SAS Basic Concepts #1
The language, log, output, SAS data set and others

JC Wang
Outline

1. The Language
   - Language Elements
   - SAS Log
   - SAS Libraries

2. SAS Data Sets
   - Two Portions of a SAS Data Set

3. SAS Output
   - SAS Output: An Example

4. System Options

5. SAS Comments

6. Report Writing—Round One
Language Elements

- Consists of statements
- Statements usually start with keywords (e.g., DATA, PROC, INPUT, KEEP, IF, DO, ...)
- Each statement ends with a semicolon (;)
- Words in statements are case insensitive, except for most situations when texts are enclosed in matching quotation marks, single quotes (’) or double quotes ("")
Language Elements

- Consists of statements
- Statements usually start with keywords (e.g., DATA, PROC, INPUT, KEEP, IF, DO, ...)
- Each statement ends with a semicolon (;)
- Words in statements are case insensitive, except for most situations when texts are enclosed in matching quotation marks, single quotes (’) or double quotes ("")
Language Elements

- Consists of statements
- Statements usually start with keywords (e.g., DATA, PROC, INPUT, KEEP, IF, DO, ...)
  - Each statement ends with a semicolon (;)
  - Words in statements are case insensitive, except for most situations when texts are enclosed in matching quotation marks, single quotes (’) or double quotes ("")
Language Elements

- Consists of statements
- Statements usually start with keywords (e.g., DATA, PROC, INPUT, KEEP, IF, DO, ...)
- Each statement ends with a semicolon (;)
- Words in statements are case insensitive, except for most situations when texts are enclosed in matching quotation marks, single quotes (’) or double quotes (")
Language Elements

- Consists of statements
- Statements usually start with keywords (egs: DATA, PROC, INPUT, KEEP, IF, DO, ...)
- Each statement ends with a semicolon (;)
- Words in statements are case insensitive, except for most situations when texts are enclosed in matching quotation marks, single quotes (’) or double quotes (")
Program Layout

SAS statements are in free format,

- they can begin/end anywhere on a line
- a statement can continue over several lines
- several statements can be on a line (avoid this)
Program Layout

SAS statements are in free format,
- they can begin/end anywhere on a line
- a statement can continue over several lines
- several statements can be on a line (avoid this)
Program Layout

SAS statements are in free format,

- they can begin/end anywhere on a line
- a statement can continue over several lines
- several statements can be on a line (avoid this)
Program Layout

SAS statements are in free format,

- they can begin/end anywhere on a line
- a statement can continue over several lines
- several statements can be on a line (avoid this)
Basic Building Blocks of a SAS Program

There are two basic building blocks:

- **Data steps**: read raw data, create/modify data, set/change data value, combine/merge/update data sets
- **Proc steps**: pre-written procedures that perform various tasks

Note: In addition to above, there are stand-alone statements such as OPTIONS, TITLE, FOOTNOTE, LIBNAME, FILENAME, ...
Basic Building Blocks of a SAS Program

There are two basic building blocks:

- **Data steps**: read raw data, create/modify data, set/change data value, combine/merge/update data sets
- **Proc steps**: pre-written procedures that perform various tasks

Note: In addition to above, there are stand-alone statements such as OPTIONS, TITLE, FOOTNOTE, LIBNAME, FILENAME, ...
Basic Building Blocks of a SAS Program

There are two basic building blocks:

- **Data steps**: read raw data, create/modify data, set/change data value, combine/merge/update data sets
- **Proc steps**: pre-written procedures that perform various tasks

Note: In addition to above, there are stand-alone statements such as OPTIONS, TITLE, FOOTNOTE, LIBNAME, FILENAME, ...
Basic Building Blocks of a SAS Program

There are two basic building blocks:

- **Data steps**: read raw data, create/modify data, set/change data value, combine/merge/update data sets
- **Proc steps**: pre-written procedures that perform various tasks

Note: In addition to above, there are stand-alone statements such as OPTIONS, TITLE, FOOTNOTE, LIBNAME, FILENAME, ...
Step Boundaries

- DATA & PROC statements signal the beginning of a new step.
- Subsequent DATA or PROC statements, RUN statements (for DATA steps and most PROC steps) or a QUIT statement (for interactive PROC steps such as PROC DATASETS, PROC GLM, PROC REG, PROC SQL, PROC IML) signal the end of a step.
- Upon encountering a new step, SAS stops reading statements and executes the previous step.
Step Boundaries

- DATA & PROC statements signal the beginning of a new step
- Subsequent DATA or PROC statements, RUN statements (for DATA steps and most PROC steps) or a QUIT statement (for interactive PROC steps such as PROC DATASETS, PROC GLM, PROC REG, PROC SQL, PROC IML) signal the end of a step
- Upon encountering new step, SAS stops reading statements and executes the previous step
Step Boundaries

- DATA & PROC statements signal the beginning of a new step
- Subsequent DATA or PROC statements, RUN statements (for DATA steps and most PROC steps) or a QUIT statement (for interactive PROC steps such as PROC DATASETS, PROC GLM, PROC REG, PROC SQL, PROC IML) signal the end of a step
- Upon encountering new step, SAS stops reading statements and executes the previous step
Step Boundaries

- DATA & PROC statements signal the beginning of a new step
- Subsequent DATA or PROC statements, RUN statements (for DATA steps and most PROC steps) or a QUIT statement (for interactive PROC steps such as PROC DATASETS, PROC GLM, PROC REG, PROC SQL, PROC IML) signal the end of a step
- Upon encountering new step, SAS stops reading statements and executes the previous step
upon submitting a program, SAS generates a log of processing activities

**SAS Log Example**

See eg01.sas (run 001.sas first, note that the SAS data path need to be specified accordingly)
A SAS library

is a collection of SAS files, including SAS data sets

Two types of libraries

- temporary: WORK
- permanent: all others, default is SASUSER
A SAS library
is a collection of SAS files, including SAS data sets

Two types of libraries
- temporary: WORK
- permanent: all others, default is SASUSER
A SAS library
is a collection of SAS files, including SAS data sets

Two types of libraries
- temporary: **WORK**
- permanent: all others, default is **SASUSER**
Referencing a SAS data set

\textit{libref.filenname}

- \textit{libref} gives the name of library; \textit{filename} gives the name of SAS file
- Either name consists of alphabets (lower or upper case), digits (0 to 9), and underscore (\_), and must not start with digits
- \textit{libref} can be 1 to 8 characters long
- \textit{filename} (a SAS name) can be 1 to 32 characters long (for versions 8 or later, 8 characters long for order version)
Referencing a SAS data set

\textit{libref.filenname}

- \textit{libref} gives the name of library; \textit{filenname} gives the name of SAS file
- either name consists of alphabets (lower or upper case), digits (0 to 9), and underscore (\_) and must not start with digits
- \textit{libref} can be 1 to 8 characters long
- \textit{filenname} (a SAS name) can be 1 to 32 characters long (for versions 8 or later, 8 characters long for order version)
Referencing a SAS data set

`libref.filename`

- `libref` gives the name of library; `filename` gives the name of SAS file
- either name consists of alphabets (lower or upper case), digits (0 to 9), and underscore (_) and must not start with digits
- `libref` can be 1 to 8 characters long
- `filename` (a SAS name) can be 1 to 32 characters long (for versions 8 or later, 8 characters long for order version)
Referencing a SAS data set

`libref.filename`

- `libref` gives the name of library; `filename` gives the name of SAS file
- either name consists of alphabets (lower or upper case), digits (0 to 9), and underscore (_) and must not start with digits
- `libref` can be 1 to 8 characters long
- `filename` (a SAS name) can be 1 to 32 characters long (for versions 8 or later, 8 characters long for order version)
Referencing a SAS data set

`libref.filename`

- `libref` gives the name of library; `filename` gives the name of SAS file
- either name consists of alphabets (lower or upper case), digits (0 to 9), and underscore (_) and must not start with digits
- `libref` can be 1 to 8 characters long
- `filename` (a SAS name) can be 1 to 32 characters long (for versions 8 or later, 8 characters long for order version)
Assigning a libref
an example

LIBNAME *clinic* 'f:\Stat 6800\sasdata';
Two portions of a SAS data set

Each SAS data set has two portions

- descriptor portion: contains data set attributes (name, date/time created, number of observations, number of variables, engine (version), observation length) and variable attributes (names, labels, types (numeric or character), lengths, informats, formats)
- data portion: data values in a rectangular table
  - rows = observations
  - columns = variables

Note: may contain index
Two portions of a SAS data set

Each SAS data set has two portions

- descriptor portion: contains data set attributes (name, date/time created, number of observations, number of variables, engine (version), observation length) and variable attributes (names, labels, types (numeric or character), lengths, informats, formats)
- data portion: data values in a rectangular table
  - rows = observations
  - columns = variables

Note: may contain index
Two portions of a SAS data set

Each SAS data set has two portions

- **descriptor portion:** contains data set attributes (name, date/time created, number of observations, number of variables, engine (version), observation length) and variable attributes (names, labels, types (numeric or character), lengths, informats, formats)

- **data portion:** data values in a rectangular table
  - rows = observations
  - columns = variables

Note: may contain index
Two portions of a SAS data set

Each SAS data set has two portions

- descriptor portion: contains data set attributes (name, date/time created, number of observations, number of variables, engine (version), observation length) and variable attributes (names, labels, types (numeric or character), lengths, informats, formats)
- data portion: data values in a rectangular table
  - rows = observations
  - columns = variables

Note: may contain index
Two portions of a SAS data set

Each SAS data set has two portions

- descriptor portion: contains data set attributes (name, date/time created, number of observations, number of variables, engine (version), observation length) and variable attributes (names, labels, types (numeric or character), lengths, informats, formats)
- data portion: data values in a rectangular table
  - rows = observations
  - columns = variables

Note: may contain index
Two portions of a SAS data set

Each SAS data set has two portions

- descriptor portion: contains data set attributes (name, date/time created, number of observations, number of variables, engine (version), observation length) and variable attributes (names, labels, types (numeric or character), lengths, informats, formats)

- data portion: data values in a rectangular table
  - rows = observations
  - columns = variables

Note: may contain index
Two portions of a SAS data set

Each SAS data set has two portions

- descriptor portion: contains data set attributes (name, date/time created, number of observations, number of variables, engine (version), observation length) and variable attributes (names, labels, types (numeric or character), lengths, informats, formats)
- data portion: data values in a rectangular table
  - rows = observations
  - columns = variables

Note: may contain index
Viewing Descriptor Portion of SAS Data Sets

Can use one of the following:

- PROC CONTENTS
- PROC DATASETS with CONTENTS statement
Viewing Descriptor Portion of SAS Data Sets

Can use one of the following:

- PROC CONTENTS
- PROC DATASETS with CONTENTS statement
Viewing Descriptor Portion of SAS Data Sets

Can use one of the following:

- PROC CONTENTS
- PROC DATASETS with CONTENTS statement
Outline

1 The Language
   • Language Elements
   • SAS Log
   • SAS Libraries

2 SAS Data Sets
   • Two Portions of a SAS Data Set

3 SAS Output
   • SAS Output: An Example

4 System Options

5 SAS Comments

6 Report Writing—Round One
Listing Output and HTML Output

See eg01.sas

- use of TITLE and FOOTNOTE statements
- SAS formats and FORMAT statements
Listing Output and HTML Output

See eg01.sas

- use of TITLE and FOOTNOTE statements
- SAS formats and FORMAT statements
Listing Output and HTML Output

See eg01.sas

- use of TITLE and FOOTNOTE statements
- SAS formats and FORMAT statements
Outline

1. The Language
   - Language Elements
   - SAS Log
   - SAS Libraries

2. SAS Data Sets
   - Two Portions of a SAS Data Set

3. SAS Output
   - SAS Output: An Example

4. System Options

5. SAS Comments

6. Report Writing—Round One
Setting System Options

**OPTIONS options;**

- can be anywhere in the program
- can be repeated
- many *options* are of on/off modes, eg.
  \- NUMBER|NONUMBER, DATE|NODATE, CENTER|NOCENTER
- others are in the form of *option=value*, eg.
  \- PAGENO=1, LINESIZE=80 (or LS=80), PAGESIZE=40 (or PS=40), YEARCUTOFF=1950, OBS=10, FIRSTOBS=3
- an option remains in effect until you change it
- some options can be locally set/reset in some PROCs
- use PROC OPTIONS to view options
Setting System Options

```sas
OPTIONS options;
```

- can be anywhere in the program
- can be repeated
- many `options` are of on/off modes, e.g., `NUMBER|NONUMBER, DATE|NODATE, CENTER|NOCENTER`
- others are in the form of `option=value`, e.g., `PAGENO=1, LINESIZE=80 (or LS=80), PAGESIZE=40 (or PS=40), YEARCUTOFF=1950, OBS=10, FIRSTOBS=3`
- an option remains in effect until you change it
- some options can be locally set/reset in some PROCs
- use PROC OPTIONS to view options
Setting System Options

OPTIONS options;

can be anywhere in the program

can be repeated

- many options are of on/off modes, eg. NUMBER|NONUMBER, DATE|NODATE, CENTER|NOCENTER
- others are in the form of option=value, eg. PAGENO=1, LINESIZE=80 (or LS=80), PAGESIZE=40 (or PS=40), YEARCUTOFF=1950, OBS=10, FIRSTOBS=3
- an option remains in effect until you change it
- some options can be locally set/reset in some PROCs
- use PROC OPTIONS to view options
Setting System Options

**OPTIONS** *options*;

- can be anywhere in the program
- can be repeated
- many *options* are of on/off modes, eg.
  NUMBER|NONUMBER, DATE|NODATE, CENTER|NOCENTER
- others are in the form of *option=value*, eg.
  PAGENO=1, LINESIZE=80 (or LS=80), PAGESIZE=40 (or PS=40), YEARCUTOFF=1950, OBS=10, FIRSTOBS=3
- an option remains in effect until you change it
- some options can be locally set/reset in some PROCs
- use PROC OPTIONS to view options
Setting System Options

OPTIONS options;

- can be anywhere in the program
- can be repeated
- many *options* are of on/off modes, eg. NUMBER|NONUMBER, DATE|NODATE, CENTER|NOCENTER
- others are in the form of *option=value*, eg. PAGENO=1, LINESIZE=80 (or LS=80), PAGESIZE=40 (or PS=40), YEARCUTOFF=1950, OBS=10, FIRSTOBS=3
- an option remains in effect until you change it
- some options can be locally set/reset in some PROCs
- use PROC OPTIONS to view options
Setting System Options

OPTIONS `options;`

can be anywhere in the program

can be repeated

many `options` are of on/off modes, eg.
NUMBER|NONUMBER, DATE|NODATE, CENTER|NOCENTER

others are in the form of `option=value`, eg.
PAGENO=1, LINESIZE=80 (or LS=80), PAGESIZE=40 (or PS=40), YEARCUTOFF=1950, OBS=10, FIRSTOBS=3

an option remains in effect until you change it

some options can be locally set/reset in some PROCs

use PROC OPTIONS to view options
Setting System Options

OPTIONS *options*;

- can be anywhere in the program
- can be repeated
- many *options* are of on/off modes, eg.
  NUMBER|NONUMBER, DATE|NODATE, CENTER|NOCENTER
- others are in the form of *option*=*value*, eg.
  PAGENO=1, LINESIZE=80 (or LS=80), PAGESIZE=40 (or PS=40), YEARCUTOFF=1950, OBS=10, FIRSTOBS=3
- an option remains in effect until you change it
- some options can be locally set/reset in some PROCs

use PROC OPTIONS to view options
Setting System Options

OPTIONS options;

- can be anywhere in the program
- can be repeated
- many options are of on/off modes, eg. NUMBER|NONUMBER, DATE|NODATE, CENTER|NOCENTER
- others are in the form of option=value, eg. PAGENO=1, LINESIZE=80 (or LS=80), PAGESIZE=40 (or PS=40), YEARCUTOFF=1950, OBS=10, FIRSTOBS=3
- an option remains in effect until you change it
- some options can be locally set/reset in some PROCs
- use PROC OPTIONS to view options
Outline

1. The Language
   - Language Elements
   - SAS Log
   - SAS Libraries

2. SAS Data Sets
   - Two Portions of a SAS Data Set

3. SAS Output
   - SAS Output: An Example

4. System Options

5. SAS Comments

6. Report Writing—Round One
Two styles of SAS comments

- `*text;`
- `/*text*/` (note: starts from column 3 for /* in some operating systems)

Note:

1. Each style can span over several lines
2. Combine spaces/indentations with proper comments to create a clean, readable program style
3. Use of Control-/ key shortcut for second style comment and use of Control-Shift-/ key for uncommenting second style line by line
Two styles of SAS comments

- `*text;`
- `/*text*/` (note: starts from column 3 for /* in some operating systems)

Note:
1. Each style can span over several lines
2. Combine spaces/indentations with proper comments to create a clean, readable program style
3. Use of Control-/ key shortcut for second style comment and use of Control-Shift-/ key for uncommenting second style line by line
Two styles of SAS comments

- `*text;`
- `/*text*/` (note: starts from column 3 for /* in some operating systems)

Note:

1. Each style can span over several lines
2. Combine spaces/indentations with proper comments to create a clean, readable program style
3. Use of Control-/ key shortcut for second style comment and use of Control-Shift-/ key for uncommenting second style line by line
Two styles of SAS comments

- `*text;`
- `/*text*/` (note: starts from column 3 for /* in some operating systems)

Note:

1. Each style can span over several lines
2. Combine spaces/indentations with proper comments to create a clean, readable program style
3. Use of Control-/- key shortcut for second style comment and use of Control-Shift-/- key for uncommenting second style line by line
Two styles of SAS comments

- *text;
- /*text*/ (note: starts from column 3 for /* in some operating systems)

Note:
1. Each style can span over several lines
2. Combine spaces/indentations with proper comments to create a clean, readable program style
3. Use of Control-/ key shortcut for second style comment and use of Control-Shift-/ key for uncommenting second style line by line
Two styles of SAS comments

- `*text;`
- `/*text*/` (note: starts from column 3 for /* in some operating systems)

Note:

1. Each style can span over several lines
2. Combine spaces/indentations with proper comments to create a clean, readable program style
3. Use of Control-/ key shortcut for second style comment and use of Control-Shift-/ key for uncommenting second style line by line
Two styles of SAS comments

- `*text;`
- `/*text*/` (note: starts from column 3 for /* in some operating systems)

Note:

1. Each style can span over several lines
2. Combine spaces/indentations with proper comments to create a clean, readable program style
3. Use of Control-\ key shortcut for second style comment and use of Control-Shift-\ key for uncommenting second style line by line
Outline

1. The Language
   - Language Elements
   - SAS Log
   - SAS Libraries

2. SAS Data Sets
   - Two Portions of a SAS Data Set

3. SAS Output
   - SAS Output: An Example

4. System Options

5. SAS Comments

6. Report Writing—Round One
Use of PROC PRINT

- basic report
- selecting variables (VAR statement)
- subsetting observations (WHERE statement or WHERE= data set option)
- column sums (SUM statement)
- NOOBS option and ID statement
- BY group listing; with ID statement; with SUM statement; with PAGEBY statement
- DOUBLE option
- LABEL statement together with LABEL procedure statement option
- FORMAT statement
Use of PROC PRINT

- basic report
  - selecting variables (VAR statement)
  - subsetting observations (WHERE statement or WHERE= data set option)
  - column sums (SUM statement)
  - NOOBS option and ID statement
  - BY group listing; with ID statement; with SUM statement; with PAGEBY statement
  - DOUBLE option
  - LABEL statement together with LABEL procedure statement option
  - FORMAT statement
Use of PROC PRINT

- basic report
- selecting variables (VAR statement)
  - subsetting observations (WHERE statement or WHERE= data set option)
  - column sums (SUM statement)
  - NOOBS option and ID statement
- BY group listing; with ID statement; with SUM statement; with PAGEBY statement
- DOUBLE option
- LABEL statement together with LABEL procedure statement option
- FORMAT statement
Use of PROC PRINT

- basic report
- selecting variables (VAR statement)
- subsetting observations (WHERE statement or WHERE= data set option)
- column sums (SUM statement)
- NOOBS option and ID statement
- BY group listing; with ID statement; with SUM statement; with PAGEBY statement
- DOUBLE option
- LABEL statement together with LABEL procedure statement option
- FORMAT statement
Use of PROC PRINT

- basic report
- selecting variables (VAR statement)
- subsetting observations (WHERE statement or WHERE= data set option)
- column sums (SUM statement)
- NOOBS option and ID statement
- BY group listing; with ID statement; with SUM statement; with PAGEBY statement
- DOUBLE option
- LABEL statement together with LABEL procedure statement option
- FORMAT statement
Use of PROC PRINT

- basic report
- selecting variables (VAR statement)
- subsetting observations (WHERE statement or WHERE= data set option)
- column sums (SUM statement)
- NOOBS option and ID statement
  - BY group listing; with ID statement; with SUM statement; with PAGEBY statement
  - DOUBLE option
  - LABEL statement together with LABEL procedure statement option
  - FORMAT statement
Use of PROC PRINT

- basic report
- selecting variables (VAR statement)
- subsetting observations (WHERE statement or WHERE= data set option)
- column sums (SUM statement)
- NOOBS option and ID statement
- BY group listing; with ID statement; with SUM statement; with PAGEBY statement
- DOUBLE option
- LABEL statement together with LABEL procedure statement option
- FORMAT statement
Use of PROC PRINT

- basic report
- selecting variables (VAR statement)
- subsetting observations (WHERE statement or WHERE= data set option)
- column sums (SUM statement)
- NOOBS option and ID statement
- BY group listing; with ID statement; with SUM statement; with PAGEBY statement
- DOUBLE option
- LABEL statement together with LABEL procedure statement option
- FORMAT statement
Use of PROC PRINT

- basic report
- selecting variables (VAR statement)
- subsetting observations (WHERE statement or WHERE= data set option)
- column sums (SUM statement)
- NOOBS option and ID statement
- BY group listing; with ID statement; with SUM statement; with PAGEBY statement
- DOUBLE option
- LABEL statement together with LABEL procedure statement option
- FORMAT statement
Use of PROC PRINT

- basic report
- selecting variables (VAR statement)
- subsetting observations (WHERE statement or WHERE= data set option)
- column sums (SUM statement)
- NOOBS option and ID statement
- BY group listing; with ID statement; with SUM statement; with PAGEBY statement
- DOUBLE option
- LABEL statement together with LABEL procedure statement option
- FORMAT statement