

# Efficiency: Improving the Performance of Data Manipulation

Zhichang Jiang

Alberta Health Services  
[zhichang.jiang@albertahealthservices.ca](mailto:zhichang.jiang@albertahealthservices.ca)

# What Is Efficiency

---

- CPU time
- Disk space
- I/O
- Technical support
- Accomplishment

# Steps of Data Manipulation

---

- Reading data
- Exploring data
- Manipulating data
- Storing data

# Reading Raw Data

---

- Read only the needed variables

```
INPUT id $1-7 name $251-260;
```

- When subsetting, read key variable first

```
INPUT id $1-7  
      type $8-15 .....;  
IF id = '1234567';
```

```
INPUT id $1-7 @;  
IF id = '1234567';  
INPUT type $8-15 .....;
```

- Choose faster forms of INPUT statement

```
INPUT @1 postalcode $6.;
```

```
INPUT @1 postalcode $CHAR6.;
```

# Reading SAS Data Sets

---

- Read only the needed variables with KEEP option  
 DATA one (KEEP=id name);  
 SET two (KEEP=id name);
- IF vs WHERE

Time Saving Using WHERE Against IF		
Obs Selected	WIN NT	UNIX
20%	21%	33%
30%	11%	26%
50%	-4%	11%
80%	-15%	-12%

# Exploring Data

---

- Use `_NULL_` data set for report
- Use `Formats` to check for invalid values
- Use `Informats` to check for invalid values
- Use `PROC COMPARE` to compare two data sets

# Avoid DATA Step

---

- Copy with PROC COPY  
Data NEW.address; set OLD.address;  
PROC COPY IN=old OUT=new; SELECT address;
- Append with PROC APPEND  
DATA large; SET large small;  
PROC APPEND DATA=small BASE=large;
- Modify data set structure with PROC DATASETS

## Modify Data Set Structure with PROC DATASETS

---

```
PROC DATASETS LIBRARY=work;  
  CHANGE oldname=newname;  
  MODIFY newname;  
  RENAME lastname=lname;  
  LABEL fname='First Name';
```

# Avoid Sorting Data

---

- TABLE option in PROC FREQ

- CLASS option in PROC MEANS, PROC SUMMARY

```
PROC MEANS DATA=pop;  
BY year;  
VAR population;
```

```
PROC MEANS DATA=pop;  
CLASS year;  
VAR population;
```

- NOTSORTED option;

```
PROC PRINT DATA=sales;  
BY store NOTSORTED;  
VAR amout;
```

# Speeding Up Sorting

---

- The SORTSIZE=MAX option

- KEEP and DROP options

```
PROC SORT DATA=pop;
```

```
PROC SORT DATA=pop (KEEP= year population) OUT=pop1;
```

- Sort in ascending order

# Calling Functions

---

- Replacing a function call in general
- Replacing SUBSTR calls  
pc3=SUBSTR(postalcode, 1, 3)  
LENGTH pc3 \$ 3; pc3 = postalcode;
- Replacing PUT and INPUT calls  
charvar = PUT(value, 3.);  
LENGTH charvar \$ 3; charvar=value;

# Miscellaneous Techniques

---

- Checking for missing values  
total = missing + a + b;    total = a + b + missing;
- Computing: e.g. multiplying, not dividing;  
b = a / 2;    b = a \* 0.5;
- Checking for division by zero;  
b = b / a;    IF a THEN b = b / a;
- Reduce the number of comparisons  
IF a = 1 THEN b = a;    IF a = 1 THEN b = a;  
IF a = 2 THEN b = a \* 2;    ELSE IF a = 2 THEN b = a \* 2;  
IF a = 3 THEN b = a \* 3;    ELSE IF a = 3 THEN b = a \* 3;
- Perform all data manipulation in a single DATA step

# Storage Space

---

- Controlling the lengths of variables  
`str=SCAN(string, 2);    LENGTH str $ 10; str=SCAN(string, 2);`
- Delete old data sets, or reuse data set names
- Assign a length to shorten numeric variable
- Store a view instead of a data;  
`DATA pop / VIEW = pop;`  
`CREATE VIEW pop AS .....`

# Low Maintenance Techniques

---

- Comments
- Programming style
- Programming techniques
- MACRO

# Questions ?