**DATA** COST\_LIVING;

INPUT City $12. Index Prev\_yr\_index Housing Food Travel Utility Education Leisure Other;  
DATALINES;

Adelaide 85 83 35 10 10 9 14 10 12

Beijing 90 92 40 10 15 10 18 5 2

Copenhagen 65 64 25 15 10 10 12 12 16

Doha 56 50 30 15 5 10 10 20 10

Dubai 75 76 30 16 14 10 20 8 2

Dublin 45 43 30 10 8 12 10 15 15

Hong Kong 83 88 45 5 10 15 15 9 1

Johannesburg 35 40 45 5 5 15 15 10 5

Manila 41 42 25 10 15 15 20 10 5

Moscow 48 53 40 20 5 5 10 10 10

Mumbai 83 85 40 10 15 15 10 9 1

Munich 65 64 35 10 10 10 10 10 15

New York 89 85 40 10 15 10 20 5 5

Oslo 60 58 25 15 5 5 15 20 15

Paris 70 70 30 10 5 10 10 20 15

Seoul 73 75 30 10 10 10 15 15 10

Singapore 75 74 35 15 10 10 20 5 5

Tokyo 87 85 40 15 10 5 15 14 1

Zurich 63 61 30 10 10 15 10 10 15

;

**RUN;**

Program 2.1 Cost of Living Dataset Creation - PDV Illustration

**DATA** WORK.Air;

SET SASHELP.Air;

**RUN;**

Program 2.2 Basic form of Data Step

**DATA**;

INPUT Id;

DATALINES;

1

2

;

**RUN**;

Program 2.3 No dataset name specified

**PROC SQL;**

CREATE TABLE Table\_Name AS

SELECT

FROM

WHERE

GROUP BY;

**QUIT;**

Program 2.4 Basic SQL query

**DATA** WORK.Air (COMPRESSION = YES);

SET SASHELP.Air;

**RUN;**

Program 2.5 Compressing a dataset

**DATA** WORK.Air (ENCRYPT = YES READ=CHAPTER2);

SET SASHELP.Air;

**RUN;**

**PROC** **PRINT** DATA = WORK.Air(READ=CHAPTER2);

**RUN**;

Program 2.6 Encrypting dataset

**PROC DATASETS** LIBRARY=WORK;

MODIFY Cost\_Living;

INDEX CREATE City;

**RUN**;

Program 2.7 Creating a simple index

**DATA** COST\_LIVING;

INPUT City $12. Index Prev\_yr\_index Housing Food Travel Utility Education Leisure Other Updated MMDDYY6.;  
DATALINES;

Adelaide 85 83 35 10 10 9 14 10 12 010118

Beijing 90 92 40 10 15 10 18 5 2 010118

Copenhagen 65 64 25 15 10 10 12 12 16 020118

Doha 56 50 30 15 5 10 10 20 10 030118

Dubai 75 76 30 16 14 10 20 8 2 040118

.

.

.

;

Program 2.8 Cost of Living Dataset Creation (Partial View) – Date Variable Added

**DATA** Num\_to\_Char;

SET Cost\_Living;

Index\_char = PUT(Index, 3.);

**RUN**;

Program 2.9 Numeric to Character conversion

**DATA** Convert;

INPUT Id\_Char $4. Turnover $7. Turnover\_w\_Currency $8. Source\_Mixed $3.;

DATALINES;

0001 20,000 $20,000 A1

0002 10,000 $10,000 2

;

Program 2.10 Character to Numeric conversion dataset

**DATA** DateTime;

INPUT Id Date\_Time Datetime20.;

DATALINES;

1 01aug19:09:10:05.2

2 01aug20:19:20:10.4

;

**DATA** Convert\_DateTime;

SET DateTime;

FORMAT Orig\_Date Datetime.;

Orig\_Date = Date\_Time;

FORMAT Orig\_Date\_1 Datetime7.;

Orig\_Date\_1 = Date\_Time;

FORMAT Orig\_Date\_2 Datetime12.;

Orig\_Date\_2 = Date\_Time;

**RUN**;

Program 2.11 Informat and Formats for Date Time Variable

**DATA** Updated\_2019;

SET Cost\_Living;

WHERE Year(Updated) = 2019;

**RUN**;

Program 2.12 Introduction to the WHERE statement

**DATA** Updated\_2019;

SET Cost\_Living;

WHERE Year(Updated) = 2019;

IF Index >= 80;

**RUN**;

Program 2.13 Using WHERE and IF in same data step

**DATA** Known\_Components;

SET Index;

Known\_Component\_Index = Index – Other;

WHERE Known\_Component\_Index >= 80;

**RUN**;

Program 2.14 Incorrect use of WHERE

**PROC PRINT** DATA = Cost\_Living (FIRSTOBS=4 OBS=5);

**RUN**;

Program 2.15 Using the Obs and Firstobs option

**DATA** Keep\_and\_Drop (DROP = Prev\_yr\_index);

SET Cost\_Living (KEEP = City Index Prev\_yr\_index);

WHERE Index < Prev\_yr\_index;

**RUN**;

Program 2.16 Using the Drop and Keep option

**PROC DATASETS** Library=Work;

CONTENTS DATA=Cost\_Living;

**RUN**;

Program 2.17 Viewing properties of a table

PROC SQL;

Select \* From Dictionary.Tables;

QUIT;

Fig 2.18 Snapshot of Dictionary Table Output

PROC SQL;

Select \* From Dictionary.Tables

Where Libname eq "WORK";

QUIT;

PROC SQL;

Select \* From Dictionary.Columns

Where Libname eq "WORK";

QUIT;

Fig 2.19 Snapshot of Work Library

Data Updated\_2019;

Set Cost\_Living;

Where Year(Updated) = 2019;

If Index >= 80;

Put \_All\_ ;

Run;

/\*Above doesn’t have a graphic output and the output is in the chapter word file\*/

PROC PRINT DATA = Updated\_2019;

VAR Updated;

RUN;

Title '\_ALL\_ in a Data Step';

PROC PRINT DATA = Updated\_2019;

VAR \_ALL\_;

RUN;

Fig 2.20 \_ALL\_ in a Data Step

Data Test;

Set Updated\_2019;

If 1 < \_N\_ <10;

Run;

Proc print;

Run;

Fig 2.21 Selecting Observations using \_N\_

Data Automatic;

Input A $ B;

Counter = \_N\_;

Datalines;

X 1

Y 2

Z 3

;

Run;

Data Automatic\_Challenge;

Input A $ B;

Retain Counter 2;

\_N\_ = Counter+1;

Test\_N = \_N\_;

Datalines;

X 1

Y 2

Z 3

;

Run;

Title '\_N\_ automatic value';

Proc Print Data=Automatic;

Run;

Title '\_N\_ automatic value overwritten';

Proc Print Data=Automatic\_Challenge;

Run;

Fig 2.22 Overwriting value of \_N\_