	90.0
7	Louise	F	12	56.3	77.0
8	Mary	F	15	66.5	112.0
9	Philip	M	16	72.0	150.0
10	Robert	M	12	64.8	128.0
11	Ronald	M	15	67.0	133.0
12	Thomas	M	11	57.5	85.0
13	William	M	15	66.5	112.0

EQUALS || NOEQUALS Options with Proc Sort

```
proc sort data=class
  out=byyears noequals;
  by name;
run;
```

Obs	NAME	SEX	AGE	HEIGHT	WEIGHT
1	Alfred	M	14	69.0	70.0
2	Alfred	M	14	69.0	112.5
3	Alice	F	13	56.5	70.0
4	Alice	F	13	56.5	84.0
5	Barbara	F	13	65.3	98.0
6	Barbara	F	13	65.3	50.0
7	Carol	F	14	62.8	70.0
8	Carol	F	14	62.8	102.5
9	Henry	M	14	63.5	170.0
10	Henry	M	14	63.5	102.5
11	James	M	12	57.3	90.0
12	James	M	12	57.3	83.0



Obs	NAME	SEX	AGE	HEIGHT	WEIGHT
1	Alfred	M	14	69.0	112.5
2	Alfred	M	14	69.0	70.0
3	Alice	F	13	56.5	84.0
4	Alice	F	13	56.5	70.0
5	Barbara	F	13	65.3	98.0
6	Barbara	F	13	65.3	50.0
7	Carol	F	14	62.8	102.5
8	Carol	F	14	62.8	70.0
9	Henry	M	14	63.5	102.5
10	Henry	M	14	63.5	170.0
11	James	M	12	57.3	83.0
12	James	M	12	57.3	90.0

Sort order of resultant dataset differs in NOEQUALS, where as in EQUALS it resembles the input dataset

Collating sequence

SAS users no longer work exclusively with English-language data.

- ASCII
- DANISH
- EBCDIC
- FINNISH
- NATIONAL
- NORWEGIAN
- REVERSE
- SWEDISH

INPUT DATA

Obs	AETERM
1	Fever
2	Pain
3	pain
4	fever
5	Swelling
6	swell
7	Rash
8	Skin RASH

```
proc sort data=ae out = ae2  
          SORTSEQ =LINGUISTIC ;  
  by aeterm;  
run;
```

```
proc sort data=ae out = ae2  
  SORTSEQ =LINGUISTIC  
  (case_first=upper) ;  
  by aeterm;  
run;
```

Obs	AETERM
1	fever
2	Fever
3	pain
4	Pain
5	Rash
6	Skin RASH
7	swell
8	Swelling

Obs	AETERM
1	Fever
2	fever
3	Pain
4	pain
5	Rash
6	Skin RASH
7	swell
8	Swelling

NUMERIC_COLLATION

INPUT DATA

Obs	VISIT
1	Day 22
2	Day 3
3	Day 2
4	Day 1
5	Day 11

```
proc sort data=visits out = vis;  
  by visit;  
run;
```

```
proc sort data=visits out = vis1 SORTSEQ  
=LINGUISTIC (NUMERIC_COLLATION=ON);  
  by visit;  
run;
```

Obs	VISIT
1	Day 1
2	Day 11
3	Day 2
4	Day 22
5	Day 3

Obs	VISIT
1	Day 1
2	Day 2
3	Day 3
4	Day 11
5	Day 22

ALTERNATE_HANDLING

INPUT DATA

Obs	INV
1	INVESTIGATOR1
2	INVESTIGATOR3
3	INVESTIGATOR11
4	INVESTIGATOR2
5	INVESTIGATOR33
6	INVESTIGATOR22

```
proc sort data=invest out=inv1
  SORTSEQ =LINGUISTIC
  (ALTERNATE_HANDLING=SHIFTED) ;
  by inv;
run;
```

Obs	INV
1	INVESTIGATOR1
2	INVESTIGATOR11
3	INVESTIGATOR2
4	INVESTIGATOR22
5	INVESTIGATOR3
6	INVESTIGATOR33

ALTERNATE_HANDLING with STRENGTH

INPUT DATA

Obs	INV
1	StANN
2	STALLEN
3	STAL
4	STELLER
5	Ac STELLER
6	AcSTELLER

```
proc sort data=invest out=inv1  
  SORTSEQ =LINGUISTIC (STRENGTH=3  
  CASE_FIRST=UPPER);  
  by inv;  
run;
```

Obs	INV
1	Ac STELLER
2	AcSTELLER
3	STAL
4	STALLEN
5	StANN
6	STELLER

REFERENCES

<https://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a002197022.htm>

<https://documentation.sas.com/?docsetId=proc&docsetVersion=9.4&docsetTarget=n1i7kdzswf8t78n1j9fdk6qcqo2k.htm&locale=en>



Thank You

Name: Soujanya Konda

Organization: IQVIA

E-mail: soujanya.konda@quintiles.com

Web: <https://www.iqvvia.com>

Twitter: https://twitter.com/IQVIA_global