1. Given the following data:

X Y Z

1 3 15

7 13 7

8 12 5

3 4 14

4 7 10

1. Write a SAS program to compute the Pearson correlation coefficient between X and Y; x and Z. What is the significance of each?
2. Chane the correlation request to produce a correlation matrix; that is, the correlation coefficient between each variable versus every other variable.

Code:

**DATA** Cor;

INPUT X Y Z;

DATALINES;

1 3 15

7 13 7

8 12 5

3 4 14

4 7 10

;

**run**;

/\*proc print data=Pearson;

run;\*/

Title 'Pearson Correlation of X with Y & Z';

**proc** **corr** data=Cor outp=corr;

var X;

with Y Z;

**run**;

Title;

Title 'Correlation Matrix';

**proc** **print** data=corr;

**run**;

Title;

Log:

341 DATA Cor;

342 INPUT X Y Z;

343 DATALINES;

NOTE: The data set WORK.COR has 5 observations and 3 variables.

NOTE: DATA statement used (Total process time):

real time 0.01 seconds

cpu time 0.01 seconds

349 ;

350 run;

351

352 /\*proc print data=Pearson;

353 run;\*/

354

355 Title 'Pearson Correlation of X with Y & Z';

356 proc corr data=Cor outp=corr;

357 var X;

358 with Y Z;

359 run;

NOTE: Writing HTML Body file: sashtml1.htm

NOTE: The data set WORK.CORR has 5 observations and 3 variables.

NOTE: PROCEDURE CORR used (Total process time):

real time 0.84 seconds

cpu time 0.46 seconds

360 Title;

361

362 Title 'Correlation Matrix';

363 proc print data=corr;

364 run;

NOTE: There were 5 observations read from the data set WORK.CORR.

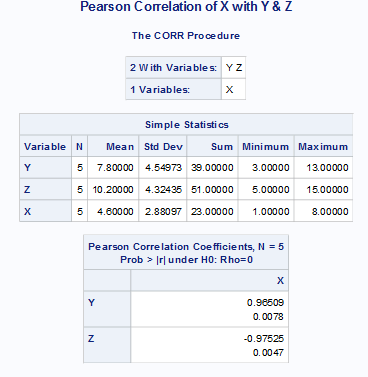
NOTE: PROCEDURE PRINT used (Total process time):

real time 0.11 seconds

cpu time 0.01 seconds

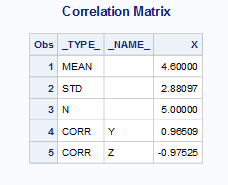
365 Title;

Output:



X and Y have a correlation of 0.96509 , with Y increasing as X increases.

X and Z have a correlation of -0.97525, with Z decreasing as X increases.



2 Given the following data:

AGE SBP

15 116

20 120

25 130

30 132

40 150

50 148

How much of the variance of SBP (systolic blood pressure) can be explained by the fact that there is variability in AGE? (Use SAS to compute the correlation between SBP and AGE.)

Code:

**DATA** Cor\_age\_sbp;

INPUT AGE SBP;

DATALINES;

15 116

20 120

25 130

30 132

40 150

50 148

;

**run**;

Title 'Correlation of Age with SBP';

**proc** **corr** data=Cor\_age\_sbp;

var AGE;

with SBP;

**run**;

Title;

Log:

74 DATA Cor\_age\_sbp;

75 INPUT AGE SBP;

76 DATALINES;

NOTE: The data set WORK.COR\_AGE\_SBP has 6 observations and 2 variables.

NOTE: DATA statement used (Total process time):

real time 0.02 seconds

cpu time 0.01 seconds

83 ;

84 run;

85

86 Title 'Correlation of Age with SBP';

87 proc corr data=Cor\_age\_sbp;

88 var AGE;

89 with SBP;

90 run;

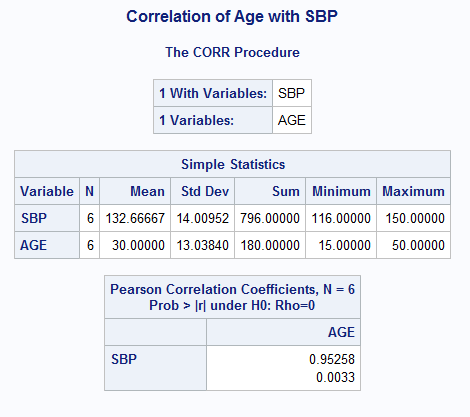
NOTE: PROCEDURE CORR used (Total process time):

real time 0.07 seconds

cpu time 0.03 seconds

91 Title;

Output:



Age and SBP are correlated 0.95258. As age increases systolic blood pressure increases.

3 From the data for X and Y below:

X Y Z

1 3 15

7 13 7

8 12 5

3 4 14

4 7 10

1. Compute a regression line (Y on X). Y is the dependent variable, X the independent variable.
2. What is slope and intercept?
3. Are they significantly different from zero?

Code:

**data** linear\_reg;

Input X Y Z;

Datalines;

1 3 15

7 13 7

8 12 5

3 4 14

4 7 10

;

**run**;

**proc** **reg** data=linear\_reg;

model Y=X;

test1: test X =**0**;

**run**;

Log:

384 data linear\_reg;

385 Input X Y Z;

386 Datalines;

NOTE: The data set WORK.LINEAR\_REG has 5 observations and 3 variables.

NOTE: DATA statement used (Total process time):

real time 0.01 seconds

cpu time 0.01 seconds

392 ;

393 run;

394

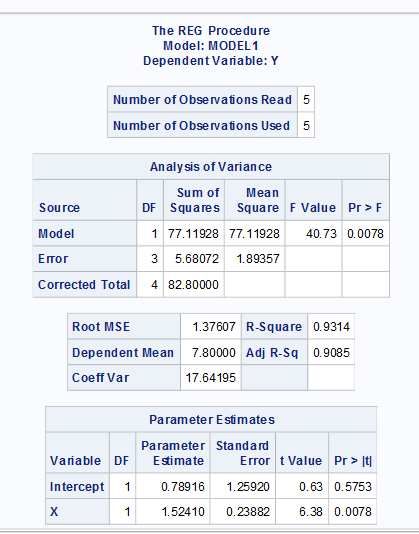
395 proc reg data=linear\_reg;

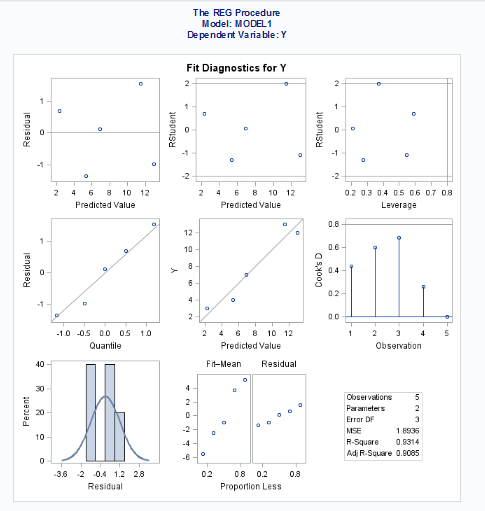
396 model Y=X;

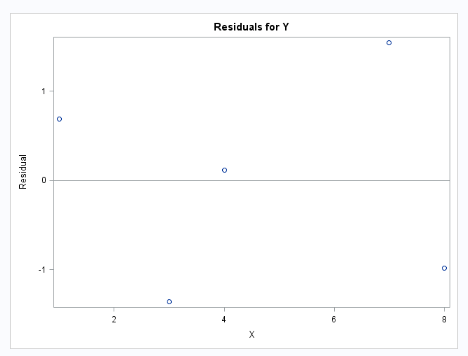
397 test1: test X =0;

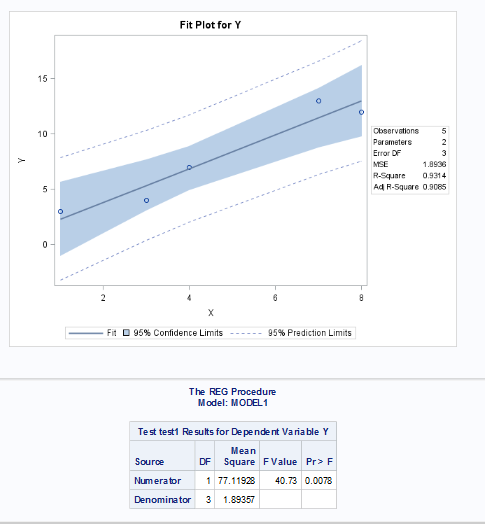
398 run;

Output:









Slope =0.78916

Intercept =1.52410

4 Using the data below compute 3 new variables LX, LY and LZ, which are the natural logarithms of the original values. Compute a correlation matrix for the three new variables. HINT: The function to compute a natural log is the LOG function.

X Y Z

1 3 15

7 13 7

8 12 5

3 4 14

4 7 10

Code:

**data** XYZ;

Input X Y Z;

Datalines;

1 3 15

7 13 7

8 12 5

3 4 14

4 7 10

;

**run**;

**data** log\_cor;

set XYZ;

LX=log(X);

LY=log(Y);

LZ=log(Z);

**run**;

Title 'Correlation of LX with LY LZ';

**proc** **corr** data=log\_cor outp=log\_corr;

var LX;

with LY LZ;

**run**;

Title;

**proc** **print** data=log\_corr;

**run**;

Log:

683 data log\_cor;

684 set XYZ;

685 LX=log(X);

686 LY=log(Y);

687 LZ=log(Z);

688 run;

NOTE: There were 5 observations read from the data set WORK.XYZ.

NOTE: The data set WORK.LOG\_COR has 5 observations and 6 variables.

NOTE: DATA statement used (Total process time):

real time 0.02 seconds

cpu time 0.03 seconds

689

690

691

692 Title 'Correlation of LX with LY LZ';

693 proc corr data=log\_cor outp=log\_corr;

694 var LX;

695 with LY LZ;

696 run;

NOTE: The data set WORK.LOG\_CORR has 5 observations and 3 variables.

NOTE: PROCEDURE CORR used (Total process time):

real time 0.12 seconds

cpu time 0.04 seconds

697 Title;

698

699 Title 'XYZ';

700 proc print data=log\_corr;

701 run;

NOTE: There were 5 observations read from the data set WORK.LOG\_CORR.

NOTE: PROCEDURE PRINT used (Total process time):

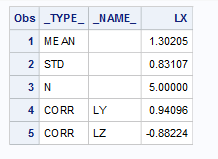
real time 0.12 seconds

cpu time 0.04 seconds

702 Title;

Output:





1. Using the data below generate:

X Y Z

1 3 15

7 13 7

8 12 5

3 4 14

4 7 10

1. A plot of Y versus X
2. A plot of the regression line and the original data on the same set of axes.

Code:

**data** YXZ;

Input X Y Z;

Datalines;

1 3 15

7 13 7

8 12 5

3 4 14

4 7 10

;

**run**;

\*symbol value=dot line=1 width=2;

\*options linesize=70 pagesize=35;

**proc** **gplot** data=YXZ;

plot y\*x;

**run**;

**proc** **reg** data=YXZ;

model Y=X;

plot Y\*X;

**run**;

**quit**;

Log:

429 data YXZ;

430 Input X Y Z;

431 Datalines;

NOTE: The data set WORK.YXZ has 5 observations and 3 variables.

NOTE: DATA statement used (Total process time):

real time 0.01 seconds

cpu time 0.01 seconds

437 ;

438 run;

439 \*symbol value=dot line=1 width=2;

440 \*options linesize=70 pagesize=35;

441

442 proc gplot data=YXZ;

443 plot y\*x;

444 run;

NOTE: 8289 bytes written to C:\Users\Samil\AppData\Local\Temp\SAS

Temporary Files\\_TD14624\_DESKTOP-H3OLVE7\_\gplot6.png.

445

NOTE: There were 5 observations read from the data set WORK.YXZ.

NOTE: PROCEDURE GPLOT used (Total process time):

real time 0.25 seconds

cpu time 0.15 seconds

446 proc reg data=YXZ;

447 model Y=X;

448 plot Y\*X;

449 run;

NOTE: 16181 bytes written to C:\Users\Samil\AppData\Local\Temp\SAS

Temporary Files\\_TD14624\_DESKTOP-H3OLVE7\_\reg3.png.

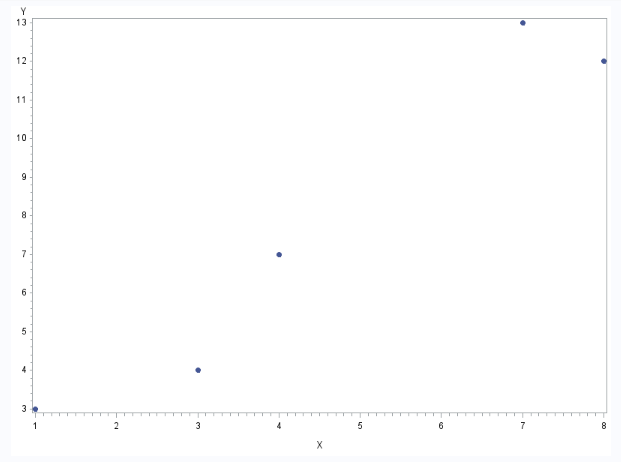
450 quit;

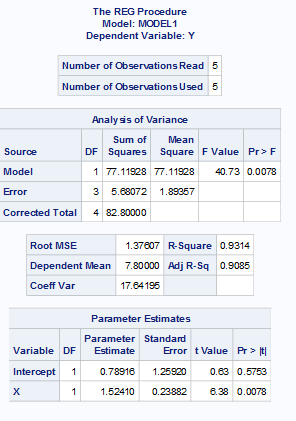
NOTE: PROCEDURE REG used (Total process time):

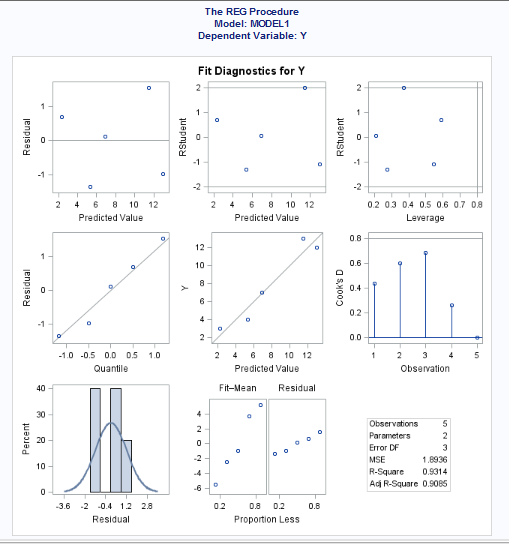
real time 1.74 seconds

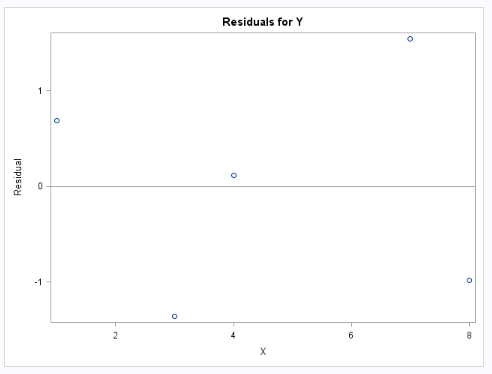
cpu time 0.65 seconds

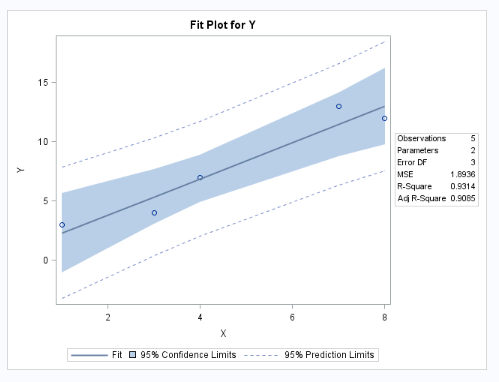
Output:

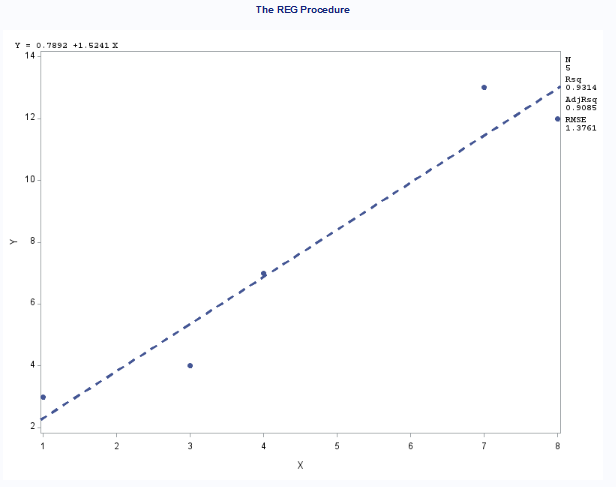












6 Given the data set (variables are COUNTY POP HOSPITAL FIRE\_CO RURAL)

1 35 1 2 YES

2 88 5 8 NO

3 5 0 1 YES

4 55 3 3 YES

5 75 4 5 NO

6 125 5 8 NO

7 225 7 9 YES

8 500 10 11 NO

1. Write a SAS program to create a SAS data set of the previous data.
2. Run PROC UNIVARIATE to check the distributions for the variables POP, HOSPITAL and FIRE\_CO.
3. Compute a correlation matrix for the variables POP, HOSPITAL, and FIRE\_CO. Produce both Pearson and Spearman correlations.
4. Recode POP, HOSPITAL, and FIRE\_CO so that they each have two levels (use a median cut or a value somewhere near the 50th percentile). Compute crosstabulations between the variable RURAL and the recoded variables.

Code:

**data** County ;

input COUNTY POP HOSPITAL FIRE\_CO RURAL $;

datalines;

1 35 1 2 YES

2 88 5 8 NO

3 5 0 1 YES

4 55 3 3 YES

5 75 4 5 NO

6 125 5 8 NO

7 225 7 9 YES

8 500 10 11 NO

;

**run**;

Title 'County';

**proc** **print** data=County;

**run**;

Title;

Title 'Pearson and Spearman Correlation ';

**proc** **corr** data=County spearman Pearson ;

var POP;

with HOSPITAL FIRE\_CO;

**run**;

Title;

**proc** **univariate** data=County;

var POP HOSPITAL FIRE\_CO;

**run**;

**proc** **format**;

value po low-**81.5**="G1"

**81.6**-High="G2";

value Hos low-**4.5**="G1"

**4.6**-High="G2";

value Fire low-**6.5**="G1"

**6.6**-High="G2";

**run**;

**proc** **freq** data=County;

tables Rural\*(POP HOSPITAL FIRE\_CO);

format POP po. HOSPITAL Hos. FIRE\_CO Fire.;

**run**;

Log:

492 data County ;

493 input COUNTY POP HOSPITAL FIRE\_CO RURAL $;

494 datalines;

NOTE: The data set WORK.COUNTY has 8 observations and 5 variables.

NOTE: DATA statement used (Total process time):

real time 0.07 seconds

cpu time 0.03 seconds

503 ;

504 run;

505

506 Title 'County';

507 proc print data=County;

508 run;

NOTE: There were 8 observations read from the data set WORK.COUNTY.

NOTE: PROCEDURE PRINT used (Total process time):

real time 0.34 seconds

cpu time 0.01 seconds

509 Title;

510

511 Title 'Pearson and Spearman Correlation ';

512 proc corr data=County spearman Pearson ;

513 var POP;

514 with HOSPITAL FIRE\_CO;

515 run;

NOTE: PROCEDURE CORR used (Total process time):

real time 0.13 seconds

cpu time 0.06 seconds

516 Title;

517

518 proc univariate data=County;

519 var POP HOSPITAL FIRE\_CO;

520 run;

NOTE: PROCEDURE UNIVARIATE used (Total process time):

real time 0.18 seconds

cpu time 0.09 seconds

521

522 proc format;

523 value po low-81.5="G1"

524 81.6-High="G2";

NOTE: Format PO is already on the library WORK.FORMATS.

NOTE: Format PO has been output.

525 value Hos low-4.5="G1"

526 4.6-High="G2";

NOTE: Format HOS is already on the library WORK.FORMATS.

NOTE: Format HOS has been output.

527 value Fire low-6.5="G1"

528 6.6-High="G2";

NOTE: Format FIRE is already on the library WORK.FORMATS.

NOTE: Format FIRE has been output.

529 run;

NOTE: PROCEDURE FORMAT used (Total process time):

real time 0.02 seconds

cpu time 0.03 seconds

530

531 proc freq data=County;

532 tables Rural\*(POP HOSPITAL FIRE\_CO);

533 format POP po. HOSPITAL Hos. FIRE\_CO Fire.;

534 run;

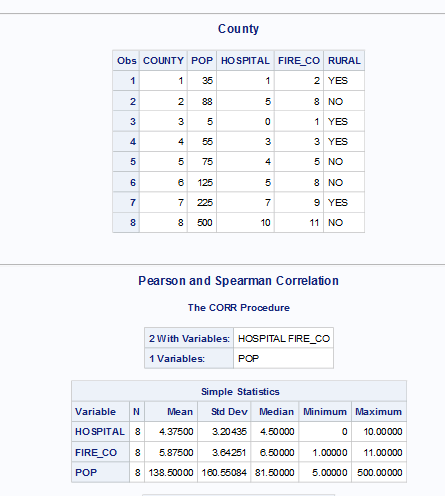
NOTE: There were 8 observations read from the data set WORK.COUNTY.

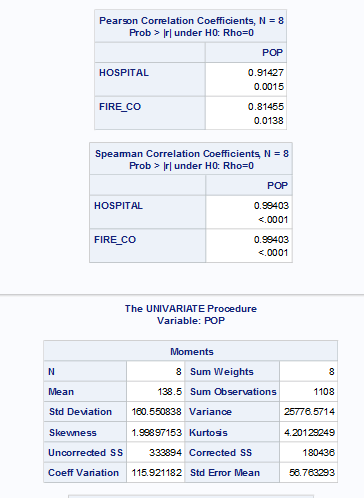
NOTE: PROCEDURE FREQ used (Total process time):

real time 0.23 seconds

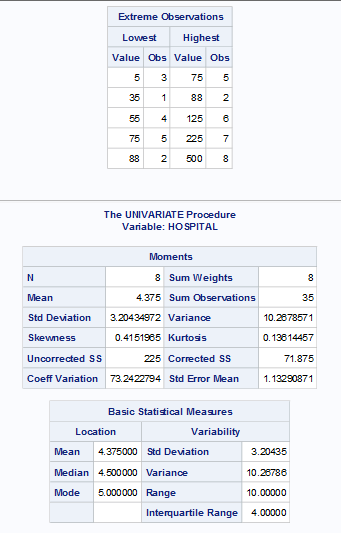
cpu time 0.12 seconds

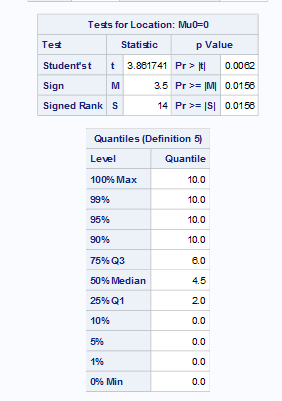
Output:

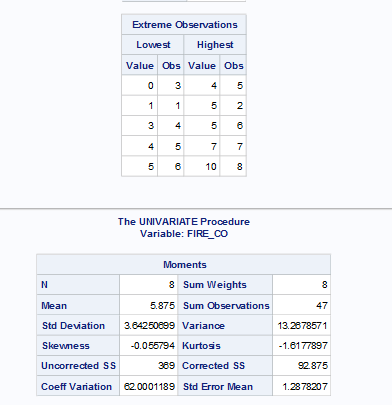


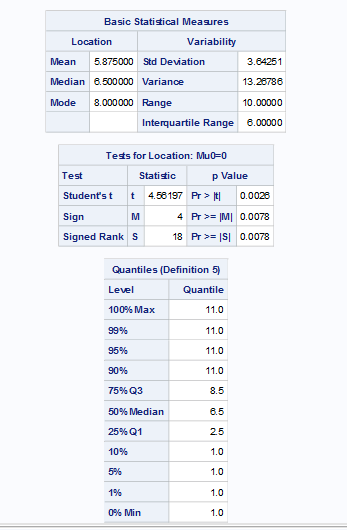


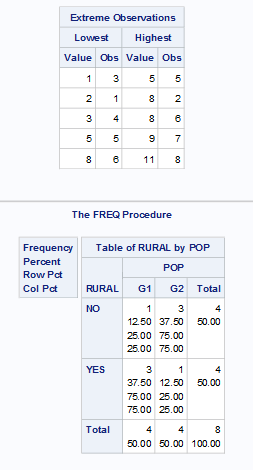


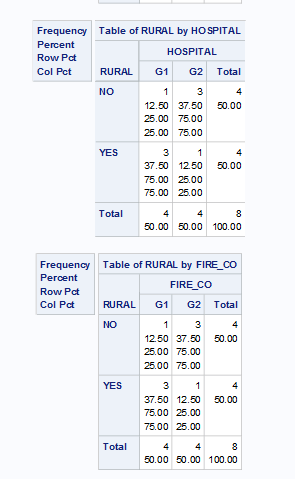












7 What’s wrong with the following program? (NOTE: There may be missing values for X, Y, and Z)

1. **data** many-err;
2. input x y z;
3. if x le **0** then x=**1**;
4. if y le **0** then y=**1**;
5. if z le **0** then z=**1**;
6. logx=log(x);
7. logy=log(y);
8. logz=log(z);
9. datalines;

1 2 3

. 7 8

4 . 10

7 8 11

1. ;
2. **proc** **corr** data=many-err / pearson spearman;
3. var x - logz;
4. **run**;

Corrected Code:

**data** many\_err;

input x y z;

if x le **0** then x=**1**;

if y le **0** then y=**1**;

if z le **0** then z=**1**;

logx=log(x);

logy=log(y);

logz=log(z);

datalines;

1 2 3

. 7 8

4 . 10

7 8 11

;

**run**;

**proc** **print** data=many\_err;

**run**;

**proc** **corr** data=many\_err pearson spearman ;

var x logz;

**run**;

8 Refer to the GRADES data. Suppose that the instructor of the class wants to see if students performed at consistent levels during the semester. There would be a problem with the grading procedure if, for example, students who earned high grades at the beginning of the semester tended to have lower grades toward the end of the semester, or if students who performed well in one week performed poorly the next week. One way to evaluate this consistency numerically is to use a split-half reliability coefficient. Choose one way to divide the 13 homework grades into two groups: one with seven assignments, one with six assignments. For example, you may choose to divide the assignment into early and late assignments, odd-number and even-number assignments, or a randomly chosen group of seven and the remaining six assignments. Then calculate the correlation between the total of the first group of assignments and the total of the second group of assignments. Typically, for a grading procedure to be considered ‘reliable’, this correlation should be 0.7 or higher. Would you conclude that the grading policy is reliable from your calculations?

Code:

**data** grades\_cross;

set college.grades\_2;

array S(**13**) S1-S13;

early\_ass=**0**;

late\_ass=**0**;

do i=**1** to **13**;

if i<=**7** then early\_ass+S(i);

else late\_ass+S(i);

drop i;

end;

**run**;

/\*proc print data=grades\_cross;

run;\*/

**proc** **corr** data=grades\_cross ;

var early\_ass ;

with late\_ass;

**run**;

Log:

576 data grades\_cross;

577 set college.grades\_2;

578 array S(13) S1-S13;

579 early\_ass=0;

580 late\_ass=0;

581 do i=1 to 13;

582 if i<=7 then early\_ass+S(i);

583 else late\_ass+S(i);

584 end;

585 run;

NOTE: There were 23 observations read from the data set

COLLEGE.GRADES\_2.

NOTE: The data set WORK.GRADES\_CROSS has 23 observations and 17

variables.

NOTE: DATA statement used (Total process time):

real time 0.04 seconds

cpu time 0.04 seconds

586

587 /\*proc print data=grades\_cross;

588 run;\*/

589

590

591 proc corr data=grades\_cross ;

592 var early\_ass ;

593 with late\_ass;

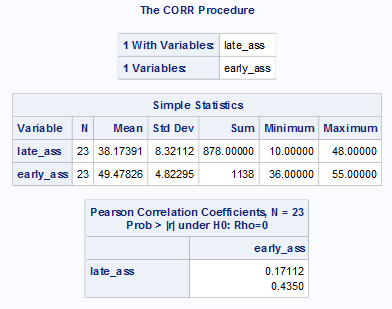
594 run;

NOTE: PROCEDURE CORR used (Total process time):

real time 0.10 seconds

cpu time 0.04 seconds

Output:



According to the observation above the grading policy is not reliable as it is below 0.7.

9 Refer to the IRIS data. Find all of the pairwise correlations among sepal length, sepal width, petal length and petal width, using all of the observations. Now, calculate the averages of sepal length, sepal width, petal length, and petal width separately for each of the three species. You can do this within PROC UNIVARIATE by listing all four variables in the VAR statement, then use an OUTPUT statement similar to this:

OUTPUT OUT=IRISAVGS MEAN=AVGSL AVGSW AVGPL AVGPW;

Next, find the correlations among the four averages. For example, one of these is the correlation between the average sepal length and the average sepal width, calculated from three data points: one from *setosa* averages, one from *versicolor* averages, and one from *virginica* averages. The first correlations are called total correlations, while the correlation between averages are called between-groups correlations. These two types of correlations are used in Fisher’s linear discriminant analysis to find mathematical expressions which describe the differences among the three species.

Code:

libname college 'C:\Users\Samil\Desktop\Sem 1\Stats for programming\SAS 9.4';

/\*proc print data=college.iris;

run;\*/

**proc** **corr** data=college.iris out=iris\_2 noprint;

var SL ;

with SW PL PW;

**run**;

**proc** **print** data=iris\_2;

**run**;

**proc** **univariate** data=college.iris noprint;

OUTPUT OUT=IRISAVGS MEAN=AVGSL AVGSW AVGPL AVGPW;

var SL SW PL PW;

by CLASS;

**run**;

**proc** **print** data=IRISAVGS;

**run**;

/\*proc corr data=IRISAVGS;

By Class;

var AVGSL ;

with AVGSW;

run;\*/

/\*proc corr data=IRISAVGS;

By Class;

var AVGPL ;

with AVGPW;

run;\*/

/\*proc corr data=IRISAVGS;

\*By Class;

var AVGSL ;

with AVGSW;

run;\*/

**proc** **corr** data=IRISAVGS out=iris\_3 noprint;

\*By Class;

var AVGSL ;

with AVGSW AVGPL AVGPW;

**run**;

**proc** **print** data=iris\_3;

**run**;

/\*proc discrim data=college.iris outstat=irisstat

method=normal pool=test

Tcorr crosslisterr noprint;

class Class;

var SL SW PL PW;

run;

proc print data=irisstat;

run;\*/

Log:

125 libname college 'C:\Users\Samil\Desktop\Sem 1\Stats for programming\SAS 9.4';

NOTE: Libref COLLEGE was successfully assigned as follows:

Engine: V9

Physical Name: C:\Users\Samil\Desktop\Sem 1\Stats for programming\SAS 9.4

126

127 /\*proc print data=college.iris;

128 run;\*/

129

130 proc corr data=college.iris out=iris\_2 noprint;

131 var SL ;

132 with SW PL PW;

133 run;

NOTE: The data set WORK.IRIS\_2 has 6 observations and 3 variables.

NOTE: PROCEDURE CORR used (Total process time):

real time 0.02 seconds

cpu time 0.03 seconds

134

135 proc print data=iris\_2;

136 run;

NOTE: There were 6 observations read from the data set WORK.IRIS\_2.

NOTE: PROCEDURE PRINT used (Total process time):

real time 0.10 seconds

cpu time 0.03 seconds

137

138 proc univariate data=college.iris noprint;

139 OUTPUT OUT=IRISAVGS MEAN=AVGSL AVGSW AVGPL AVGPW;

140 var SL SW PL PW;

141 by CLASS;

142 run;

NOTE: The data set WORK.IRISAVGS has 3 observations and 5 variables.

NOTE: PROCEDURE UNIVARIATE used (Total process time):

real time 0.06 seconds

cpu time 0.06 seconds

143

144 proc print data=IRISAVGS;

145 run;

NOTE: There were 3 observations read from the data set WORK.IRISAVGS.

NOTE: PROCEDURE PRINT used (Total process time):

real time 0.11 seconds

cpu time 0.01 seconds

146

147 /\*proc corr data=IRISAVGS;

148 By Class;

149 var AVGSL ;

150 with AVGSW;

151 run;\*/

152

153 /\*proc corr data=IRISAVGS;

154 By Class;

155 var AVGPL ;

156 with AVGPW;

157 run;\*/

158

159 /\*proc corr data=IRISAVGS;

160 \*By Class;

161 var AVGSL ;

162 with AVGSW;

163 run;\*/

164

165 proc corr data=IRISAVGS out=iris\_3 noprint;

166 \*By Class;

167 var AVGSL ;

168 with AVGSW AVGPL AVGPW;

169 run;

WARNING: The input TYPE=OUT data set is treated as an ordinary SAS data set in proc CORR.

NOTE: The data set WORK.IRIS\_3 has 6 observations and 3 variables.

NOTE: PROCEDURE CORR used (Total process time):

real time 0.04 seconds

cpu time 0.04 seconds

170

171 proc print data=iris\_3;

172 run;

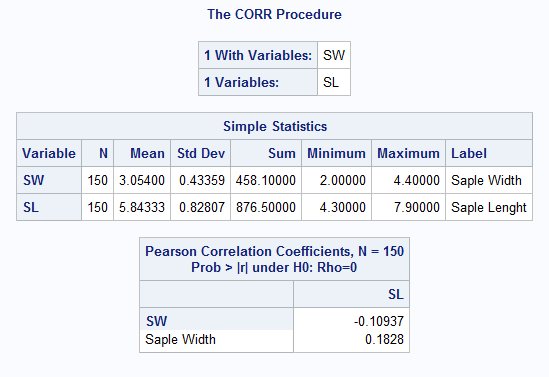
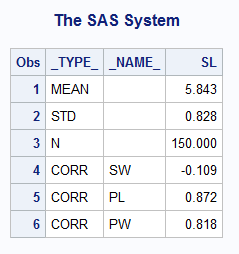
NOTE: There were 6 observations read from the data set WORK.IRIS\_3.

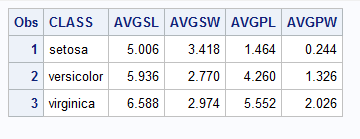
NOTE: PROCEDURE PRINT used (Total process time):

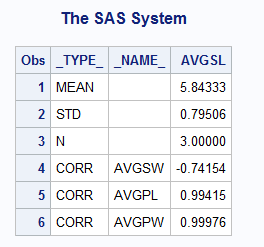
real time 0.10 seconds

cpu time 0.03 seconds

Output:







10 I was trying out a new bread recipe the other day. I spilled something on the recipe booklet, and I can’t read how much flour I’m supposed to use in the recipe. I do know that I need to use 1 cup of water, 2 tablespoons of oil, 2 tablespoons of sugar, 1 ½ teaspoons of salt, and 2 ¼ teaspoons of yeast.

Help me out. Refer to BREAD data. Find the least-squares regression equation to predict flour amounts from water, oil, sugar, salt, and yeast, and use that equation to estimate how much flour I need in my recipe. Make sure that SAS prints the estimated amount of flour needed.

Code:

libname college 'C:\Users\Samil\Desktop\Sem 1\Stats for programming\SAS 9.4';

/\*proc print data=college.bread;

run;\*/

**data** my\_bread;

input dough $ water oil sugar salt dry\_milk flour yeast wheat oregano eggs;

datalines;

my\_bread 1 2 2 1.5 0 . 2.25 0 0 0

;

**run**;

**data** bread\_2;

set college.bread my\_bread;

**run**;

/\*proc print data=bread\_2;

run;\*/

**PROC** **Reg** data=bread\_2;

model flour=water oil sugar salt yeast;

output out=predicted\_bread predicted=flour\_pred;

**run**;

Title 'Flour Predicted';

**proc** **print** data=predicted\_bread;

**run**;

Title;

\*flour=1.91244+1(0.26825)+2(0.00110)+2(0.19465)+1.5(-0.07809)+2.25(0.31990);

Log:

642 libname college 'C:\Users\Samil\Desktop\Sem 1\Stats for

642! programming\SAS 9.4';

NOTE: Libref COLLEGE was successfully assigned as follows:

Engine: V9

Physical Name: C:\Users\Samil\Desktop\Sem 1\Stats for

programming\SAS 9.4

643

644 /\*proc print data=college.bread;

645 run;\*/

646

647 data my\_bread;

648 input dough $ water oil sugar salt dry\_milk flour yeast wheat

648! oregano eggs;

649 datalines;

NOTE: The data set WORK.MY\_BREAD has 1 observations and 11 variables.

NOTE: DATA statement used (Total process time):

real time 0.03 seconds

cpu time 0.04 seconds

651 ;

652 run;

653

654 data bread\_2;

655 set college.bread my\_bread;

656 run;

NOTE: There were 11 observations read from the data set COLLEGE.BREAD.

NOTE: There were 1 observations read from the data set WORK.MY\_BREAD.

NOTE: The data set WORK.BREAD\_2 has 12 observations and 11 variables.

NOTE: DATA statement used (Total process time):

real time 0.04 seconds

cpu time 0.04 seconds

657

658

659 /\*proc print data=bread\_2;

660 run;\*/

661

662 PROC Reg data=bread\_2;

663 model flour=water oil sugar salt yeast;

664 output out=predicted\_bread predicted=flour\_pred;

665 run;

666

667 Title 'Flour Predicted';

NOTE: The data set WORK.PREDICTED\_BREAD has 12 observations and 12

variables.

NOTE: PROCEDURE REG used (Total process time):

real time 2.16 seconds

cpu time 0.85 seconds

668 proc print data=predicted\_bread;

669 run;

NOTE: There were 12 observations read from the data set

WORK.PREDICTED\_BREAD.

NOTE: PROCEDURE PRINT used (Total process time):

real time 0.38 seconds

cpu time 0.01 seconds

670 Title;

671 \*flour=1.91244+1(0.26825)+2(0.00110)+2(0.19465)+1.5(-0.07809)+2.2

671! 5(0.31990);

Output:

