```
/* ALL CODE FOR INTRODUCTION TO SAS 9.4 (SAS Studio/OnDemand) SEMINAR */

**********************************************************
*                   Accessing Data                        *
**********************************************************;

/*-------------------
  Importing data
---------------------*/

* Import wizard and  proc import;
proc import datafile="/home/u63687742/sas_data/hs0.xlsx" dbms = xlsx replace out=work.temp;
 sheet = hs0;
 getnames = yes;
run;

/*-------------------
  Saving data
---------------------*/

* Save temporary dataset "temp" as a permanent file;

data '/home/u63687742/sas_data/hs01';
 set temp;
run;


*Print the first 10 observations;
proc print data='/home/u63687742/sas_data/hs01' (obs = 10);
run;

/*-------------------
  libname
---------------------*/

* define a library named IN;

libname IN '/home/u63687742/sas_data';
```

```
*Or instead of folder address we can use library IN which we have defined before;
*Caution this step will overwrite an existing file if you have the same file name;
*By default SAS uses work library;
data in.hs01;
  set temp;
run;

* Create a temporary dataset called hs0 ;
* This temporary dataset will be save in work library;

data work.hs0;
  set in.hs0;
run;

*We can also use point and click to import data from different file types;
* Code below is created using point and click;

/* Generated Code (IMPORT) */
/* Source File: hs0.csv */
/* Source Path: /home/sjalal0/sas_data */

%web_drop_table(WORK.IMPORT);


FILENAME REFFILE '/home/u63687742/sas_data/hs0.csv';

PROC IMPORT DATAFILE=REFFILE
    DBMS=CSV
    OUT=WORK.IMPORT;
    GETNAMES=NO;
RUN;

PROC CONTENTS DATA=WORK.IMPORT; RUN;


%web_open_table(WORK.IMPORT);
```

```
*********************************************************
*                   Exploring Data                      *
*********************************************************;

* Examine data using proc contents and proc print;
proc contents data=hs0;
run;

* Print the first 20 observations;
proc print data=hs0 (obs=20);
run;

* If we only want to print some variables, we can use the "var statement";
proc print data=hs0 (obs=20);
  var gender id race ses schtyp prgtype read;
run;

/*---------------------------------
   Descriptive statistics : means
-----------------------------------*/


* Descriptive statistics with proc means;
proc means data=hs0;
run;

* Means for a subset of variables using var;
* We can add what kind of summary statistics we need to be printed;
proc means data=hs0 n mean median std var;
  var read math science write;
run;

* Means for a subset of variables using var;
* Filtering observations using where;
proc means data=hs0 n mean median std var;
  var read math science write;
  where read>=60;
run;
```

```
* Means broken down by group (ses) using class;
proc means data=hs0 n mean median std var;
  class ses;
  var read math science write;
run;

/*---------------------------------------
   Descriptive statistics: univariate
----------------------------------------*/


* Descriptive statistics using proc univariate;
proc univariate data=hs0;
   var read write;
run;

* Histogram with normal curve overlay from proc univariate;
* Option noprint supresses the output of univariate command and returns the histogram only;
proc univariate data=hs0 noprint;
  var write;
  histogram / normal;
run;

/*-------------------
   Frequency table
---------------------*/


* Frequency distribution table;
proc freq data=hs0;
  table ses gender schtyp prgtype;
run;

* A crosstab using proc freq;
proc freq data=hs0;
  table prgtype*ses;
run;
```

```
/*---------------
   Correlation
---------------*/


* Correlations using proc corr with pairwise
deletion of missing observations (default);
proc corr data=hs0;
  var write read science;
run;

* Correlations using proc corr with listwise
deletion of missing observations (nomiss option);
proc corr data=hs0 nomiss;
  var write read science;
run;

/*---------------
   Plots
---------------*/

* Scatter plot matrix;
proc corr data=hs0 nomiss plots=matrix;
  var write read science;
run;

* Scatter plot;
proc sgplot data = hs0;
  scatter x = read  y = write;
run;

* Scatter plot with gender of observation indicated;
proc sgplot data=hs0;
  scatter x=write y=read / group=gender;
run;

* Vertical bar chart representing mean of variable write by ses with error bars;
proc sgplot data=hs0;
```

```
  vbar ses /response = write stat=mean limits=both;
run;

* histogram of variable read with normal curve and density plot overlayed;
proc sgplot data=hs0;
  histogram read;
  density read / type=normal;
  density read / type = kernel;
run;

*********************************************************
*                   Modifying Data                      *
*********************************************************;

/*--------------
   proc Format
--------------*/

* Create value labels for the variable schtyp;
proc format;
  value scl 1 = "Public"
            2 = "Private";
run;

* Frequency table using the labels with a format statement;
proc freq data = hs0;
  tables schtyp;
  format schtyp scl.;
run;


* permanently apply a value label to a variable in a data step;
data hs0;
  set in.hs0;
  format schtyp scl.;
run;

*proc contents;
procedure contents data=hs0;
```

```
run;

*Recoding a continuous variable using formats;
proc format;
   * create format for test score;
   value score 25 - 60 = "low score"
           61 -80 = "high score";
run;

data hs0;
   set hs0;
 * apply value labels to variable read;
 format read score.;
run;

 * variable read can be used in its original format;
proc means data=hs0;
  var read;
run;

 * variable read can be also be used in class statement as categorical;
proc means data=hs0;
  class read;
  var math;
run;


/*---------------
   label
---------------*/

* label the dataset and variable schtyp;
data hs0(label="High School and Beyond");
  set hs0;
  label schtyp = "type of school";
run;

*proc contents;
proc contents data=hs0;
```

```
run;

/*--------------
   rename
--------------*/

* Rename schtype to public and gender to female in a temporary dataset hs0b;
data hs0;
   set hs0 (rename=(gender=female));
run;
*This is another way to rename;
data hs0;
   set hs0;
   rename schtyp=public;
run;

/*------------------------------------
   if statment and if-then statment
------------------------------------*/

*****Now we will run a longer data step to do a variety of tasks****;

*  proc format that define a variety of formats;
proc format;
  * create value labels for schtyp ;
  value scl 1 = "public"
            2 = "private";

  * create value labels abcdf for grade ;
  value abcdf 0 = "F"
              1 = "D"
              2 = "C"
              3 = "B"
              4 = "A";

  * create value labels for female ;
  value fm 1 = "female"
           0 = "male";
run;
```

```
*** Note the code below replicates some of task we did above***;
* create data file hs1, label it High School and Beyond;
data hs_temp(label="High School and Beyond") ;
  *inpute data from library IN, rename gender to female;
  set in.hs0 (rename=(gender=female));

  * label the variable schtyp ;
  label schtyp = "type of school";

  * apply value labels to schtyp;
  format schtyp scl.;

  * apply value labels to female;
  format female fm.;

  * the if statement recodes values of 5 in the variable race to be missing (.) ;
  if race = 5 then race = .;

  * the if-then statements create a new variable, called prog,
    which is numeric variable ;
  if prgtype = "academic"   then prog = 1;
  if prgtype = "general"    then prog = 2;
  if prgtype = "vocational" then prog = 3;


  * create a variable called total that is the sum of read, write, math, and science ;
  total = read + write + math + science;
  * label the variable total ;
  label total = "Total grade";

  * the if-then statements recode the variable total into the variable grade ;
  if (total < 80)              then grade = 0;
  if (80  <= total  <  110) then grade = 1;
  if (110 <= total  <  140) then grade = 2;
  if (140 <= total  <  170) then grade = 3;
  if (total  >= 170)         then grade = 4;
  if (total = .)             then grade = .;
```

```
 * apply value labels to variable grade;
  format grade abcdf.;
run;

*Check the output using proc contents;
proc contents data = hs_temp;
run;

*print the first 20 observations;
proc print data = hs_temp (obs = 20);
run;
*proc freq uses labels in the result;
proc freq data = hs_temp;
  tables schtyp*female;
run;

* Save temporary dataset as a permanent dataset;
data in.hsb1;
  set hs_temp;
run;


/*--------------
   functions
---------------*/

*Create variables using SAS function;
data hs_temp;
  set hs_temp;
  total2 = sum(of read write math science);
  * similarly, mean, max, min and more;
  mean= mean(of read write math science);
run;

*Modifying variables using procedures;
*There are also a number of SAS procidures and functions that can use for modifying data;
*standardize read and write using proc standard;
```

```
proc standard data = hs_temp mean=0 std=1 out=hs_temp;
   var read write;
run;

* look at the mean and standard deviation;
proc means data=hs_temp mean std;
 var read write;
run;

*********************************************************
*                   Managing Data                       *
*********************************************************;

/*----------------------------------------------
   if and where statment to filter observations
-----------------------------------------------*/

*Selecting cases using if statements;
data highread lowread;
  set in.hs1;
  if read >=60 then output highread;
  if read < 60 then output lowread;
run;

*set the title in the result table;
title "high reading scores";

*mean of read for highread data;
proc means data=highread n mean;
  var read;
run;

*mean of read for lowread data;
title "low reading scores";
proc means data=lowread n mean;
  var read;
run;

title; /* this statement clears the title we set earlier */
```

```
* Selecting cases using where statement;
data highread;
  set in.hs1;
  where read >=60;
run;

/*-----------------------------------------------
   keep and drop variables
-----------------------------------------------*/

*Keeping variables id, female, and write;
data hskept;
  set highread;
  keep id female read write;
run;

* dropping variables ses and prog;
data hsdropped;
  set highread;
  drop ses prog;
run;

/*-----------------------------------------------
   appending datasets
-----------------------------------------------*/

* first we let's create two subset of data for female and male students;
data hsfemale hsmale;
  set in.hs1;
  if female=1 then output hsfemale;
  if female=0 then output hsmale;
run;


* Use DATA step to combine the two files and save them as hs1 ;
data hs1;
  set hsmale hsfemale;
run;
```

```
* Now you should have a file with both males and females;
proc means data=hs1;
  class female;
  var write;
run;


/*----------------------------------------------
   Merging datasets;
-----------------------------------------------*/


* examine the two datasets;
proc print data=in.hsdem;
run;

proc print data=in.hstest;
run;

* sort both files by the variable that identifies the cases in each file (id);
proc sort data=in.hsdem out=dem;
  by id;
run;

proc sort data=in.hstest out=test;
  by id;
run;

* merge the datasets;
data all;
  merge dem test;
  by id;
run;
* print merged dataset;
proc print data=all;
run;
*********************************************************
*                   Analyzing Data                      *
```

```
**********************************************************;

* Chi-squared test;
proc freq data=in.hs1;
  table prgtype*ses / chisq expected;
run;

/*-----------
   t-test
-------------*/

* Paired t-test;
proc ttest data=in.hs1;
  paired write*read;
run;

* Two sample independent t-test;
proc ttest data=in.hs1 plots=none;
  class female;
  var write;
run;


/*--------------
   Regression
----------------*/


* Regression;
proc reg data=in.hs1;
  model write = female read;
run;
quit;


* This regression code outputs a temporary dataset (temp)
* that contains the predicted values of math and the residuals ;
proc reg data =in.hs1;
  model math = write socst;
  output out=temp p=predict r=resid;
run;
quit;

* Inspect the temporary dataset (temp);
proc print data=temp (obs=20);
  var math predict resid;
run;

/*-----------------------
   Logistic Regression
------------------------*/


* Logistic regression;
* Create a dichotomous variable honors;
data hs2;
  set in.hs1;
  honors = (write >= 60);
run;

* Logistic regression with descending option (so model predicts 1s rather than 0s);
* Almost always use descending;
proc logistic data=hs2 descending;
  model honors = female read;
run;


***********************
*****   The end   *****
***********************;
```