

An Introduction to SAS[®] University Edition

Ron Cody

Student Solutions



support.sas.com/bookstore

This set of Solutions for Students is a companion piece to the following SAS Press book: Cody, Ron. 2015. *An Introduction to SAS® University Edition*. Cary, NC: SAS Institute Inc.

An Introduction to SAS® University Edition

Copyright © 2015, SAS Institute Inc., Cary, NC, USA

ISBN 978-1-62959-770-6 (Hard copy)

ISBN 978-1-62960-007-9 (EPUB)

ISBN 978-1-62960-008-6 (MOBI)

ISBN 978-1-62960-010-9 (PDF)

All rights reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

September 2015

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

Appendix: Solutions to the Odd-Numbered Problems

Solution 8-1;

```
data Quick_Survey;
  infile '/folders/myfolders/Quick.txt';
  informat Subj $3.
           Gender $1.
           DOB mmddyy10.
           Income_Group $1.;
  input Subj
        Gender
        DOB
        Height
        Weight
        Income_Group;
  format DOB mmddyy10.;
run;
title "Listing of Data Set QUICK_SURVEY";
proc print data=Quick_Survey;
  id Subj;
run;

*Solution 8-3;
title "Frequencies";
proc freq data=Quick_Survey order=freq;
  tables Gender Income_Group / nocum;
run;
```

Solution 8-5;

```
data Quick_Survey;
  infile '/folders/myfolders/Quick.csv' dsd;
  informat Subj $3.
           Gender $1.
           DOB mmddyy10.
           Income_Group $1.;
  input Subj
        Gender
        DOB
        Height
        Weight
        Income_Group;
```

```

format DOB mmddyy10.;
run;
title "Listing of Data Set QUICK_SURVEY";
proc print data=Quick_Survey;
  id Subj;
run;

```

Solution 8-7;

```

/** Import an XLSX file.  **/

PROC IMPORT DATAFILE="/folders/myfolders/Grades.xlsx"
  OUT=WORK.MYEXCEL
  DBMS=XLSX
  REPLACE;

RUN;
*Comment removed from the next line;
PROC PRINT DATA=WORK.MYEXCEL; RUN;

```

Solution 8-9;

```

data Formatted;
  infile '/folders/myfolders/Quick_Cols.txt' pad;
  input @1 Subj      $3
        @4 Gender   $1.
        @5 DOB      mmddyy10.
        @15 Height  2.
        @17 Weight  3.
        @20 Income_Group $1.;
  format DOB mmddyy10.;
run;
title "Listing of Data Set Formatted";
proc print data=Formatted noobs;
run;

```

Solution 9-1;

```

title "PROC CONTENTS for SASHELP.HEART";
proc contents data=sashelp.Heart;
run;

title "PROC CONTENTS with the VARNUM option";
proc contents data=sashelp.Heart VARNUM;
run;

*Solution 9-3;
libname easy '/folders/myfolders';
data easy.Heart_Vars;
  set sashelp.heart(keep=BP_Status Chol_Status Systolic Diastolic
  Status);
run;

```

Solution 9-5;

```
libname sasdata '/folders/myshortcuts/sasdata';
data sasdata.Young_Males;
  set sashelp.class(where=(Sex = 'M' and Age in (11 12)));
run;
```

Solution 10-1;

```
proc format;
  value Gender 1='Male' 2='Female';
  value $Ques '1'='Strongly Disagree' '2'='Disagree' '3'='No opinion'
             '4'='Agree' '5'='Strongly Agree';
  value AgeGrp 0-20='Young' 21-40='Still Young' 41-60='Middle'
             61-high='Older';
run;

data Questionnaire;
  informat Gender 1. Q1-Q4 $1. Visit date9.;
  input Gender Q1-Q4 Visit Age;
  format Gender gender. Q1-Q4 $Ques. Visit mmdyy10. Age AgeGrp.;
datalines;
1 3 4 1 2 29May2015 16
1 5 5 4 3 01Sep2015 25
2 2 2 1 3 04Jul2014 45
2 3 3 3 4 07Feb2015 65
;
title "Listing of Data Set QUESTIONNAIRE";
proc print data=Questionnaire noobs;
run;
```

Solution 10-3;

```
proc format;
  value $Grades 'A','B' = 'Good'
              'C'      = 'Average'
              'D'      = 'Poor'
              'F'      = 'Fail'
              'I'      = 'Incomplete'
              ' '      = 'Missing'
              Other    = 'Invalid';
run;
```

Solution 11-1;

```
data Group_Fish;
  set SASHELP.Fish(keep=Species Weight Height);
  if missing(Weight) then Fish_Grp = .;
/* Alternative:
  if Weight = . then Fish_Grp = .;
*/

  else if Weight le 100 then Fish_Grp = 1;
```

```

        else if Weight le 200 then Fish_Grp = 2;
        else if Weight le 500 then Fish_Grp = 3;
        else if Weight le 1000 then Fish_Grp = 4;
        else if Weight ge 1001 then Fish_Grp = 5;
run;
title "Listing of first 10 Observations in Group_Fish";
proc print data=Group_Fish(obs=10) noobs;
run;
*Solution 11-3;
data High_BP;
    set sashelp.Heart(keep=Diastolic Systolic Status);
    if Systolic gt 250 or Diastolic gt 180;
run;
title "Listing of High_BP";
proc print data=High_BP noobs;
run;

```

Solution 11-5;

```

/*
1. data Weights;
2.     input Wt;
3.     if Wt lt 100 then Wt_Group = 1;
Missing values will be in Wt_Group 1

4.     if Wt lt 200 then Wt_Group = 2;
Should be Else if

5.     if Wt lt 300 then Wt_Group = 3;
Should be Else if
6. datalines;
50
150
250
;
*/
data Weights;
    input Wt;
    if missing(Wt) then Wt_Group = .;
    else if Wt lt 100 then Wt_Group = 1;
    else if Wt lt 200 then Wt_Group = 2;
    else if Wt lt 300 then Wt_Group = 3;
datalines;
50
150
250
;
title "Listing of Weights";
proc print data=Weights noobs;
run;

```

Solution 12-1;

```

data Wt_Convert;
  do Pounds = 0 to 100 by 10;
    Kg = Pounds/2.2;
    output;
  end;
run;

title "Weight Conversion Table";
proc print data=Wt_Convert noobs;
run;

```

Solution 12-3;

```

data Study;
  do Group = 'A', 'B', 'C';
    input Score;
    output;
  end;
datalines;
10
11
12
20
21
22
;
title "Listing of Study";
proc print data=Study noobs;
run;

```

Solution 12-5;

```

data Interest;
  Money = 100;
  do until (Money gt 200);
    Year + 1;
    Money = Money + .03*Money;
    output;
  end;
run;

title "Listing of Interest";
proc print data=Interest noobs;
run;

```

Solution 13-1;

```

data Read_Dates;
  input @1 Date1 mmddyy10.
        @12 Date2 date9.;

```

```

format Date1 Date2 mmddyy10.;
datalines;
10/21/2015 12Jun2015
12/25/2015 9Apr2014
;
title "Listing of Dates";
proc print data=Read_Dates;
run;

```

Solution 13-3;

```

data Dates;
  set sashelp.Retail(keep=Month Day Year);
  SAS_Date = mdy(Month,Day,Year);
  format SAS_Date mmddyy10.;
run;
title "Listing of Dates";
proc print data=Dates(obs=5) noobs;
run;

```

Solution 13-5;

```

data Study;
  call streaminit(13579);
  do Subj = 1 to 10;
    Date = '01Jan2015'd + int(rand('uniform')*300);
    output;
  end;
  format Date date9.;
run;

title "Out of Range Dates";
data _null_;
  set Study;
  where Date lt '01Jan2015'd or Date gt '04Jul2015'd;
  file print; *Send output to Result window;
  put Subj= Date=;
run;

```

Solution 14-1;

```

data Small_Perch;
  set SASHELP.Fish;
  where Species = 'Perch' and Weight lt 50;
run;
title "Listing of Small Perch";
proc print data=Small_Perch noobs;
run;

```


Solution 14-3;

```

data Questionnaire;
  informat Gender 1. Q1-Q4 $1. Visit date9.;
  input Gender Q1-Q4 Visit Age;
  if sum(of Q1-Q3) ge 6;
  format Viit date9.;
datalines;
1 3 4 1 2 29May2015 16
1 5 5 4 3 01Sep2015 25
2 2 2 1 3 04Jul2014 45
2 3 3 3 4 07Feb2015 65
;

title "Listing of Data Set QUESTIONNAIRE";
proc print data=Questionnaire noobs;
run;

```

Solution 14-5;

```

data FirstQtr;
  input Name $ Quantity Cost;
datalines;
Fred 100 3000
Jane 90 4000
April 120 5000
;
data SecondQtr;
  input Name $ Quantity Cost;
datalines;
Ron 200 9000
Jan 210 9500
Steve 177 5400
;
data FirstHalf;
  set FirstQtr SecondQtr;
run;
title "Listing of Data Set FirstHalf";
proc print data=FirstHalf noobs;
run;

```

Solution 14-7;

```

data First;
  input ID $ X Y Z;
datalines;
001 1 2 3
004 3 4 5
002 5 7 8
006 8 9 6
;
data Second;

```

```

        input ID $ Name $;
datalines;
002 Jim
003 Fred
001 Susan
004 Jane
;
proc sort data=First;
    by ID;
run;
proc sort data=Second;
    by Id;
run;

data Both;
    merge First(in=In_One) Second(in=In_Two);
    by ID;
    if In_One and In_Two;
run;
title "Listing of Data Set Both";
proc print data=Both noobs;
run;

```

Solution 14-9;

```

data Prices;
    input Item_Number $ Price;
datalines;
A123 $123
B76 4.56
X200 400
D88 39.75
;

data New;
    input Item_Number $ Price;
datalines;
X200 410
A123 121
;
proc sort data=Prices;
    by Item_Number;
run;
proc sort data=New;
    by Item_Number;
run;

data New_Prices;
    update Prices New;
    by Item_Number;
run;

```

```

title "Listing of New_Prices";
proc print data=New_Prices noobs;
run;

```

Solution 15-1;

```

data Questionnaire2;
  input Subj $ Q1-Q20;
datalines;
001 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
002 . . . . 3 2 3 1 2 3 4 3 2 3 4 3 5 4 4 4
003 1 2 1 2 1 2 12 3 2 3 . . . . . 4 5 5 4
004 1 4 3 4 5 . 4 5 4 3 . . 1 1 1 1 1 1 1 1
;

data Score_Quest;
  set Questionnaire2;
  if n(of Q1-Q10) ge 7 then Score1 = mean(of Q1-Q10);
  if nmiss(of Q11-Q20) le 5 then Score2 = median(of Q11-Q20);
  Score3 = max(Q1-Q10);
  Score4 = sum (largest(1,of Q1-Q10), largest(2,of Q1-Q10));
  drop Q1-Q20;
run;

title "Listing of Data Set Score_Quest";
proc print data=Score_Quest noobs;
run;

```

Solution 15-3;

```

data Char_Data;
  length Date $10 Weight Height $ 3;
  input Date Weight Height;
datalines;
10/21/1966 220 72
5/6/2000 110 63
;
data Num_Data;
  set Char_Data(rename=(Date=C_Date Weight=C_Weight
Height=C_Height));
  Date = input(C_date,mmdyy10.);
  Weight=input(C_Weight,12.);
  Height = input(C_Height,12.);
  format Date date9.;
  drop C_;;
  *Note: The colon in the DROP statement says to drop all variables
that start with C_. The colon is like a wild-card and says to
reference all the variables with the same beginning characters;
run;
title "Listing of Data Set Num_Data";
proc print data=Num_Data noobs;
run;

```

Solution 15-5;

```

data Oscar;
  length String $ 10 Name $ 20 Comment $ 25 Address $ 30
         Q1-Q5 $ 1;
  infile datalines dsd dlm=" ";
  *Note: the DSD option is needed to strip the quotes from
  the variables that contain blanks;
  input String Name Comment Address Q1-Q5;
  L1 = lengthn(String);
  L2 = lengthc(String);
datalines;
AbC "jane E. MarPle" "Good Bad Bad Good" "25 River Road" y n N Y Y
12345 "Ron Cody" "Good Bad Ugly" "123 First Street" N n n n N
98x "Linda Y. d'amore" "No Comment" "1600 Penn Avenue" Y Y y y y
. "First Middle Last" . "21B Baker St." . . . Y N
;
title "Listing of Selected Variables from Data Set Oscar";
proc print data=Oscar noobs;
  var String L1 L2;
run;

```

Solution 15-7;

```

data Oscar;
  length String $ 10 Name $ 20 Comment $ 25 Address $ 30
         Q1-Q5 $ 1;
  length Two_Three $ 2;
  infile datalines dsd dlm=" ";
  *Note: the DSD option is needed to strip the quotes from
  the variables that contain blanks;
  input String Name Comment Address Q1-Q5;
  Two_Three = substrn(String,2,2);
datalines;
AbC "jane E. MarPle" "Good Bad Bad Good" "25 River Road" y n N Y Y
12345 "Ron Cody" "Good Bad Ugly" "123 First Street" N n n n N
98x "Linda Y. d'amore" "No Comment" "1600 Penn Avenue" Y Y y y y
. "First Middle Last" . "21B Baker St." . . . Y N
;

title "Listing of Selected Variables from Oscar";
proc print data=Oscar noobs;
  var String Two_Three;
run;

```

Solution 15-9;

```

Data How_Tall;
  input Ht $ @@;
  *Note: the @@ at the end of the INPUT statement allows you
  to place several observations on one line of data;
  Height = input(compress(Ht,,'kd'),12.);

```

```

        if find(Ht,'cm','i') then Height = Height/2.54;
datalines;
65inches 200cm 70In. 220Cm. 72INCHES
;
title "Listing of Data Set How_Tall";
proc print data=How_Tall noobs;
run;

```

Solution 15-11;

```

data Oscar;
  length String $ 10 Name $ 20 Comment $ 25 Address $ 30
         Q1-Q5 $ 1;
  infile datalines dsd dlm=" ";
*Note: the DSD option is needed to strip the quotes from
the variables that contain blanks;
  input String Name Comment Address Q1-Q5;
  Name = propcase(Name," ");
  Address = tranwrd(Address,'Street','St. ');
  Address = tranwrd(Address,'Road','Rd. ');
  Address = tranwrd(Address,'Avenue','Ave. ');
  Last_Name = scan(Name,-1,' ');
datalines;
AbC "jane E. MarPle" "Good Bad Bad Good" "25 River Road" y n N Y Y
12345 "Ron Cody" "Good Bad Ugly" "123 First Street" N n n n N
98x "Linda Y. d'amore" "No Comment" "1600 Penn Avenue" Y Y y y y
. "First Middle Last" . "21B Baker St." . . . Y N
;
title "Selected Variables from Data Set Oscar";
proc print data=Oscar noobs;
  var Address;
run;

```

Solution 16-1;

```

data Clinic;
  informat Date mmddyy10. Subj $3.;
  input Subj Date Heart_Rate Weight;
  format Date date9.;
datalines;
001 10/1/2015 68 150
003 6/25/2015 75 185
001 12/4/2015 66 148
001 11/5/2015 72 152
002 1/1/2014 75 120
003 4/25/2015 80 200
003 5/25/2015 78 190
003 8/20/2015 70 179
;
proc sort data=Clinic;
  by Subj Date;
run;

```

```

data Diff;
  set Clinic;
  by Subj;
  if first.Subj and last.Subj then delete;
  Diff_HR = Heart_Rate - lag(Heart_Rate);
  *Alternative: Diff_HR = dif(Heart_Rate);
  Diff_Weight = dif(Weight);
  if not first.Subj then output;
run;
title "Listing of Data Set Clinic";
proc print data=Diff noobs;
run;

```

Solution 16-3;

```

* Observation Last_x
  1          .
  2          6
  3          .
  4          7
  5         10;

```

```

*Solution 17-1;

```

```

data Probl;
  length Char1-Char5 $ 8;
  input x1-x5 Char1-Char5;
  array x[5] x1-x5;
  array Char[5] Char1-Char5;
  *No need for $ in this array statement because Char1-Char5
  already declared character with a length of 8;
  do i = 1 to 5;
    x[i] = round(x[i]);
    Char[i] = upcase(Char[i]);
  end;
  drop i;
datalines;
1.2 3 4.4 4.9 5 a b c d e
1.01 1.5 1.6 1.7 1.8 frank john mary jane susan
;
title "Listing of Data Set Probl";
proc print data=Probl noobs;
run;

```

Solution 17-3;

```

data Missing;
  input w x y z C1 $ C2 $ C3 $;
  array Allnums[*] x y z;
  array Allchars[*] C1-C3;
  do i = 1 to dim(Allnums);
    if Allnums[i] = 999 then Allnums[i] = .;
  end;

```

```

do i = 1 to dim(Allchars);
  if find(Allchars[i], 'NA', 'i') then Allchars[i] = ' ';
end;
drop i;
datalines;
999 1 999 3 Fred NA Jane
8 999 10 20 Michelle Mike John
11 9 8 7 NA na Peter
;
title "Listing of Data Set Missing";
proc print data=Missing noobs;
run;

```

Solution 18-1;

```

title "Listing of the First 10 Observations in Data Set Fish";
title2 "Prepared by: Ron Cody";
title3 "-----";

proc print data=SASHELP.Fish(Obs=10 drop=Length1-Length3);
  id Species;
run;

*Solution 18-3;
proc sort data=SASHELP.Fish out=Fish;
  by Species;
run;

title "Listing of Fish Broken Down by Species";
proc print data=Fish(drop=Length1-Length3);
  by Species;
  id Species;
run;

```

Solution 19-1;

```

title "Statistics for Height and Weight in the Heart Data Set";
proc means data=SASHELP.Heart n nmiss mean std min max maxdec=2;
  var Height Weight;
run;

*Solution 19-3;
title "Statistics for Height and Weight in the Heart Data Set";
proc means data=SASHELP.Heart n nmiss mean std min max maxdec=2;
  class Status;
  var Height Weight;
run;

```

Solution 19-5;

```
proc means data=SASHELP.Heart n nmiss mean std min max maxdec=2
      noprint nway;
  var Height Weight;
  output out=Summary mean= n= nmiss= std= min= max= / autoname;
run;
title "Listing of Data Set Summary";
proc print data=Summary noobs;
run;
```

Solution 19-7;

```
title "PROC UNIVARIATE Statistics for Height and Weight";
proc univariate data=SASHELP.Heart;
  var Height Weight;
  histogram;
run;
```

Solution 20-1;

```
title "Summary Data from SASHELP Heart Data Set";
proc freq data=SASHELP.Heart;
  tables Status BP_Status Smoking_Status / nocum;
run;
```

Solution 20-3;

```
proc format;
  value $Status 'Dead' = '1-Dead'
               'Alive' = '2-Alive';
run;
title "Summary Data from SASHELP Heart Data Set";
proc freq data=SASHELP.Heart order=formatted;
  format Status $Status.;
  tables Status;
run;
```

Solution 20-5;

```
title "Summary Data from SASHELP Heart Data Set";
proc freq data=SASHELP.Heart (where=(Weight_Status ne 'Underweight'));
  tables Sex*Weight_Status*Status / chisq;
run;
```