
Index

A

Alignment argument, 189
Ampersand format modifier, 168
Array processing, 121–137

- applications that use multi-dimensional arrays, 133–136
 - calculating average SBP for pre- and post-treatment, 133–134
- data set transformation, 136
- nested loop, 134
- number of observations, 135
- one-dimensional array, 134
- post-treatment results, 134
- pre-treatment measurements, 133
- restructuring data sets by using multi-dimensional array, 135–136
- subscript, 133
- SUM statement, 135
- syntax, 133
- two-dimensional array, 134

array applications, 130–133

- calculating products of multiple variables, 131–132
- creating a group of variables by using arrays, 130–131
- data set transformation, 132
- DO loop, 133
- list of variables, 130
- long format, 132
- pre-treatment measurements, 130
- product of multiple variables, 131
- restructuring data sets using one-dimensional arrays, 132–133
- SUM function, 131
- temporary data elements, 131

compilation and execution phases, 125–126

- array name, 125
- iterations, 127, 128
- PDV creation, 125

syntax errors, 126

defining and referencing

- one-dimensional arrays, 123–125
- array-elements, 123, 124
- array-name, 123, 124, 125
- asterisk, 123
- character elements, 123
- delimiters argument, 123
- initial-value-list, 124
- keywords, 124
- RETAIN statement, 124
- subscript, 125

enriched syntax, 121

exercises, 136–137

functions and operators, 126–130

- array-elements, 126
- bound-n, 126
- DIM function, 127
- DIM, HBOUND, and LBOUND functions, 126–129
- functions, 126
- missing values, 129
- multi-dimensional array, 126
- OF operator, 130
- similar syntax, 126
- using IN and OF operator with array, 129–130

situations for utilizing, 121–122

- DATA step variables, 122
- example, 121
- IF statements, 122
- multi-dimensional arrays, 122
- one-dimensional array, 122
- referencing of data, 122

B

BY-group processing in DATA step, 79–93

- applications, 85–92
 - calculating mean score within each BY group, 87–88
 - counter variable, 87

- creating data sets with duplicate or non-duplicate observations, 88–89
- cumulating variable, 85, 87
- data set observations, 92
- example, 89
- identical observations, 88
- long format, 91
- longitudinal data, 85
- obtaining the most recent non-missing data within each BY group, 89–91
- previously sorted variable, 87
- restructuring data sets from long format to wide format, 91–92
- RETAIN statement, 90
- wide format, 91
- BY-group processing, 79–86
 - execution phase of BY-group processing, 81–85
 - fifth iteration, 86
 - first iteration, 82
 - FIRSTVARIABLE and LASTVARIABLE, 79–81
 - fourth iteration, 85
 - grouping variables, 79
 - longitudinal data, 79
 - multiple variables, 79
 - second iteration, 83
 - subsetting IF statement, 83
 - SUM statement, 81
 - third iteration, 84
 - total score for each subject, 81
- exercises, 92–93

C

- Cartesian product, 147
- Catalog-specification, 233
- Character variables, 2
- Charlist argument, 195
- Column pointer-control, 162
- Combining data sets, 139–153
 - exercises, 152–153
 - horizontally combining data sets, 143–152
 - Cartesian product, 147
 - compilation phase, 144
 - descriptor information, 146, 149

- example, 143
- execution phase, 144, 146, 149
- master data set, 151
- match-merging, 147–151
- merge-type, 145
- merging syntax, 146
- one-to-many matching, 147
- one-to-one merging, 146
- one-to-one reading, 143–145
- output data set, 145
- transaction data set, 151
- unmatched observations, 150
- updating data sets, 151–152
- variables from different sources, 143
- vertically combining data sets, 139–143
 - common variables, 141
 - compilation phase, 141, 143
 - concatenating data sets, 139–142
 - execution phase, 141, 143
 - input data sets, sorting of, 142
 - interleaving data sets, 142–143
 - new-name, 141
 - old-name, 141
 - program data vector, 141, 143
 - sum of observations, 139
 - syntax, 139
 - variables from different source, 139
- COMPARE procedure, use of to
 - compare contents of two data sets, 208–215
- base data set, 214
- common variables, 210
- comparing observations with
 - common ID values, 212–215
- comparison data set, 214
- first weight value, 209
- information provided from PROC COMPARE, 209–212
- matching variables, 209
- observation differences, 210
- options, 209
- VAR statement, 212
- Count argument, 194

D

- Data entry error, 56
- Data input and output, 155–179

- creating text files, 175–177
 - alignment parameters, 176
 - column output, 175–176
 - end-column, 175
 - formatted output, 176
 - last-column, 175
 - list output, 177
 - PUT statement, 177
 - start-column, 175
 - variable, 175
- exercises, 177–179
- reading text files, 160–174
 - ampersand format modifier, 168
 - character values, 161
 - column input, 161–162
 - column pointer-control, 162
 - consecutive blanks, 168
 - creating observations by using
 - line-hold specifiers, 172–174
 - creating observations by using
 - line pointer-controls, 171–172
 - data execution, 165
 - data in free format, 169
 - Delimiter-Sensitive-Data
 - option, 167
 - double trailing “at” signs, 172, 173
 - embedded blanks, 168
 - example, 163, 171
 - formatted input, 162–164
 - INFILE statement, 161, 172
 - informat in modified list input, 169
 - input buffer, 174
 - input pointer, 171, 173
 - INPUT statement, 164
 - input values, 161
 - line pointer-control, 162
 - list input, 164–168
 - missing values, 167
 - mixed input, 170–171
 - modified list input, 168–170
 - nonstandard numeric value, 170
 - pointer-control, 162
 - program data vector, 160
 - subsetting IF statement, 174
 - syntax, modified list input, 168
- reading and writing text files, 155–160
 - character values, 157
 - checklist, 156
 - data format, 158
 - data informat, 157–158
 - data values, 159
 - decimal scaling factor, 158
 - embedded characters, 158
 - examples of date formats, 160
 - examples of date informats, 159
 - external-file, 155
 - FILENAME statement, 155
 - file-specification, 155, 156, 157
 - INFILE statement, 156
 - named input method, 156, 157
 - observations and variables, 155
 - SAS date and time values, 158–160
 - standard numeric values, 157
 - steps for reading text files,
 - 155–156
 - steps for writing text files, 157
 - text files, 156
- Data sets, combining, *see* Combining data sets
- Data step functions, 181–204
 - character functions, 190–197
 - assignment statement, 191
 - CATT function, 192
 - changing character cases, 190–191
 - character-to-replace, 196
 - character variable, 192
 - charlist argument, 195
 - concatenating character strings,
 - 191–194
 - concatenation operator, 193
 - count argument, 194
 - default delimiter, 195
 - delimiters argument, 190
 - excerpt argument, 194
 - functions for aligning character strings and trimming blanks,
 - 192, 193
 - INDEX function, 195
 - PROPCASE function, 190
 - searching, exacting, and replacing
 - character strings, 194–197
 - source argument, 197
 - string argument, 194
 - syntax, 190
 - target argument, 197
 - TRANWARD function, 196
 - date and time functions, 185–190
 - alignment argument, 189

- creating date and time values, 185–186
- date and time interval functions, 188–190
- date value, 185
- DHMS function, 186
- extracting components from date and time values, 187–188
- functions for creating date and time values, 186
- increment argument, 188
- interval argument, 188
- INTNX function, 189
- numeric values, 185
- start-from argument, 188, 189
- variable description of TENANT data, 187
- exercises, 203–204
- field description for ID variable, 203
- functions and CALL routines, 181–185
 - algebraic expression, 182
 - argument, 181, 183
 - array-name, 181
 - assignment statement, 181
 - CALL routines, 182–184
 - categories, 184–185
 - character variables, 183
 - descriptive statistics
 - functions, 184
 - functions, 181–182
 - mathematical functions, 184
 - MEAN function, 182
 - numeric variables, 183
 - RANUNI function, 185
 - variable-list, 181
- functions for converting variable types, 198–203
 - automatic conversion, 199
 - automatic numeric-to-character conversion, 201
 - character informat, 200
 - informat argument, 200
 - INPUT function, 198–201
 - missing value, 199
 - operator, 201
 - PUT function, 201–203
 - returned values, 202
 - reversed direction of conversion, 198
 - source argument, 200, 201
- DATA step operation, 55–77
 - conditional processing in DATA step, 66–70
 - detecting the end of data set by using the END= option, 68
 - expression, 67
 - final data set, 70
 - initialized variable, 68
 - long format, 68
 - observations, 69
 - restructuring data sets from wide format to long format, 68–70
 - subsetting IF statement, 66–67
 - temporary variable, 68
 - wide format, 68
- DATA step processing overview, 55–62
 - compilation phase, 55
 - data entry error, 56
 - DATA step compilation phase, 57–58
 - DATA step execution phase, 58–61
 - declarative statements, 55
 - difference between reading a raw data set and SAS data set, 61–62
 - end-of-file marker, 59
 - error, 55
 - executable statements, 55
 - execution phase, 55
 - first iteration, 59
 - importance of OUTPUT statement, 61
 - iteration, 55
 - program data vector, 57
 - raw data file, 61
 - second iteration, 60
 - syntax errors, 58
 - third iteration, 60
- debugging techniques, 70–76
 - BREAK command, 75
 - format option, 74
 - highlighted statement, 74
 - logic errors, 70, 74
 - missing variable, 76
 - strategy, 70
 - syntax errors, 70
 - using DATA step debugger, 74–76

- using PUT statement to observe contents of PDV, 70–73
 - variable-list, 71
- exercises, 76–77
- retaining the value of newly created variables, 62–66
 - declarative statement, 63
 - execution phase, 63
 - expression, 65
 - first iteration, 64
 - initialized variable, 62
 - RETAIN statement, 62–64
 - second iteration, 65
 - SUM statement, 64–66
 - third iteration, 66
- Date values, 159
- Decimal scaling factor, 158
- Declarative statement, 55, 63
- Delimiters argument, 123, 190
- Descriptive statistics functions, 184
- E**
- End-of-file marker, 59
- Excerpt argument, 194
- Executable statements, 55
- Exercises
 - array processing, 136–137
 - BY-group processing in DATA step, 92–93
 - combining data sets, 152–153
 - conditional creation of variables, 54
 - data input and output, 177–179
 - data step functions, 203–204
 - DATA step operation, 76–77
 - introduction to SAS®, 33–34
 - SAS® procedures, 239
 - writing loops in DATA step, 117–119
- Explicit loops, 96–102
 - CENTER variable, 103
 - combining loops, 103
 - DO loop, first two iterations, 99
 - expression, evaluated, 101
 - identical statements, 97
 - ID variable, 101
 - increment value, 98
 - index-variable, 98, 102
 - inner loop, 102
 - last two iterations, 100
 - nested loops, 102
 - number of iterations, 99
 - outer loop, 102
 - programming error, 101
 - start value, 98
 - stop value, 98
- F**
- File-specification, 111, 155, 156, 157
- Formatted input method, 164
- G**
- Global statements, 1
- Grouping variables, 79
- H**
- Horizontally combining data sets, 143–152
 - Cartesian product, 147
 - compilation phase, 144
 - descriptor information, 146, 149
 - example, 143
 - execution phase, 144, 146, 149
 - master data set, 151
 - match-merging, 147–151
 - merge-type, 145
 - merging syntax, 146
 - one-to-many matching, 147
 - one-to-one merging, 146
 - one-to-one reading, 143–145
 - output data set, 145
 - transaction data set, 151
 - unmatched observations, 150
 - updating data sets, 151–152
 - variables from different sources, 143
- I**
- IF-THEN/ELSE statement, 35–45
 - bytes of storage space, 42
 - character variable, comparison of, 38
 - DO group, 43–45
 - evaluation of variables, 35
 - FREQ procedure, 43
 - handling missing values when creating variables, 37–39
 - indicator variable, 36

LENGTH attribute, 41–43
 multiple, 45–48, 53
 numeric variable, 40
 PROC MEANS, 39
 steps for creating a variable, 35–37
 TRUE and FALSE (logical expressions), 39–41
 variable attributes, 35
 Implicit loops, 95–96
 DATA step execution, 96
 example, 95
 implicit OUTPUT statement, 96
 RANUNI function, 96
 Increment argument, 188
 Increment value, 98
 Index-variable, 98, 102
 Informat argument, 200
 Input and output, *see* Data input and output
 Interval argument, 188

L

Line pointer-control, 162
 List input method, 166
 List output method, 177
 Longitudinal data, 79, 85
 Loops, *see* Writing loops in DATA step

M

Master data set, 151
 Match-merging, 147
 Mathematical functions, 184

N

Named input method, 156, 157
 Nested loops, 102, 134
 Numeric variables, 2

O

OPTIONS procedure, use of to modify
 SAS system options, 236–239
 LINESIZE option, 238
 listing output, 237
 options, 237, 238
 page breaks, 237
 settings, 237

P

PDV, *see* Program data vector
 Pointer-control, 162
 Procedures, *see* SAS® procedures
 Program data vector (PDV), 57, 141, 160

R

RACE variable, 30
 Reading text files, 160–174
 ampersand format modifier, 168
 character values, 161
 column input, 161–162
 column pointer-control, 162
 consecutive blanks, 168
 creating observations by using
 line-hold specifiers, 172–174
 creating observations by using line
 pointer-controls, 171–172
 data execution, 165
 data in free format, 169
 Delimiter-Sensitive-Data option, 167
 double trailing “at” signs, 172, 173
 embedded blanks, 168
 example, 163, 171
 formatted input, 162–164
 INFILE statement, 161, 172
 informat in modified list input, 169
 input buffer, 174
 input pointer, 171, 173
 INPUT statement, 164
 input values, 161
 line pointer-control, 162
 list input, 164–168
 missing values, 167
 mixed input, 170–171
 modified list input, 168–170
 nonstandard numeric value, 170
 pointer-control, 162
 program data vector, 160
 subsetting IF statement, 174
 syntax, modified list input, 168

S

SAS®, introduction to, 1–34
 base SAS procedures, 13–24
 BY variable, 14

- CLASS statement, 18
- common statements in SAS
 - procedures (TITLE, BY, and WHERE statements), 13–14
- CONTENTS procedure, 14–16
- default title, 13
- descriptor, 14
- FREQ procedure, 20–24
- HEARING data set, 15
- MEANS procedure, 18–20
- PRINT procedure, 16–18
- RACE variable, 17
- SMOKE variable, 19
- SORT procedure, 16
- TABLES statement, 20
- VARNUM option, 15
- VAR statement, 17
- WORK library, 14
- changing the appearance of data, 28–33
 - category of ethnicity, 30
 - character data, 31
 - dissociation of format from variable, 32
 - formatting variable values using SAS FORMATS, 31–33
 - labeling variables, 29–31
 - nonstandard numeric format, 32
 - preferred format, 28
 - RACE variable, 30
 - removal of labels from variables, 29
- creating and modifying variables, 9–13
 - assignment statement and SAS expression, 9–12
 - creating variables conditionally, 12–13
 - expression categories, 11
 - function, 11
 - IF-THEN/ELSE statements, 9, 12
 - INCOME variable, 13
 - mnemonic-equivalent forms, 9
 - PRINT procedure, 12
- exercises, 33–34
- reading data into SAS, 2–9
 - character variables, 2
 - floating-point numbers, 2
 - INFILE statement, 7
 - input data set, 4
 - input methods, 7
 - library, 2
 - message component, 4
 - naming rules, 3
 - numeric variables, 2
 - output data set, 4
 - reading data entered directly into program, 8–9
 - reading a raw data file with fixed fields, 6–8
 - reading a SAS data set, 3–6
 - rows, 2
 - SAS data set and SAS library, 2–3
 - SAS log, 5
 - terminologies, 2
 - text file, 6
- SAS program and language, 1
 - global statements, 1
 - language elements, 1
 - PROC steps, 1
- subsetting data by selecting variables, 24–28
 - DROP= data set option or DROP statement, 27
 - KEEP= data set option or KEEP statement, 25–26
 - name prefix lists, 24
 - numbered range lists, 24
 - variable-list notation, 25
 - variable-lists, types of, 24
 - where to specify DROP= and KEEP= data set options and DROP/KEEP statements, 27–28
- SAS® procedures, 205–239
 - creating user-defined format using FORMAT procedure, 227–236
 - catalog-specification, 233
 - continuous variables, 233
 - creating user-defined formats, 228–233
 - creating variables by using user-defined formats, 235–236
 - examples of value-or-range sets, 230
 - format catalog, 231
 - formatted-value, 229
 - HEARING data set, 233
 - indicator variable, 235

- informats, 227
- keywords, 229
- retrieving user-defined formats, 233–235
- value-or-range, 229
- value-range-set, 228
- VALUE statement, 228
- WORK library, 233
- exercises, 239
- restructuring data sets using
 - TRANPOSE procedure, 215–227
 - CONTENTS procedure, 218
 - COPY statement, 219
 - duplicated records, 223
 - handling duplicated observations
 - using the LET option, 222–224
 - introduction to transposing BY groups, 219–220
 - observations, 216
 - RENAME suboption, 220
 - situations requiring two or more transpositions, 224–227
 - suffix, 217
 - syntax, 216
 - transposing entire data set, 216–218
 - uninformative name, 220
 - user-defined PREFIX, 222
 - variables, 216
 - where ID statement does not work
 - for transposing BY groups, 220–221
 - where ID statement is essential
 - for transposing BY groups, 221–222
- using COMPARE procedure to
 - compare contents of two data sets, 208–215
 - base data set, 214
 - common variables, 210
 - comparing observations with common ID values, 212–215
 - comparison data set, 214
 - first weight value, 209
 - information provided from PROC COMPARE, 209–212
 - matching variables, 209
 - observation differences, 210
 - options, 209
 - VAR statement, 212
- using OPTIONS procedure to
 - modify SAS system options, 236–239
 - LINESIZE option, 238
 - listing output, 237
 - options, 237, 238
 - page breaks, 237
 - settings, 237
- using SORT procedure to eliminate
 - duplicate observations, 205–208
 - adjacent observations, 207
 - data set, 205
 - eliminating duplicate observations, 207–208
 - eliminating observations with duplicate BY values, 205–207
 - input data set, 206
 - NODUPKEY, 205, 206
 - observations, 205
 - PROC SORT, 206
 - SORT procedure, 205
- Select-expression, 48, 49
- SORT procedure, use of to eliminate
 - duplicate observations, 205–208
 - adjacent observations, 207
 - data set, 205
 - eliminating duplicate observations, 207–208
 - eliminating observations with duplicate BY values, 205–207
 - input data set, 206
 - NODUPKEY, 205, 206
 - observations, 205
 - PROC SORT, 206
 - SORT procedure, 205
- Source argument, 197, 200, 201
- Start-from argument, 188, 189
- String argument, 194
- Syntax
 - array processing, 121, 126
 - character functions, 190
 - combining data sets, 139, 146
 - errors, DATA step operation, 58, 70
 - reading text files, 168
 - TRANPOSE procedure, 216

T

- Target argument, 197
- Text files, *see* Data input and output
- Transaction data set, 151
- TRANSPOSE procedure, restructuring
 - data sets using, 215–227
 - CONTENTS procedure, 218
 - COPY statement, 219
 - duplicated records, 223
 - handling duplicated observations
 - using the LET option, 222–224
 - introduction to transposing BY
 - groups, 219–220
 - observations, 216
 - RENAME suboption, 220
 - situations requiring two or more
 - transpositions, 224–227
 - suffix, 217
 - syntax, 216
 - transposing entire data set, 216–218
 - uninformative name, 220
 - user-defined PREFIX, 222
 - variables, 216
 - where ID statement does not work
 - for transposing BY groups, 220–221
 - where ID statement is essential
 - for transposing BY groups, 221–222

U

- User-defined format, creation of
 - using FORMAT procedure, 227–236
 - catalog-specification, 233
 - continuous variables, 233
 - creating user-defined formats, 228–233
 - creating variables by using user-defined formats, 235–236
 - examples of value-or-range sets, 230
 - format catalog, 231
 - formatted-value, 229
 - HEARING data set, 233
 - indicator variable, 235
 - informats, 227
 - keywords, 229

- retrieving user-defined formats, 233–235
- value-or-range, 229
- value-range-set, 228
- VALUE statement, 228
- WORK library, 233

V

Variables

- character, 2
- continuous, 233
- counter, 87
- creating and modifying, 9–13
 - assignment statement and SAS expression, 9–12
 - creating variables conditionally, 12–13
 - expression categories, 11
 - function, 11
 - IF-THEN/ELSE statements, 9, 12
 - INCOME variable, 13
 - mnemonic-equivalent forms, 9
 - PRINT procedure, 12
- cumulating, 85, 87
- dissociation of format from, 32
- grouping, 79
- indicator, 235
- lists, types of, 24
- matching, 209
- numeric, 2
- product of, 131
- removal of labels from, 29
- temporary, 104
- Variables, conditional creation of, 35–54
 - executing one of several statements, 45–52
 - AGEGROUP variable, 45
 - executing statements using
 - SELECT group, 48–52
 - multiple IF-THEN/ELSE statements, 45–48
 - OTHERWISE statement, 49
 - select-expression, 48, 49
 - threshold value, 46
 - when-expression, 49
- exercises, 54
- IF-THEN/ELSE statement, 35–45

- bytes of storage space, 42
 - character variable, comparison of, 38
 - DO group, 43–45
 - evaluation of variables, 35
 - FREQ procedure, 43
 - handling missing values when creating variables, 37–39
 - indicator variable, 36
 - LENGTH attribute, 41–43
 - numeric variable, 40
 - PROC MEANS, 39
 - steps for creating a variable, 35–37
 - TRUE and FALSE (logical expressions), 39–41
 - variable attributes, 35
 - modifying of IF-THEN/ELSE statement with assignment statement, 52–54
 - AGEGROUP variable, 53, 54
 - comparison evaluation, 53
 - indicator variable, 52
 - multiple IF-THEN/ELSE statement, 53
 - Vertically combining data sets, 139–143
 - common variables, 141
 - compilation phase, 141, 143
 - concatenating data sets, 139–142
 - execution phase, 141, 143
 - input data sets, sorting of, 142
 - interleaving data sets, 142–143
 - new-name, 141
 - old-name, 141
 - program data vector, 141, 143
 - sum of observations, 139
 - syntax, 139
 - variables from different source, 139
- W**
- When-expression, 49
 - WORK library, 14
 - Writing loops in DATA step, 95–119
 - exercises, 117–119
 - implicit and explicit loops, 95–103
 - CENTER variable, 103
 - combining loops, 103
 - DATA step execution, 96
 - DO loop, first two iterations, 99
 - example, 95
 - explicit loops, 96–102
 - expression, evaluated, 101
 - identical statements, 97
 - ID variable, 101
 - implicit loops, 95–96
 - implicit OUTPUT statement, 96
 - increment value, 98
 - index-variable, 98, 102
 - inner loop, 102
 - last two iterations, 100
 - nested loops, 102
 - number of iterations, 99
 - outer loop, 102
 - programming error, 101
 - RANUNI function, 96
 - start value, 98
 - stop value, 98
 - using looping to read a list of external files, 110–117
 - DO loop, first iteration, 114
 - DO loop, second iteration, 115
 - DO loop, third iteration, 116
 - DO UNTIL loop, 111
 - dummy, 112
 - end-of-file marker, 110, 111
 - explicit OUTPUT statement, 113
 - external file, 110–111
 - file-specification, 111
 - INFILE statement, 111
 - multiple external files, 111–117
 - placeholder, 111
 - reading multiple files, 112
 - specifying a condition, 110
 - variable, 110
 - utilizing loops to create samples, 103–109
 - algorithm, 108, 109
 - components, 104
 - CONTENTS procedure, 105
 - creating random sample with replacement, 106–108
 - creating random sample without replacement, 108–109
 - creating systematic sample, 105–106
 - data set, 103

- direct-access mode, 104–105
- DO loop, first iteration, 106
- increment values, 105
- input-data-set, 105
- last two iterations, 107
- observation to be selected, 108
- Rannum, 108
- sampling schemes, 103
- STOP statement, 104
- temporary variable, 104

