Midterm 1 Notes

SAS Environment

5 Main Windows

1. Editor
2. Log
3. Output
4. Results
5. Explorer

SAS File Types

1. .sas are code files
2. .log files are produced when a program is run, this log file contains info about the SAS job (warning/errors, time taken, etc.)
3. The .lst file contains the output of the statistical procedures

SAS File Statements

SAS Statements begin with an identifying keyword and they always end with a semicolon. SAS statements are free-format.

Submitting SAS Programs

1. Run -> Submit
2. Running Man icon
3. F8

DATA steps and PROC Steps

- DATA Step: Read in a data file, assign variable name/labels/formats, or select specific observations.
- PROC Step: Perform 'utility' operations on a dataset, analyze data, or output results/imports.

SAS Libraries

You can create libraries through the `libname` command:

```
libname mylib 'C:\'
```

SAS has 2 default libraries:

1. 'Work' is the default SAS library, but is temporary
2. 'sasuser' is the permanent library

Library names can't be longer than 8 characters.

SAS Datasets

Are comprised of two parts.

1. Descriptor: contains information about the dataset: dataset name, labels, date/time created, storage info, # of observations, then info for each variable such as name/type/length/position/format/informat/label
2. Data portion: Contains the observation for each variable.

`PROC Contents`

Command to display the descriptor portion of a SAS Dataset.

SAS Variables

- Character: Stored as a string. Stored from 1 byte 32767 bytes, where 1 byte is 1 character.
- Numeric: Stored as a floating point number in 8 bytes of storage by default. 8 bytes of storage allows for 16 or 17 bytes of significant digits.

Variable and Library names

SAS names can be 32 characters at maximum, regardless of case. Can start with either lowercase or uppercase.

Library names must be less than 9 characters but are case-*insensitive*. Note that you cannot use things like "SASHELP"/"SASMSG"/"SASUSER" for user-defined libraries.

Variables vs Observations

Variables = Columns, Observations = Rows

Titles and Footnotes

Titles

Format of title statement:

```
TITLE<n> 'Text'
```

The default title is 'The SAS System', and there are up to 16 configurable lines of title. Note that an unnumbered title is equivalent to using "Title1".

Running the command `title;` will cancel all titles.

Footnotes

Format of footnote statement: `FOOTNOTE <n> 'Text';`

There is *not* a default footnote.

SAS System Options

General format is `OPTIONS <option>`. To generate the list of OPTIONS, run:

```
PROC options;
run;
```

The full list of options is below:

```
ANIMATION=STOP    Specifies whether to start or stop animation.
 ANIMDURATION=MIN  Specifies the number of seconds that each animation frame
displays.
 ANIMLOOP=YES      Specifies the number of iterations that animated images repeat.
 ANIMOVERLAY       Specifies that animation frames are overlaid in order to view
all frames.
 APPEND=           Specifies an option=value pair to insert the value at the end of
the existing option value.
 APPLETLOC=        Specifies the location of Java applets, which is typically a URL.
 ARMAGENT=         Specifies an ARM agent (which is an executable module or keyword,
such as LOG4SAS) that contains a specific
                   implementation of the ARM API.
 ARMLOC=ARMLOG.LOG Specifies the location of the ARM log.
 ARMSUBSYS=(ARM_PROC)
                   Specifies the SAS ARM subsystems to enable or disable.
 AUTOCORRECT       Automatically corrects misspelled procedure names and keywords,
and global statement names.
 AUTOEXEC=/opt/sasinside/SASConfig/Lev1/SASApp/WorkspaceServer/autoexec.sas
                   Specifies the location of the SAS AUTOEXEC files.
 AUTOSAVELOC=      Specifies the location of the Program Editor auto-saved file.
 NOAUTOSIGNON      Disables a SAS/CONNECT client from automatically submitting the
SIGNON command remotely with the RSUBMIT command.
 BINDING=DEFAULT   Specifies the binding edge type of duplexed printed output.
 BOMFILE           Writes the byte order mark (BOM) prefix when a Unicode-encoded
file is written to an external file.
 BOTTOMMARGIN=0.000 IN
                   Specifies the size of the margin at the bottom of a printed page.
 BUFNO=1           Specifies the number of buffers for processing SAS data sets.
 BUFSIZE=0         Specifies the size of a buffer page for output SAS data sets.
 BYERR             SAS issues an error message and stops processing if the SORT
procedure attempts to sort a _NULL_ data set.
 BYLINE            Prints the BY line above each BY group.
 BYSORTED          Requires observations in one or more data sets to be sorted in
alphabetic or numeric order.
 NOCAPS            Does not convert certain types of input, and all data lines,
into uppercase characters.
 NOCARDIMAGE       Does not process SAS source code and data lines as 80-byte
records.
 CATCACHE=0        Specifies the number of SAS catalogs to keep open in cache
memory.
 CBUFNO=0          Specifies the number of extra page buffers to allocate for each
open SAS catalog.
```

```
 CENTER            Center SAS procedure output.
 CGOPTIMIZE=3      Specifies the level of optimization to perform during code
compilation.
 NOCHARCODE        Does not substitute specific keyboard combinations for special
characters that are not on the keyboard.
 NOCHKPTCLEAN      Does not erase files in the Work library after a batch program
successfully executes in checkpoint mode or
                   restart mode.
 CLEANUP           Performs automatic continuous cleanup of non-essential resources
in out-of-resource conditions.
 NOCMDMAC          Does not check window environment commands for command-style
macros.
 CMPLIB=           Specifies one or more SAS data sets that contain compiler
subroutines to include during compilation.
 CMPMODEL=BOTH     Specifies the output model type for the MODEL procedure.
 CMPOPT=(NOEXTRAMATH NOMISSCHECK NOPRECISE NOGUARDCHECK NOGENSYMNAMES
NOFUNCDIFFERENCING SHORTCIRCUIT NOPROFILE)
                   Specifies the type of code-generation optimizations to use in
the SAS language compiler.
 NOCOLLATE         Does not collate multiple copies of printed output.
 COLOPHON=         Specifies the comment text that is included in graphic stream
files.
 COLORPRINTING     Prints in color if color printing is supported.
 COMAMID=TCP       Specifies the communication access method for connecting client
and server sessions across a network.
 COMPRESS=NO       Specifies the type of compression to use for observations in
output SAS data sets.
 CONNECTEVENTS     Clients receive SAS events propagated from a SAS/CONNECT server.
 CONNECTMETACONNECTION
                   At sign-on, connects the SAS/CONNECT server to the SAS Metadata
server.
 CONNECTOUTPUT=BUFFERED
                   Specifies whether to send the SAS/CONNECT server log and list
output immediately, or to buffer the output.
 CONNECTPERSIST    Continues a client/server connection after an RSUBMIT statement
has completed.
 CONNECTREMOTE=    Specifies the ID of a specific server session that a client
connects to.
 CONNECTSTATUS     Displays the Transfer Status window during file transfers.
 CONNECTWAIT       Executes RSUBMIT statements synchronously.
 COPIES=1          Specifies the number of copies to print.
 CPUCOUNT=2        Specifies the number of processors that thread-enabled
applications should assume are available for concurrent
                   processing.
 CPUID             Prints the CPU identification number at the beginning of the SAS
log.
 CSTGLOBALLIB=     Specifies the location of the SAS Clinical Standards Toolkit
global library.
 CSTSAMPLELIB=     Specifies the location of the SAS Clinical Standards Toolkit
sample library.
 DATAPAGESIZE=CURRENT
                   Specifies whether the page size for a data set or utility file
is compatible with SAS 9.3 processing, or is
                   determined by the current version of SAS.
 DATASTMTCHK=COREKEYWORDS
                   Specifies which SAS statement keywords are prohibited from being
specified as a one-level DATA step name to
```

```
                   protect against overwriting an input data set.
 DATE              Prints the date and time that a SAS program started.
 DATESTYLE=MDY     Specifies the sequence of month, day, and year when ANYDTDTE,
ANYDTDTM, or ANYDTTME informat data is ambiguous.
 NODBFMTIGNORE     Uses the numeric data type in tables.
 NODBIDIRECTEXEC   The SQL pass-through facility does not optimize the handling of
SQL statements.
 DBSLICEPARM=(THREADED_APPS, 2)
                   Specifies whether SAS procedures, applications, and the DATA
step can read DBMS tables in parallel, and the
                   number of threads to use to read the DBMS tables.
 DBSRVTP=NONE      Specifies whether SAS/ACCESS engines hold or block the
originating client while making performance-critical
                   calls to the database.
 DCSHOST=LOCALHOST Specifies the host name of the SAS Document Conversion Server.
 DCSPORT=7111      Specifies the port number of the SAS Document Conversion Server.
 DECIMALCONV=COMPATIBLE
                   Specifies the binary to decimal conversion and formatting
methodology.
 DEFLATION=6       Specifies the level of compression for device drivers that
support the Deflate compression algorithm.
 NODETAILS         Does not display additional information when files are listed in
a SAS library.
 DEVICE=           Specifies the device driver to which SAS/GRAPH sends procedure
output.
 DFLANG=LOCALE     Specifies the language for international date informats and
formats.
 DKRICOND=ERROR    Specifies the error level to report when a variable is missing
from an input data set during the processing of a
                   DROP=, KEEP=, or RENAME= data set option.
 DKROCOND=WARN     Specifies the error level to report when a variable is missing
from an output data set during the processing of
                   a DROP=, KEEP=, or RENAME= data set option.
 NODLCREATEDIR     Does not create a directory for the SAS library that is named in
a LIBNAME statement when the directory does not
                   already exist.
 DLDMGACTION=FAIL  Specifies the type of action to take when a SAS data set or a
SAS catalog is detected as damaged.
 NODMR             Does not invoke a server session for use with a SAS/CONNECT
client.
 NODMS             Starts SAS using an interactive line-mode session.
 NODMSEXP          Starts SAS using an interactive line-mode session.
 DMSLOGSIZE=99999  Specifies the maximum number of rows that the SAS Log window can
display.
 DMSOUTSIZE=2147483647
                   Specifies the maximum number of rows that the SAS Output window
can display.
 DMSPGMLINESIZE=136
                   Specifies the maximum number of characters in a Program Editor
line.
 NODMSSYNCHK       Disables syntax check mode for DATA step and PROC step
processing in the windowing environment.
 DQLOCALE=         Specifies the Data Quality Server ordered list of locales for
data cleansing.
 DQOPTIONS=        Specifies the SAS session parameters for data quality programs.
 DQSETUPLOC=       Specifies the location of the Quality Knowledge Base root
directory.
```

```
 DS2ACCEL=NONE     Provides support for DS2 code pass-through acceleration.
 DS2SCOND=WARN     Specifies the type of message that PROC DS2 generates.
 DSACCEL=NONE      Provides support for code pass-through acceleration.
 DSNFERR           Issues an error message and stops processing when a SAS data set
cannot be found.
 DTRESET           SAS updates the date and time in the titles of the SAS log and
the procedure output file.
 NODUPLEX          Does not print output using duplex (two-sided) printing.
 NOECHOAUTO        Does not write statements that are in the AUTOEXEC file to the
SAS log as they are executed.
 EMAILACKWAIT=30   Specifies the number of seconds to wait for the SMTP server
acknowledgement.
 EMAILAUTHPROTOCOL=NONE
                   Specifies the SMTP e-mail authentication protocol.
 NOEMAILFROM       Does not require the FROM e-mail option when sending e-mail by
using the FILE or FILENAME statements.
 EMAILHOST=LOCALHOST
                   Specifies one or more domain names for SMTP e-mail servers.
 EMAILID=          Specifies the SAS user's logon ID, profile or e-mail address.
 EMAILPORT=25      Specifies the port number for the SMTP e-mail server that is
specified in the EMAILHOST option.
 EMAILPW=XXXXXXXX  Specifies the password for the e-mail address specified by the
EMAILID option.
 EMAILUTCOFFSET=   For SMTP e-mail sent using the FILENAME statement, specifies a
UTC offset that is used in the Date header field
                   of the e-mail message.
 NOENCRYPTFIPS     Does not limit SAS/SECURE and SSL security services to use FIPS
140-2 algorithms.
 ENGINE=V9         Specifies the default access method for SAS libraries.
 NOERRORABEND      Does not end SAS for most errors, issues an error message, sets
OBS=0, and goes into syntax check mode.
 NOERRORBYABEND    Does not end a SAS program when an error occurs in BY-group
processing, issues an error, and continues
                   processing.
 ERRORCHECK=NORMAL Specifies whether SAS enters syntax-check mode when errors are
found in the LIBNAME, FILENAME,  %INCLUDE, and
                   LOCK statements.
 ERRORS=20         Specifies the maximum number of observations for which SAS
issues complete error messages.
 EVENTDS=(DEFAULTS)
                   Specifies one or more data sets that define custom holiday
events.
 NOEXPLORER        Does not invoke Explorer and the Program Editor when SAS starts.
 EXTENDOBSCOUNTER=YES
                   Specifies whether to extend the maximum number of observations
in a new SAS data file.
 FILESYNC=HOST     Specifies when operating system buffers that contain contents of
permanent SAS files are written to disk.
 FIRSTOBS=1        Specifies the observation number or external file record that
SAS processes first.
 NOFMTERR          Issues a note for missing variable formats, uses w. or $w., and
continues processing.
 FMTSEARCH=( APFMTLIB WORK LIBRARY )
                   Specifies the order in which format catalogs are searched.
 FONTEMBEDDING     Enables font embedding for Universal Printing and SAS/GRAPH
printing.
 FONTRENDERING=FREETYPE_POINTS
```

```
                   Specifies whether some SAS/GRAPH devices render fonts by using
the operating system or by using the Free Type
                   engine.
 FONTSLOC=/opt/sasinside/SASHome/ReportFontsforClients/9.4
                   Specifies the location of the fonts that are supplied by SAS.
Names the default font file location for
                   registering fonts that use the FONTREG procedure.
 FORMCHAR=|----|+|---+=|-/\<>*
                   Specifies the default output formatting characters.
 FORMDLIM=         Specifies the character to delimit page breaks in SAS output for
the LISTING destination.
 FORMS=DEFAULT     If forms are used for printing, specifies the default form to
use.
 GSTYLE            Uses ODS styles to generate graphs that are stored as GRSEG
catalog entries.
 GWINDOW           Displays SAS/GRAPH output in the GRAPH window.
 HELPADDR=         Specifies the address of the remote Help system.
 HELPBROWSER=REMOTE
                   Specifies the browser to use for SAS Help and ODS output.
 HELPENCMD         Uses the English version of the keyword list for the
command-line Help.
 HELPHOST=         Specifies the name of the computer where the remote browser is
to send Help and ODS output.
 HELPINDEX=(/help/common.hlp/index.txt /help/common.hlp/keywords.htm common.hhk)
                   Specifies one or more index files for SAS Help and Documentation.
 HELPPORT=0        Specifies the port number for the remote browser client.
 HELPTOC=(/help/helpnav.hlp/navigation.xml /help/common.hlp/toc.htm common.hhc)
                   Specifies the table of contents files for the online SAS Help
and Documentation.
 HOSTINFOLONG      Print operating environment information in the SAS log when SAS
starts.
 HTTPSERVERPORTMAX=0
                   Specifies the highest port number that can be used by the SAS
HTTP server for remote browsing.
 HTTPSERVERPORTMIN=0
                   Specifies the lowest port number that can be used by the SAS
HTTP server for remote browsing.
 IBUFNO=0          Specifies the number of extra buffers to be allocated for
navigating an index file.
 IBUFSIZE=0        Specifies the buffer page size for an index file.
 IMLPACKAGEPRIVATE=~/sas/iml/packages
                   Specifies the location for SAS/IML packages in the private
collection.
 IMLPACKAGEPUBLIC=/opt/sas/iml/packages
                   Specifies the location for SAS/IML packages in the public
collection.
 IMLPACKAGESYSTEM=!SASROOT/misc/iml/packages
                   Specifies the location for SAS/IML packages in the system
collection.
 NOIMPLMAC         Does not check for statement-style macros.
 INITCMD=          Specifies commands to open applications, or windows and text
editor commands, after SAS executes the AUTOEXEC=
                   file and the INITSTMT= value.
 INITSTMT=         Specifies SAS statements to execute after any statements in the
AUTOEXEC= file and before any statements from
                   the SYSIN= file.
 INSERT=           Specifies an option=value pair to insert the value at the
```

```
beginning of the existing option value.
 INTERVALDS=       Specifies interval=library pairs.  Library is a SAS data set that contains a custom interval data set; interval
                   can be used in the INTNX and INTCK functions.
 INVALIDDATA=.     Specifies the value that SAS assigns to a variable when invalid numeric data is encountered.
 NOIPADDRESS       Disables the IP address to appear in SAS/CONNECT messages when using TCP/IP.
 JPEGQUALITY=75    Specifies the JPEG quality factor that determines the ratio of image quality to the level of compression for
                   JPEG files produced by the JPEG device driver.
 LABEL             Enables procedures to use labels with variables.
 NOLABELCHKPT      For batch programs, disables the recording of checkpoint-restart data for labeled code sections.
 LABELCHKPTLIB=WORK
                   Specifies the libref of the library where the checkpoint-restart data is saved for labeled code sections.
 NOLABELRESTART    Disables restart mode, which executes batch programs using checkpoint-restart data collected at labeled code
                   sections.
 LEFTMARGIN=0.000 IN
                   Specifies the print margin for the left side of the page.
 LINESIZE=132      Specifies the line size for the SAS log and for SAS procedure output for the LISTING destination.
 LOCALEDATA=SASLOCALE
                   Specifies the location of the locale database.
 LOCKDOWN          Specifies that access to files and certain SAS features will be restricted. This feature is only applicable for
                   a SAS session executing in a batch or server processing mode.
 LOGAPPLNAME=      Specifies a SAS session name for SAS logging.
 LOGCONFIGLOC=/opt/sasinside/SASConfig/Lev1/SASApp/WorkspaceServer/logconfig.xml
                   Specifies the name of the XML configuration file or a basic logging configuration that is used to initialize the
                   SAS logging facility.
 LOGLANGCHG        Enables changing the language of the SAS output when the LOCALE= option is changed.
 NOLOGLANGENG      Write SAS log messages based on the values of the LOGLANGCHG, LSWLANG=, and LOCALE= options when SAS started.
 LOGPARM=WRITE=BUFFERED ROLLOVER=NONE OPEN=REPLACE
                   Specifies when SAS log files are opened, closed, and according to the LOG= system option, how they are named.
 LRECL=32767       Specifies the default logical record length to use for reading and writing external files.
 LSWLANG=LOCALE    Specifies the language for SAS log and ODS messages when the LOCALE= option is set after SAS starts.
 MACRO             Enables the macro facility.
 MAPEBCDICTOASCII= Specifies the transcoding table that is used to convert characters from ASCII to EBCDIC and EBCDIC to ASCII.
 MAPS=MAPS         Specifies the location of SAS/GRAPH map data sets.
 MAPSGFK=          Specifies the location of GfK maps.
 MAPSSAS=          Specifies the location of SAS map data sets.
 NOMAUTOCOMPLOC    Does not display the autocall macro source location in the SAS log when the autocall macro is compiled.
 NOMAUTOLOCDISPLAY Disables the macro facility from displaying the autocall macro source location in the log.
 NOMAUTOLOCINDES   Does not prepend the full pathname of the autocall macro source file to the autocall macro catalog entry
```

```
                   description field in the WORK.SASMACR catalog.
 MAUTOSOURCE       Enables the macro autocall feature.
 MAXSEGRATIO=75    Specifies the upper limit for the percentage of index segments
that the SPD Engine identifies as containing the
                   value referenced in the WHERE expression.
 MCOMPILE          Allows new macro definitions.
 MCOMPILENOTE=NONE Specifies what to write to the SAS log when a macro compiles
successfully.
 NOMCOVERAGE       Disables the generation of coverage analysis data for SAS macros.
 MCOVERAGELOC=     Specifies the location of the macro coverage analysis data file.
 MERGENOBY=NOWARN  Specifies the type of message that is issued when MERGE
processing occurs without an associated BY statement.
 MERROR            Issues a warning message for an unresolved macro reference.
 METAAUTORESOURCES=SASApp
                   Specifies the metadata resources that are assigned when SAS
starts.
 METACONNECT=      Specifies the profile from the metadata user connection profiles
that is used to connect to the SAS Metadata
                   Server.
 METAENCRYPTALG=SASPROPRIETARY
                   Specifies the type of encryption to use to communicate with the
SAS Metadata Server.
 METAENCRYPTLEVEL=CREDENTIALS
                   Specifies the level of encryption that is used to communicate
with the SAS Metadata Server.
 METAID=           Specifies the ID of the SAS Metadata Server.
 METAPASS=XXXXXXXX Specifies the SAS Metadata Server password.
 METAPORT=8561     Specifies the TCP port for the SAS Metadata Server.
 METAPROFILE=      Specifies the XML document that contains SAS Metadata Server
user connection profiles.
 METAPROTOCOL=BRIDGE
                   Specifies the network profile to use to connect to the SAS
Metadata Server.
 METAREPOSITORY=Foundation
                   Specifies the name of the SAS Metadata Server Repository.
 METASERVER=       Specifies the host name or address of the SAS Metadata Server.
 METASPN=          Specifies the service principal name (SPN) for the SAS Metadata
Server.
 METAUSER=         Specifies the user ID that is used to connect to the SAS
Metadata Server.
 NOMEXECNOTE       Does not display the macro execution information in the SAS log
when the macro is invoked.
 MEXECSIZE=65536   Specifies the maximum macro size that can be executed in memory.
 NOMFILE           Does not write MPRINT output to an external file.
 MINDELIMITER=     Specifies the character delimiter for the macro IN operator.
 NOMINOPERATOR     Disables IN logical operators in expressions.
 MINPARTSIZE=16777216
                   Specifies the minimum size of the data component partitions for
SPD Engine data sets.
 MISSING=.         Specifies the character to print for missing numeric values.
 NOMLOGIC          Does not trace macro execution or write the results to the SAS
log.
 NOMLOGICNEST      Does not display the macro nesting information in the SAS log
for MLOGIC output.
 NOMPRINT          Does not display the SAS statements that are generated by macro
execution.
 NOMPRINTNEST      Does not display the macro nesting information from the MPRINT
```

```
output in the SAS log.
 NOMRECALL         Searches the autocall libraries only once for a requested macro.
 MREPLACE          Enables updates to macro definitions in the Work library.
 MSGLEVEL=N        Specifies the level of detail in SAS log messages.
 NOMSTORED         Does not search for stored compiled macros.
 MSYMTABMAX=4194304
                   Specifies the maximum amount of memory available to the macro
variable symbol table or tables.
 NOMULTENVAPPL     List only operating environment fonts in the font selector
window of a SAS application.
 MVARSIZE=65534    Specifies the maximum size for a macro variable that is stored
in memory.
 NONETENCRYPT      Does not require encryption for client/server data transfers.
 NETENCRYPTALGORITHM=SASPROPRIETARY
                   Specifies one or more algorithms to use for encrypted
client/server data transfers.
 NETENCRYPTKEYLEN=0
                   Specifies the key length that is used by the encryption
algorithm for encrypted client/server data transfers.
 NEWS=!SASROOT/misc/base/news
                   Specifies the location of the news file that is to be written to
the SAS log immediately after the header.
 NONLDECSEPARATOR  Disables formatting of numeric output using the decimal
separator for the locale.
 NOTES             SAS writes notes to the SAS log.
 NUMBER            Prints the page number on the first title line of each page of
SAS output.
 OBJECTSERVER      Enables SAS to run as an Integrated Object Model (IOM) server.
 OBS=9223372036854775807
                   Specifies the observation that is used to determine the last
observation to process, or specifies the last
                   record to process.
 ODSDEST=AUTO      Specifies the default ODS destination.
 ODSGRAPHICS=AUTO  Specifies the setting for ODS graphics.
 ODSLANGCHG        Enables the language of the SAS message text in ODS output to
change when the LOCALE option is set after start
                   up.
 ODSSTYLE=AUTO     Specifies the ODS HTML default style.
 OLAPCONFIG=       Specifies the name of the XML configuration file that is used to
initialize an OLAP server.
 ORIENTATION=PORTRAIT
                   Specifies the paper orientation to use when printing to a
printer.
 NOOVP             Disables overprinting of error messages to make them bold.
 NOPAGEBREAKINITIAL
                   Does not begin SAS log and procedure output for the LISTING
destination on a new page.
 PAGENO=1          Resets the SAS output page number.
 PAGESIZE=60       Specifies the number of lines that compose a page of the SAS log
and SAS output.
 PAPERDEST=        Specifies the name of the output bin to receive printed output.
 PAPERSIZE=LETTER  Specifies the paper size to use for printing.
 PAPERSOURCE=      Specifies the name of the paper bin to use for printing.
 PAPERTYPE=PLAIN   Specifies the type of paper to use for printing.
 PARM=             Specifies a parameter string that is passed to an external
program.
 PARMCARDS=FT15F001
```

```
                   Specifies the file reference to open when SAS encounters the
PARMCARDS statement in a procedure.
 PDFACCESS         Enables screen readers to read PDF text and graphics.
 NOPDFASSEMBLY     Disables assembly of PDF documents.
 NOPDFCOMMENT      Disables comments in PDF documents from being modified.
 NOPDFCONTENT      Disables modification of PDF document content.
 PDFCOPY           Enables PDF document text and graphics to be copied.
 PDFFILLIN         Enables PDF forms to be filled in.
 PDFPAGELAYOUT=DEFAULT
                   Specifies the page layout for PDF documents.
 PDFPAGEVIEW=DEFAULT
                   Specifies the page viewing mode for PDF documents.
 PDFPASSWORD=XXXXXXXX
                   Specifies the password to use to open a PDF document and the
password used by a PDF document owner.
 PDFPRINT=HRES     Specifies the resolution to print PDF documents.
 PDFSECURITY=NONE  Specifies the level of encryption to use for PDF documents.
 NOPRESENV         Specifies that collecting data for the preservation of the SAS
environment is disabled.
 PRIMARYPROVIDERDOMAIN=
                   Specifies the domain name of the primary authentication provider.
 PRINTERPATH=PDF   Specifies the name of a registered printer to use for Universal
Printing.
 NOPRINTINIT       Preserves the procedure output file for the LISTING destination
if no new output is created.
 PRINTMSGLIST      Specifies to print the entire list of messages to the SAS log.
 QUOTELENMAX       Writes a warning message to the SAS log if a quoted string
exceeds the maximum length allowed.
 REPLACE           Enables replacement of permanent SAS data sets.
 REUSE=NO          Specifies whether SAS reuses space when observations are added
to a compressed SAS data set.
 RIGHTMARGIN=0.000 IN
                   Specifies the print margin for the right side of the page.
 NORLANG           Disables SAS from executing R language statements.
 RSASIOTRANSERROR  Displays a transcoding error when illegal values are read from a
remote application.
 RSASUSER          Opens the Sasuser library in Read-Only mode.
 S=0               Specifies the length of statements on each line of a source
statement, and the length of data on lines that
                   follow a DATALINES statement.
 S2=0              Specifies the length of statements of each line of a source
statement from an %INCLUDE statement, an AUTOEXEC=
                   file, or an autocall macro file.
 S2V=0             Specifies the column to begin reading a file with variable
length records that is specified in an %INCLUDE
                   statement, an autoexec file, or an autocall macro.
 SASAUTOS=( "SASEnvironment/SASMacro" '!SASROOT/sasautos' )
                   Specifies the location of one or more autocall libraries.
 SASCMD=           Specifies the command that starts a server session on a
symmetric multiprocessing (SMP) computer.
 SASFRSCR=#LN00002 A read-only option that contains the fileref, generated by the
SASSCRIPT option,  for SAS/CONNECT server sign-on
                   script files.
 SASHELP=( '!SASROOT/nls/u8/sascfg'          '!SASROOT/nls/u8/sashelp'
'!SASROOT/nls/en/sascfg'          '!SASROOT/sashelp' )
                   Specifies the location of the Sashelp library.
 SASMSTORE=        Specifies the libref of a SAS catalog for stored compiled SAS
```

```
macros.
 SASSCRIPT=(          '!SASROOT/misc/connect'         )
                   Specifies one or more locations of SAS/CONNECT server sign-on
script files.
 SASUSER=~/sasuser.v94
                   Specifies a libref or a path that identifies a library for the
user's profile catalog.
 SECPACKAGE=Negotiate
                   Specifies the security package that the IOM server uses to
authenticate incoming client connections.
 SECPACKAGELIST=Kerberos,NTLM
                   Specifies the security authentication packages that are used by
the server.
 SEQ=8             Specifies the length of the numeric portion of the sequence
field in input source lines or data lines.
 SERROR            Issues a warning message when a macro variable reference does
not match a macro variable.
 NOSETINIT         Disables PROC SETINIT to prevent updating site licensing
information.
 SHARESESSIONCNTL=SERVER
                   Specifies whether the SAS/SHARE server has one or multiple
connections to clients.
 SIGNONWAIT        Executes the SIGNON statement synchronously, signing on clients
to the server one at a time.
 SKIP=0            Specifies the number of lines to skip at the top of each page of
SAS output for the LISTING destination.
 SOLUTIONS         Displays the Solutions menu in SAS windows.
 SORTDUP=PHYSICAL  Specifies whether PROC SORT removes duplicate variables based on
the DROP and KEEP options or on all data set
                   variables.
 SORTEQUALS        PROC SORT maintains the relative position in the output data set
for observations with identical BY-variable
                   values.
 SORTSEQ=          Specifies a language-specific collating sequence for the SORT
and SQL procedures.
 SORTSIZE=1073741824
                   Specifies the amount of memory that is available to the SORT
procedure.
 NOSORTVALIDATE    SORT does not verify whether a data set is sorted according to
the variables in the BY statement.
 SOURCE            Writes program source statements to the SAS log.
 NOSOURCE2         Does not write secondary source statements from included files
to the SAS log.
 NOSPDEFILECACHE   Disables caching of opened SPD Engine files.
 SPDEINDEXSORTSIZE=33554432
                   Specifies the memory size for sorting index values.
 SPDEMAXTHREADS=0  Specifies the maximum number of threads that the SPD Engine can
spawn for I/O processing.
 SPDEPARALLELREAD=NO
                   Enables or disables SPD Engine parallel reads when no WHERE
clause is in effect.
 SPDESORTSIZE=33554432
                   Specifies the memory size that is used for sorting by the SPD
Engine.
 SPDEUTILLOC=      Specifies one or more locations where the SPD Engine can
temporarily store utility files.
 SPDEWHEVAL=COST   Specifies the WHERE statement evaluation process for the SPD
```

```
Engine.
 NOSPOOL           Does not write SAS statements to a utility data set in the Work
library.
 SQLCONSTDATETIME  PROC SQL replaces references to the DATE, TIME, DATETIME, and
TODAY functions with their equivalent constant
                   values before a query executes.
 SQLGENERATION=(NONE DBMS='TERADATA DB2 ORACLE NETEZZA ASTER GREENPLM HADOOP
SAPHANA IMPALA HAWQ POSTGRES REDSHIFT SQLSVR VERTICA')
                   Specifies whether and when SAS procedures generate SQL for
in-database processing of source data.
 NOSQLIPONEATTEMPT Allows an SQL query to continue processing when an implicit
pass-through request fails.
 SQLMAPPUTTO=SAS_PUT
                   Specifies the PUT function mapping to SQL.
 SQLREDUCEPUT=DBMS For PROC SQL, specifies the engine type to use to optimize a PUT
function in a query.
 SQLREDUCEPUTOBS=0 For PROC SQL, specifies the minimum number of observations that
must be in a table for PROC SQL to optimize the
                   PUT function in a query.
 SQLREDUCEPUTVALUES=0
                   For PROC SQL, specifies the maximum number of SAS format values
that can exist in a PUT function expression to
                   optimize the PUT function in a query.
 SQLREMERGE        PROC SQL processes queries that use remerged data.
 SQLUNDOPOLICY=REQUIRED
                   Specifies how PROC SQL handles updated data if errors occur
while you are updating data.
 NOSSLCLIENTAUTH   Does not require the server to perform client authentication for
a server connection.
 NOSSLCRLCHECK     Does not check the Certificate Revocation List (CRL) when a
digital certificate is validated.
 NOSSPI            Does not use Security Support Provider Interface for single
sign-on connections to IOM servers.
 NOSTARTLIB        Does not assign user-defined permanent librefs when SAS starts.
 NOSTEPCHKPT       Disables recording of checkpoint-restart data for DATA and PROC
steps for batch programs.
 STEPCHKPTLIB=WORK Specifies the libref of the library where checkpoint-restart
data for DATA and PROC steps is saved.
 NOSTEPRESTART     Disables restart mode which executes batch programs using
checkpoint-restart data collected for DATA and PROC
                   steps in a prior execution.
 STRIPESIZE=       Specifies path and size pairs to identify I/O device stripe size.
Stripe size indicates page size when creating
                   a data set or utility file.
 SUMSIZE=0         Specifies a limit on the amount of memory that is available for
data summarization procedures when class
                   variables are active.
 SVGAUTOPLAY       Starts animation when the page is loaded in the browser.
 NOSVGCONTROLBUTTONS
                   Does not display the paging control buttons and an index in a
multipage SVG document.
 SVGFADEIN=0       Specifies the number of seconds for the fade-in effect for a
graph.
 SVGFADEMODE=OVERLAP
                   Specifies whether to use sequential frames or to overlap frames
for the fade-in effect of a graph.
 SVGFADEOUT=0      Specifies the number of seconds for a graph to fade out of view.
 SVGHEIGHT=        Specifies the height of the viewport. Specifies the value of the
height attribute of the outermost SVG element.
 NOSVGMAGNIFYBUTTON
                   Disables the SVG magnifier tool.
 SVGPRESERVEASPECTRATIO=
                   Specifies whether to force uniform scaling of SVG output.
Specifies the preserveAspectRatio attribute on the
                   outermost SVG element.
 SVGTITLE=         Specifies the text in the title bar of the SVG output. Specifies
the value of the TITLE element in the SVG file.
 SVGVIEWBOX=       Specifies the coordinates, width, and height that are used to
set the viewBox attribute on the outermost SVG
                   element.
 SVGWIDTH=         Specifies the width of the viewport. Specifies the value of the
width attribute of the outermost SVG element.
 SVGX=             Specifies the x-axis coordinate of one corner of the rectangular
region for an embedded SVG element. Specifies
                   the x attribute in the outermost SVG element.
 SVGY=             Specifies the y-axis coordinate of one corner of the rectangular
region for an embedded SVG element. Specifies
                   the y attribute in the outermost SVG element.
 NOSYMBOLGEN       Does not display the results of resolving macro variable
references in the SAS log.
 SYNCHIO           Requires that data set I/O must be completed before other
logical SAS tasks can be executed.
 NOSYNTAXCHECK     Disables syntax check mode for multiple steps in non-interactive
or batch SAS sessions.
 SYSPARM=          Specifies a character string that can be passed to SAS programs.
 SYSPRINTFONT=     Specifies the default font to use for printing.
 NOSYSRPUTSYNC     Sets the %SYSRPUT macro variables in the client session when a
synchronization point is encountered.
 TBUFSIZE=0        Specifies the size of the buffer that is used by SAS
applications to transfer client/server data across a
                   network.
 TCPLISTENTIME=0   Specifies the amount of time that a SAS/CONNECT server listens
for a client to connect before terminating the
                   CONNECT server session.
 TCPPORTFIRST=0    Specifies the first value in a range of TCP/IP ports for a
client to use to connect to a server.
 TCPPORTLAST=0     Specifies the last value in a range of TCP/IP ports for a client
to use to connect to a server.
 TENANTID=         Specifies a name that identifies a tenant in a multi-tenant
environment.
 NOTERMINAL        Does not associate a terminal with a SAS session.
 TERMSTMT=         Specifies the SAS statement to execute when SAS terminates.
 TEXTURELOC=!SASROOT/misc/textures
                   Specifies the location of textures and images that are used by
ODS styles.
 THREADS           Uses threaded processing for SAS applications that support it.
 TIMEZONE='GMT-07:00'
                   Specifies a time zone.
 TOOLSMENU         Displays the Tools menu in SAS windows.
 TOPMARGIN=0.000 IN
                   Specifies the print margin at the top of the page.
 TRAINLOC=         Specifies the URL for SAS online training courses.
 TRANTAB=          Specifies the translation table catalog entries.
 TSID=             Specifies a logical server metadata object that identifies a
```

```
table service definition.
 UBUFNO=0           Specifies the number of utility file buffers.
 UBUFSIZE=0         Specifies the size of utility file buffers.
 UNIVERSALPRINT     Enables the Universal Printing windows to display and sets up
default values for printing.
 UPRINTCOMPRESSION Enables compression of files that are created by some Universal
Printers and SAS/GRAPH devices.
 URLENCODING=SESSION
                    Specifies whether the argument to the URLENCODE function and to
the URLDECODE function is interpreted using the
                    SAS session encoding or UTF-8 encoding.
 USER=              Specifies the default permanent library to use for one-level SAS
data set names.
 UTILLOC=WORK       Specifies one or more file system locations in which threaded
applications can store utility files.
 UUIDCOUNT=100      Specifies the number of UUIDs to acquire from the UUID Generator
Daemon.
 UUIDGENDHOST=      Specifies the host and port, or the LDAP URL that the UUID
Generator Daemon runs on.
 V6CREATEUPDATE=NOTE
                    Specifies the type of message to write to the SAS log when
Version 6 data sets are created or updated.
 VALIDFMTNAME=LONG Specifies the maximum size that user-created formats and
informat names can be before an error or warning is
                    issued.
 VALIDMEMNAME=COMPAT
                    Specifies the rules for naming SAS data sets, SAS data views,
and item stores.
 VALIDVARNAME=V7    Specifies the rules for valid SAS variable names that can be
created and processed during a SAS session.
 VARINITCHK=NOTE    Specifies the type of message to write to the SAS log when a
variable is not initialized.
 VARLENCHK=WARN     Specifies the type of message to write to the SAS log when the
length of the variable that is being read is
                    longer than the length that is defined for the variable.
 VBUFSIZE=65536     Specifies the buffer size for a view.
 VIEWMENU           Displays the View menu in SAS windows.
 VNFERR             SAS issues an error message when a BY variable exists in one
data set but not another when the other data set is
                    _NULL_.

WORK=/tmp/SAS_workDA510000467A_localhost.localdomain/SAS_workC5000000467A_localhost.
localdomain
                    Specifies the libref or location of the Work library.
 WORKINIT           At SAS invocation, erases files that exist from a previous SAS
session in an existing Work library.
 WORKTERM           Erases the Work files when SAS terminates.
 YEARCUTOFF=1926    Specifies the first year of a 100-year span that is used by date
informats and functions to read a two-digit
                    year.
 _LAST_=_NULL_      Specifies the most recently created data set.

Host Options:

 ALIGNSASIOFILES    Aligns SAS files on a page boundary for improved performance.
 ALTLOG=            Specifies the location for a copy of the SAS log when SAS is
running in batch mode.
```

```
 ALTPRINT=          Specifies the location for a copy of the SAS procedure output
when SAS is running in batch mode.
 AUTHPROVIDERDOMAIN=
                    Specifies the authentication provider that is associated with a
domain.
 BLKSIZE=256        Specifies the number of bytes that are read or written in one
I/O operation.
 COMAUX1=           Specifies the first alternate communication access method.
 COMAUX2=           Specifies the second alternate communication access method.
 CONFIG=( /opt/sasinside/SASHome/SASFoundation/9.4/sasv9.cfg
/opt/sasinside/SASHome/SASFoundation/9.4/nls/u8/sasv9.cfg
/opt/sasinside/SASHome/SASFoundation/9.4/sasv9_local.cfg
/opt/sasinside/SASConfig/Lev1/SASApp/sasv9.cfg
/opt/sasinside/SASConfig/Lev1/SASApp/sasv9_usermods.cfg
/opt/sasinside/SASConfig/Lev1/SASApp/WorkspaceServer/sasv9.cfg
/opt/sasinside/SASConfig/Lev1/SASApp/WorkspaceServer/sasv9_usermods.cfg )
                    Specifies the configuration file that is used when initializing
or overriding the values of SAS system options.
 DBCS               Enables double-byte character sets for encoding values that
support East Asian languages.
 DBCSLANG=UNKNOWN   Specifies a double-byte character set language.
 DBCSTYPE=UTF8      Specifies the encoding method that is used for a double-byte
character set.
 ECHO=              Specifies a message that is echoed to the SAS log while
initializing SAS.
 EDITCMD=           Specifies the host editor that is used with the HOSTEDIT command.
 EMAILSYS=smtp      Specifies the e-mail protocol that is used for sending
electronic mail.
 ENCODING=UTF-8     Specifies the default character-set encoding for the SAS session.
 FILELOCKS=( '/' FAIL )
                    Specifies whether file locking is turned on or off and what
action should be taken if a file cannot be locked.
 FILELOCKWAIT=0     Specifies the number of seconds that SAS will wait for a locked
file.
 FILELOCKWAITMAX=600
                    Specifies the maximum number of seconds that SAS waits for a
locked file to become available.
 FSDBTYPE=DEFAULT   Specifies a full-screen double-byte character set (DBCS)
encoding method.
 FSIMM=             Specifies input method modules (IMMs) for full-screen
double-byte character sets (DBCS).
 FSIMMOPT=          Specifies options for input method modules (IMMs) that are used
with a full-screen double-byte character set
                    (DBCS).
 NOFULLSTIMER       Does not write performance statistics to the SAS log.
 HELPLOC=( '!SASROOT/X11/native_help' '!SASROOT/X11/native_help'
'!SASUSER/classdoc' )
                    Specifies the location of the text and index files for the
facility that is used to view the online SAS Help and
                    Documentation.
 INGOPTS=           Specifies the options to pass to the INGRES database.
 JREOPTIONS=(
-DPFS_TEMPLATE=/opt/sasinside/SASHome/SASFoundation/9.4/misc/tkjava/qrpfstpt.xml
-Djava.class.path=/opt/sasinside/SASHome/SASVersionedJarRepository/eclipse/plugins/
sas.launcher.jar
-Djava.security.auth.login.config=/opt/sasinside/SASHome/SASFoundation/9.4/misc/tkj
ava/sas.login.config
-Djava.security.policy=/opt/sasinside/SASHome/SASFoundation/9.4/misc/tkjava/sas.pol
icy
-Djava.system.class.loader=com.sas.app.AppClassLoader
-Dlog4j.configuration=file:/opt/sasinside/SASHome/SASFoundation/9.4/misc/tkjava/sas.
log4j.properties
-Dsas.app.class.path=/opt/sasinside/SASHome/SASVersionedJarRepository/eclipse/plugi
ns/tkjava.jar
-Dsas.ext.config=/opt/sasinside/SASHome/SASFoundation/9.4/misc/tkjava/sas.java.ext.
config
-Dtkj.app.launch.config=/opt/sasinside/SASHome/SASVersionedJarRepository/picklist
)
                   Specifies the Java Runtime Environment options for SAS.
 LOADMEMSIZE=0     Specifies a suggested amount of memory that is needed for
executable programs loaded by SAS.
 LOCALE=EN_US      Specifies a set of attributes in a SAS session that reflect the
language, local conventions, and culture for a
                   geographical region.
 LOG=              Specifies a location for the SAS log when SAS is running in
batch mode.
 LPTYPE=           Specifies the UNIX command that is used to print files.
 MAXMEMQUERY=268435456
                   For certain procedures, specifies the maximum amount of memory
that can be allocated per request.
 MEMSIZE=2147483648
                   Specifies the limit on the amount of virtual memory that can be
used during a SAS session.
 MSG=!SASROOT/sasmsg
                   Specifies the path to the library that contains SAS messages.
 NOMSGCASE         Specifies that SAS writes notes, warning, and error messages in
mixed casing.
 NONLSCOMPATMODE   Encodes data using the SAS session encoding.
 NOOPLIST          Does not write SAS system option settings to the SAS log.
 PATH=( !SASROOT/dbcs/sasexe !SASROOT/sasexe )
                   Specifies one or more search paths for SAS executable files.
 PRINT=            Specifies a location for SAS output when running in batch mode.
 PRINTCMD=         Specifies the print command that SAS is to use.
 REALMEMSIZE=0     Specifies the amount of real memory SAS can expect to allocate.
 RTRACE=NONE       Specifies whether to produce a list of resources that are read
or loaded during a SAS session.
 RTRACELOC=        Specifies the pathname for the file that records the list of
resources that are read or loaded during a SAS
                   session.
 SEQENGINE=TAPE    Specifies the sequential engine to be used for accessing tapes
and pipes.
 SET=[SASROOT = /opt/sasinside/SASHome/SASFoundation/9.4] [SASAUTOS = (
'!SASROOT/sasautos'         )] [SAMPSRC =
(                  '!SASROOT/samples/base'         '!SASROOT/samples/dbi'
'!SASROOT/samples/ets'
'!SASROOT/samples/hps'         '!SASROOT/samples/hpstat'
'!SASROOT/samples/iml'         '!SASROOT/samples/stat'         )]
[SAMPSIO = (                  '!SASROOT/samples/base'
'!SASROOT/samples/dbi'         '!SASROOT/samples/ets'
'!SASROOT/samples/hps'         '!SASROOT/samples/hpstat'
'!SASROOT/samples/iml'         '!SASROOT/samples/stat'         )]
[SASHOME = /opt/sasinside/SASHome] [APFMTLIB = SASEnvironment/SASFormats]
                   Defines an environment variable.
 SORTANOM=         Specifies options for the host sort utility.
```

~~~
 SORTCUT=0          Specifies the data size in number of observations above which
SAS uses the host sort instead of the internal SAS
                    sort.
 SORTCUTP=0         Specifies the data size in bytes above which SAS uses the host
sort instead of the internal SAS sort.
 SORTDEV=           Specifies the pathname to temporary files that are created by
the host sort utility.
 SORTNAME=          Specifies the name of the host sort utility.
 SORTPARM=          Specifies the parameters for the host sort utility.
 SORTPGM=BEST       Specifies whether to use the SAS sort utility or the host sort
utility or to let SAS choose the sort utility.

SSLCALISTLOC=/opt/sasinside/SASHome/SASSecurityCertificateFramework/1.1/cacerts/tru
stedcerts.pem
                    Specifies the location of digital certificates for trusted
certification authorities (CA).
 SSLCERTLOC=        Specifies the location of the digital certificate that is used
for authentication.
 SSLCRLLOC=         Specifies the location of the Certificate Revocation List (CRL).
 SSLPKCS12LOC=      Specifies the location of the PKCS #12 DER encoding package file.
 SSLPKCS12PASS=XXXXXXXX
                    Specifies the password that SSL requires for decrypting the
private key of the PKCS #12 DER encoding package
                    file.
 SSLPVTKEYLOC=      Specifies the location of the private key that corresponds to
the digital certificate.
 SSLPVTKEYPASS=XXXXXXXX
                    Specifies the password that SSL requires for decrypting the
private key.
 NOSTDIO            SAS does not use the standard streams stdin, stdout, and stderr.
 STIMEFMT=(NLDATM2. HMS TIMEAMPM KB MEMFULL TSFULL NC)
                    Specifies the format that is used to display the FULLSTIMER and
STIMER output for timestamp, memory, CPU and
                    elapsed time statistics.
 STIMER             Writes real and CPU time to the SAS log.
 SYSIN=             Specifies the SAS program to execute in batch.
 SYSPRINT=          Specifies the destination for printed output.
 TAPECLOSE=REREAD   CLOSE disposition for a library on tape.
 USERCONFIG         Process .sasv9.cfg and sasv9.cfg configuration files in user's
home directory.
 NOVERBOSE          Does not write start-up system options to the SAS log.
 WORKPERMS=700      Sets the permissions of the SAS Work library when it is
initially created.
 NOXCMD             Disables the X command in SAS.
 ```

 ## Listing Reports/``` PROC Print```

 The default output from the PRINT procedure prints the dataset in HTML format.
Running the print procedure without specifying a dataset will print the most recent
dataset created.

 The default title is the "The SAS System" and there is no footnote default for the
PRINT procedure.

 You can specify labels in the PRINT procedure by doing:
~~~

```
 ```
 Label OldLabelName = "New Label Name";
 ```

You can specify formats by doing:
```

format ;

```
where the "format-type" is some built-in format.

To make your own format, use ```PROC FORMAT```, for example:
```

PROC FORMAT value sex 1 = "Female" 2 = "Male" ;

run;

```
To suppress the ID column on the left, use the ```NOOBS``` option (that hides the
observations).

To specify conditions on your variables, use the "WHERE = " command.  You can also
apply the regular comparison/logical operators to your variable.

To sort variables, you must use the "BY" grouping, for example:
```

PROC SORT data=input out=sorted_data; BY ; run;

```
This outputs a sorted dataset "sorted_data" that is sorted by the specific variable.

To paginate your print procedure based on a variable, you need to also use "BY" in
the print procedure. Example:
```

PROC print data; by ; pageby ; run;

~~~
To instead assign the observation # to a variable, use the command ```ID
<variable>;```.

To print a specific number of observations, you can use the "Obs" variable in
```proc print```, like:
~~~

PROC print data= my_data(obs = n) run;

````
## ```PROC Sort```

Example:
````

PROC sort Data = mydata OUT = sorteddata; by ; run;

```
Note: By default, SAS sorts in *ascending* order.

## SAS Syntax Rules

SAS statements begin with a keyword and always end with a semi-colon, and are free
format. Single statements can span multiple lines and several statements can be on
one line.

## Sources of SAS Data

Many sources of data.

1. Existing SAS dataset
2. Other software
3. Data entry
4. Raw data in SAS programs
5. Raw data in an external file

## Input statement

Identifies the variable in the raw data file. General format is:
```

INPUT ;

```
## List, Column, Formatted Input

* List input: each data value is separated by a space delimiter.

* Column input: Each data value is in a fixed location.

* Formatted input: Uses SAS Informats.


## Informats

Informal control the way SAS reads the data in.

## Relative/Absolute Pointer control
For column input, the relative pointer control lets you specify the column position
relative to one another, using the "+" notation.

Absolute pointer control lets you specify the exact column index.

## PROC IMPORT
```

````
You can use the import option to import data from another software, like Excel.

## PROC Datasets

This is used to change variable attributes. The general format is:
````

PROC DATASETS LIBRARY = libref; MODIFY ; RENAME LABEL FORMAT INFORMAT run;

````
## Keep/Drop statements, Options
Use the keep/drop statements for selecting which variables you want to preserve.
Example:
````

DATA libref.new_data_set; KEEP variables; run;

````
This affects the output data set.

IF you use the keep/drop statement in a SET statement, you can eliminate variables
from the input statement, for example:
````

DATA libref.new_data_set; SET SAS-data-set(KEEP =);

````
## Creating Variables
Use variable assignment in the ```DATA``` step to create a new variable. An
assignment statement evaluates the expression and assigns the resulting value to a
variable.

General form:
````

DATA output-sas-dataset; SET input-sas-dataset; variable = expression; run;

````
## SAS expression

An expression contains operands and operators that form a set of instructions that
produce a value. Operands are variable names or constants. Operators are arithmetic
symbols and SAS functions.

## SAS Arithmetic Operators

| Operator | Function | Priority |
|---------|-----------|------|
| + | Addition | 3 |
| - | Subtraction | 3 |
| / | Division | 2 |
| * | Multiplication | 2 |
| ** | Exponentiation | 1 |
````

```
| - | Negation | 1 |


## SAS Function
A SAS function is a routine that  that returns a value that is determined from
specific arguments. An exmaple of a SAS function is ```sum```. Note that sample
statistics function ignore missing values.

## Date constants
Use a date constnat to return a SAS date value for a specific date. Example:
```

evaldate = '14FEB2009'd;

```

sets the value of the evaldate variable to 17942 which is the SAS date value
(number of days before Jan 1, 1960) corresponding to February 14, 2009.

## DateTime value

A SAS datetime value is interpreted as the number of seconds between midnight of
January 1st 1960 and a separate date time. The date part wil return just the date
property which is the number of days overall since Jan 1 1960.


## Conditional Processing
Conditional processing allows you to execute statements from IF-THEN logic,
examples:

1. Control length of character variables with ```LENGTH```
2. Select rows to include in a dataset

## IF-THEN, ELSE

Basic format:
```

IF THEN ; ELSE ;

```

Expression contains operands and operators that form a set of instructions that
produce a value.

Note that you can only use one executable statement per each if-then-else block.


## Selecting Observations - ```DELETE```, ```WHERE```, ```IF```

In the DATA step you can make subsets of the observations by using the "WHERE",
"DELETE", and "IF" statements.

Note that if the variable *already* exists in the input data set, you can use the
"WHERE" statement, otherwise you must use the "IF" statement.

Note that you can only subset observations with the WHERE statement inside the DATA
```

```
step.



----------------------------------------

# Midterm 2 Notes

## Data Step Compilation and Execution
The DATA step is processed in 2 phases:

1. Compilation
2. Execution

At compile-time, SAS creates:

* An input buffer to hold the current raw data file record that is being processed.

* A program data vector (pdv) to hold the current SAS observation.

* The descriptor portion of the dataset.

The empty input buffer and empty program data vector are used to sotre the incoming
data. SAS then loads the data, line-by-line, parses it into variables, and then
outputs those values to the SAS dataset.

## Type of Data Errors

SAS detects data errors when:

* The input statement encounters "invalid data" in a field
* Illegal arguments are used in functions
* Impossible mathematical operations are requested


## Examining Data Errors
After a SAS dataset is encountered:

1. A note that describes the error is printed in the SAS log.

2. The input record being read is displayed in the SAS log (which is the contents
of the input buffer)

3. The values in the SAS observation being created are displayed in the SAS log
(which is the contents of the PDV)

4. A missing variable is assigned to the appropriate variable.

Finally, execution of the program continues.

## Tips to avoid/identify programming errors

1. Use the Enhanced editor because it color-codes keywords and errors

2. Write in small parts and test each part individually

3. Clear the log and output before running program
```

```
4. Review the log, looking for red or green text.

5. Confirm the number of records and variables in each dataset using the log.

6. Keep all variables in your interim data sets.

7. Inspect the datasets you create in table editor or using proc print.

## Combining Datasets

### Appending/concatenating

To concatenate two data sets, they must have the same variables. The variable names
and data types should be the same for both datasets. Use the SET statement to
append/concatenate, example:
```

DATA SAS-data-set; SET SAS-dataset-1 SAS-dataset-2; RUN;

```
### Merging data sets

To merge datasets, there must be at least one common variable between the datasets,
and also datasets must be sorted on a variable *prior* to merging. Example of Merge:
```

DATA SAS-dataset; MERGE SAS-data-sets; BY ; run;

```
### Identifier Variable

An identifier variable serves the purpose of indicating which dataset each
observation came from, so that you can *identify* an observation's original dataset
when you see it in the combined dataset.

Note that this must be done in a *separate* data step *prior* to the actual merge
step.


### Common vs unique variables

When combining multiple datasets, the common variables are the variables that exist
in all datasets, and the unique variables are those that only exist in one dataset
but not all.

### Rename option
If you're appending dataset, you can use the "RENAME=" option to create the new
common variable names. If you're merging, you can use the rename option to create
unique variable names for the merged dataset (will not affect original datasets).

Example:
```

SAS-dataset-1(RENAME = (old-name-1 = "New Name 1"))

```
### Match merging

* The records from each dataset with the same value of the (unique) 'BY' variable
are linked, and outputted as one record

* If you omit the 'BY' statement, the first record from each dataset will be
outputted as one record *without being linked* by a common variable.

In match merging, the data sets must be sorted on the BY variable *prior* to
merging.

### 1 to Many merging
Unique BY values are in one dataset and duplicate by matching values are in the
other dataset.


### Many-to-many merging
* Duplicate matching BY values are in both data sets.

### The IN option
Use the 'IN' option to create variables identifying which datasets contained the
observation.

Example:
```

DATA allscores; MERGE midterm(IN=InMidterm) final (IN=InFinal); by Name; if InMidterm and InFinal; run;

```



## Labels and formats (permanent vs. temporary?)

Using LABEL and FORMAT statements in the:

1. DATA step means that they are *permanent*
2. PROC step means that they are *temporary*


## ```PROC MEANS```

 is a helpful way of summarizing a dataset, as it will calculate and display simple
summary statistics (mean, std dev, max/min)

You can specify which variables to include by using the 'VAR' keyword. Example:
```

proc means data=ia.admit; var age height weight; run;

```
The 'BY' and 'CLASS' keywords let you create separate summaries either for each BY
group or for each CLASS group.
```

```
The 'OUTPUT' keyword lets you save the summary statistics generated to a separate
dataset.

You can also specify which summary statistics you'd like to compute by listing them
in the PROC means declaration, like so:
```

proc means data=ia.admit n mean stddev run;

```
## ```PROC FREQ```

This procedure calculates and displays frequency counts (frequency, percent,
cumulative frequency, cumulative percent). It also displays how many observations
have missing values.

You can also use the 'TABLES' statement to select which variables you want analyzed
in the FREQ procedure. Example:
```

PROC Freq Data = SAS-data-set; TABLES .... ; RUN:

```
## 1-way and 2-way tables


2-way tables (also know as crosstabs) allow you to categorize observations on the
combination of *two* sets of categories, rather than just 1. Example:
```

PROC FREQ Data=SAS-data-set; TABLES variable1 * variable2; RUN:

```
## User-defined formats and ```PROC FREQ```

Note that you can still apply user-created formats in a FREQ procedure. Example:
```

PROC format; value $code_fmt 'FLTAT1' - 'FLTAT3' = 'Flight Attendant' 'PILOT1' - 'PILOT3' = 'Pilot' run;

PROC FREQ data= ia.crew; format JobCode $code-fmt.; tables JobCode; run;

```
## Row, Column, Cell percents

Use the 'PCTN' family of statistics to compute percentages inside of a TABULATE
procedure. The PCTN family includes:

1. COLPCTN for finding column percentages
2. ROWPCTN for finding row percentages
```

```
For example:
```

proc tabulate data=sashelp.cars format=8.2; class origin type; table type all, (origin all)*(n*f=8. colpctn) / rts=10; run;

```
Will print the percentages for each column in the dataset.


## ```PROC Tabulate```

Allows you to output your dataset as a table, and offers several features:

1. Control of table construction
2. Differentiating between classification variables and anlaysis variables
3. Specifying statistics
4. Formatting of values
5. Labelling variables and statistics

The general format of a TABULATE step is:
```

PROC TABULATE DATA=SAS-data-set ; CLASS class-variables; VAR analysis-variables; TABLE page-expression, row-expression, column-expression; run;

```
Class variables are used to define subgroups on one or more dimensions.

Summary statistics are calculated for the analysis variables (in the VAR statement)

If you use the command
```

table All;

```
Then it'll concatenate information for that variable into an "All" column.

You can also instead use a comma to move to a new dimension, like so:
```

Table ,

```
To format the output, add the format at the beginning of the PROC tabulate
declaration, like so:
```

proc tabulate data=ia.flat format=dollar12.;

```

```

```
Note that the format option changes the default for all cells.

To specify another statistic besides just frequency, you can put the statistic name
(mean, etc.) at the end of the TABLE command.

Example:
```

proc tabulate data= ia.fltat format=dollar12.; WHERE location in ('CARY', 'FRANKFURT'); class Location JobCode; var Salary; table JobCode, Location*Salary*mean;

```


## ```PROC Report```

The report procedure allows you to:

1. Create listing reports
2. Create summary Reports
3. Enhance reports
4. Request separate subtotals and grand totals
5. Generate reports in an interactive point and clock or programming environment.

In a PROC REPORT, character values are *left* justified while numeric values are
*right* justified. Observations are printed in the order in which they are stored.

You can use the column statement only print selected variables, as in:
```

COLUMN ... ;

```
The DEFINE allows you to:

1. Define how each variable is used in the report
2. Assign formats to variables
3. Specify report column headers and column widths
4. Change the order of the rows in the report

In the define statement, character variables are used as Display variables and
Numeric variables are used as analysis variables.

### Width, label, format, and order options

The width option controls the width of a report column.

The format option assigns a format to a variable.

The order keyword identifies the variable used to order rows of the report.

For example, here the variable JobCode is the order variable.
```

proc report data=ia.crew nowd; *No window to supress interactive window!; column JobCode Location Salary; define JobCode / order width=8 'Job Code'; define Location / 'Home Base'; define Salary / format=dollar10.; run;

```
### Group options

You can use the report procedure to create a summary report by defining variables
as group variables -- all observatinos whose group variables have the same values
are collapsed into a single row in the report.

Note that if you have a group variable, you *cannot have any display or order
variables*.

| Group variables | Display/Order variables|
|--|--|
| Produce summary reports (observations collapsed into groups) | Produce listing
report (one row for each observation)|

### Defining Analysis variables
Default usage for numeric variables is always analysis with a default statistic of
sum.

If the report contains group variables, the report displays the *sum of the numeric
variables' values* for *each group*.

If the report contains at least 1 display or order variable and no group variables,
the report lists all of the values of the numeric variables.

If the report contains only numeric variables, the report *displays grand totals
for the numeric variables*.


### Rbreak

Use the BREAK statement to display a total at the end of the group, use the Rbreak
statement to display the grand total at the bottom of the report.

Adding the HEADLINE and HEADSKIP options (after the initial proc report procedure)
will make a basic head line between the variable and first grouping, or allow you
to skip space between the line and the first observation, respectively.

# Final Notes

## PROC GChart

The GChart procedure allows you to

* Specify the form of the chart.

* Identify a chart variable that determines the number of bars or pie slices to
create.

* (Optionally) Identify an analysis variable to use for calculating statistics that
determine the height/length of the bar/slice.
```

```
General format is:
```

PROC GCHART DATA= SAS-Data-Set;

```
To specify the chart format you can use either of:
```

HBAR chart-variable ... / ; VBAR chart-variable ... / ; PIE chart-variable ... / ;

```
### Chart Variable

Here the chart variable (either numeric/character) specifies the number of
bars/slices produced within a graph.

### Defining Graphics Device (GOptions)

THis lets you specify the storage format of the graph and can either be printers or
application drivers.

Examples:
```

goptions dev=HPD; * HP Deskjet printer; gopotions dev=HPL; * HP Laserjet printer; goptions dev=GIF; * GIF driver; goptions dev=TK1; * Tektronix driver; goptions dev=PDF; * Portal document format; goptions dev=PSL; * post-script driver; goptions dev=WMF; * windows metafile driver; goptions dev=ACTIVEX; * ActiveX control; goptions dev=ACTXIMG; * ActiveX image; goptions dev=JAVA; * Java applets; goptions dev=JAVAIMG; *Java image; goptions device=JPG; * JPEG image;

```
Example: how to route the output to a Java ActiveX image within an HTML document.
```

ods html file='vbar.html'; goptions dev=activex;

proc gchart data=ia.crew; vbar JobCode; run;

ods html close; goptions reset=all;

```
### DISCRETE option

* When graphing numeric variables, by default, the values are grouped together to
create a smaller number of lines.

* *To override this*, use the ```DISCRETE``` option in the HBAR, VBAR, or PIE
statement.
```

```
### SUMMARY and TYPE options

To plot a summary statistic, you use the ```SUMVAR= ``` and ```TYPE``` options.

The general format is:
```

SUMVAR= analysis-variable; TYPE = MEAN | SUM;

```
| Command | Description |
| ---- |  ----- |
| SUMVAR | Identifies the analysis variable to use for the sum/mean calculation.|
| TYPE | Specifies that the height/length of the bar/size of the slice represents a
mean/sum of the analysis-variable values.

### FILL and EXPLODE options

The ```FILL``` option lets you specify what type of 'fill' you want to use for the
bars/slices you're using (for example, 'x' fills the slices with a bunch of little
x's).
You can use ``` fill = <fill-type> ``` in the VBAR/PIE/HBAR commands. Note that the
default 'fill' is solid.


The ```EXPLODE``` options let you 'explode' out a slice of the pie chart, to show
emphasis. You can use ```explode = <variable-name>``` in the options for the PIE
statement.

## Title/Footnte options

General format:
```

COLOR = color | C = color FONT = | F= HEIGHT = n | H = n

```
Examples:
```

title color=green 'Number of Pilots by Job Level'; title font=brush color=red 'March Flights'; title height = 3 in font = duplex 'Flights to RDU'; footnote height= 3 'IA's Gross Revenue by Region'; footnote height = 3cm 'Average Salary by Job Level'; footnote height = 3 pct 'Total Flights by Model';

```
## PROC GPLOT

The GPLOT procedure produces scatterplots and line plots. General format:
```

PROC GPLOT DATA=SAS-dataset; PLOT * / ; run;

```
Example:
```

ods html file='plot.html'; ods graphics;

proc gplot data=data1.flight114; where date between '02mar2001'd and '08mar2001'd; * this filters one week; plot Boarded * Date; run;

```
### SYMBOL statement

You can use the symbol statement to:

1. define plotting symbols
2. draw lines through data points
3. specify the color of the plotting symbols/lines

General format:
```

SYMBOL options;

```
For example, to use the square as a plotting symbol and join the points with
straight lines, do:
```

proc gplot data=ia.flight114; where date between '02mar2001'd and '08mar2001'd; plot Boarded * Date; symbol value = square i = join; run;

```
The symbol statement has 2 properties:

1. *Global*: after they are defined, they remain in effect until they are changed
or until the end of the SAS session.
2. *Additive*: Specifying the value of one option doesn't affect the values of the
other options.

The available Symbol values are: Plus, Star, Square, Diamond, Triangle, or none.

The ```I=``` statement above specifies the **interpolation option**. Below are the
valid interpolation values:

| Value | Description |
| ----- |  ---------- |
| Join | Joints the points with straight lines |
| Spline | Joins the points with a smooth line. |
| Needle | Draws vertical lines from the points to horizontal axes. |

To draw a line-plot, you would use:
```

SYMBOL value=none

````
To cancel SYMBOL statements, use either:
````

symbol1;

````
OR
````

goptions reset=symbol;

````
### Axis appearance

Use:

| Command | Description |
| ---- |    ------- |
| HAXIS = <values> | Scales the horizontal axis. |
| VAXIS = <values> | Scales the vertical axis. |
| CAXIS = <color> | specifies the color of both axes. |
| CTEXT= <color> | specifies the color of the text on both axes.|

Supply these as an option field in the PLOT statement, that runs within a PROC
GPLOT.

You can also specify labels for your axes by doing: ```label <var-name> = 'New axis
name';```

For example:
````

PROC GPLOT data = plot HorizontalVar * VerticalVar / vaxis= to by ctext = ; LABEL = 'New Axis Name'; run;

````
### Regression

To add regression lines, regression equations, and confidence interval, you can add
````

/ regeqn;

````
to the PLOT statement, and:
````

i=rlCLM;

````
as a symbol option, to create a linear regression line with a <CI> confidence
interval.
````

```
## Controlling Data step processing

### Automatic output

```

data forecast; set prog2.growth; ; run;

```
This automatically outputs to the dataset, because by default, SAS writes a record
the output datatset when it reaches the run statement at the end of a dataset.


### The OUTPUT statement

Including an OUTPUT statement in the Data step means that SAS will:

1. write a record immediately
2. Not automatically write at the end of a data step

Using OUTPUT lets you:

1. Create multiple records from a single observation
2. Write to multiple output datasets
3. Combine information from multiple observations into a single record (when
combined with the RETAIN statement)


### RETAIN statement

By default, SAS sets the values of all variables to missing at the start of each
pass through the DATA step, so as a result, you can't save the values of a variable
from one record to the next.

To get around this, use the retain statement to keep the value of a calculated
variable available for use with the next observation.

### First.ByVariable and Last.ByVariable

When a PROC sort procedure is used to sort a dataset, it creates two variables that
can be used to identify the First record in the By Group and the Last record in the
By Group.

### Drop/Keep

By default, the SAS system writes all variables from every input data set to every
output data set. To get around this, use the DROP/Keep statements to control which
variables are written to output data sets.

 ### OBS

 The ```OBS=``` option lets you specify how many observations are read from the
input data set.

### SUM Statement (NOT function)

The sum statement general form is: ```<variable> + <expression>```.
```

```
This statement:

1. Creates the variable on the left side of the plus sign if it does not already
exist.
2. Initializes the variable to zero before the first iteration of the data step.
3. Automatically retains the variable.
4. Adds the value of the expression to the variable at execution.
5. Ignores missing values.
6. Is an alternative to using the RETAIN statement.


### Accumulating totals

After sorting the data on the BY-variable, to make a running total you must:

1. Set the accumulator variable to 0 at the beginning of the loop.
2. Increment the accumulating variable with a sum statement.
3. Output only the last observation of the BY group.

## Writing data to excel file

Example:
```

ods html file = "admit.xls"; title; proc print data=ia.admit noobs; run; ods html close;

```
## Writing data to a CSV file

Example:
```

ods csvall file='admit.csv'; title; proc print data=ia.admit noobs; run; ods csvall close;

```
## Colon Modifier

Use the colon modifier to read each value only as far as the next delimiter. This
allows you to use Informats with list input, but handle non-standard data values.

Example:
```

data airplanes; infile 'airdata.txt'; input ID $ InService : date9. PassCap CargoCap;

```
## Delimiter Options

In SAS, the default delimiter is a space.

Here are the delimiter options:

| Problem | Option |
| ------ | ------- |
```

````
| Non-blank delimiters | DLM='<delimiter>' |
| Missing data at end of row | MISSOVER |
| Missing data represented by consecutive delimiters | DSD |
| Embedded delimiters where values are surrounded by double quotes | DSD |

The DLM option works with both Input statements and FILE statements (when writing *out* data.)

### Missing data at end of row

By default, when there is missing data at the end of a row:

1. SAS loads the next record to finish the observation
2. A note is written to the log
3. SAS loads a new record at the top of the DATA step and continues processing.

The MISSOVER option tells SAS NOT to continue reading from the next line if not all variables have values.

### DSD Option

1. Sets the default delimiter to a comma.
2. Treats consecutive delimeters as missing values.
3. Enables SAS to read values with embedded delimiters if the value is surrounded by double quotes.

## The @ modifier

The trailing '@' modifier tells sAS to hold the current input line for further processing.

| Option | Effect |
| ---- | ------ |
| Single trailing @ | Holds raw data record until 1) an input statement with no trailing @ and 2) the bottom of a DATA step |
| Double trailing @ | Holds raw data records in the input buffer until SAS reads past the end of the line |


## Variable lists

You can use the following variable lists:

1. Numbered range lists
2. Name range lists
3. Name prefix lists
4. Special SAS name lists

### Numbered range lists

Allows you to refer to a set of variables that all start with the same variable name, and end with a number. For example, ```VAR Week1-Week52``` is a variable range list.

### Named Range lists

Allows you to refer to a series of variables that appear in consecutive order
````

```
within a dataset.

For example:

```

Input Salesman $ Mon Tues Wed Thur Fri Sat Sun Region;

PROC PRINT; * name range list ; Var Mon -- Fri; run;

```

### Name Prefix List

You can refer to all variables that begin with a specified character string. For
example,

```

sum(of SALES:) ``` Lets you sum all the variables that are prefixed with the string "SALES".

Special SAS name lists

SAS has a few built-in name lists.

Name	Description
`_NUMERIC_`	All numeric variables that are already defined in the current data step
`_CHARACTER_`	All character variables that are currently defined in the current data step
`_ALL_`	All variables that are currently defined in the current DATA step.