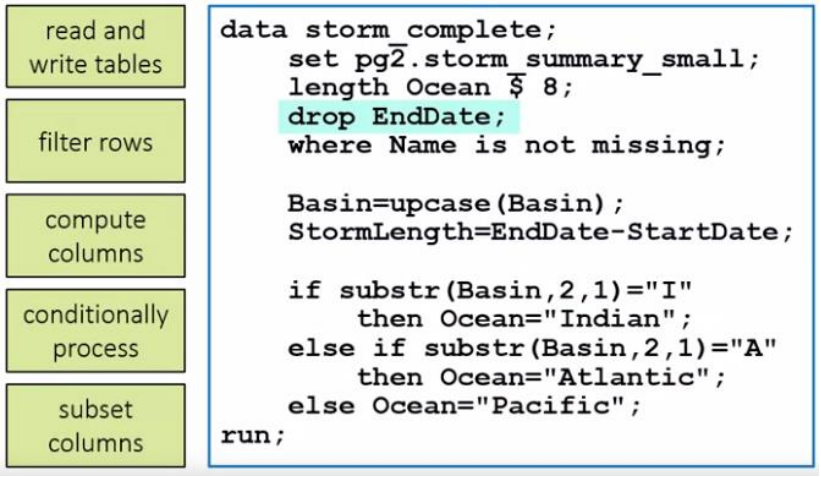
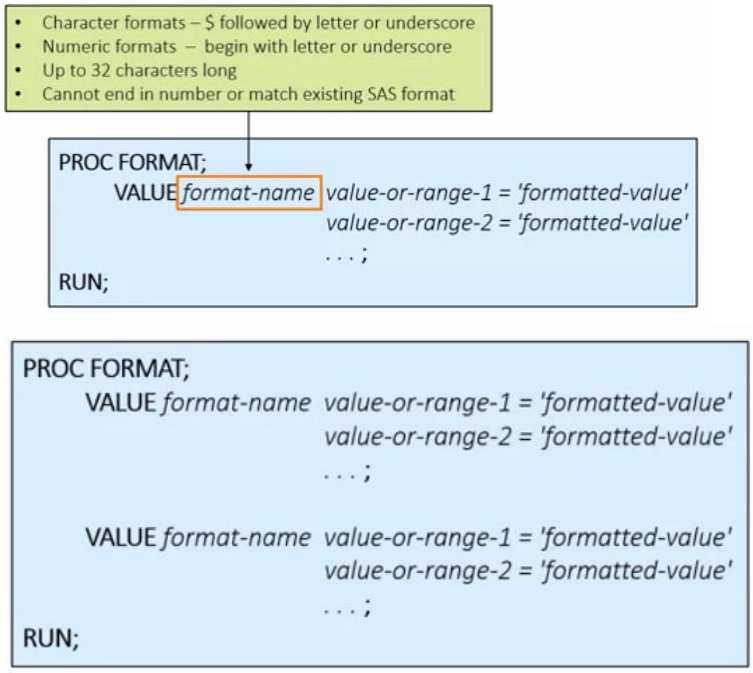
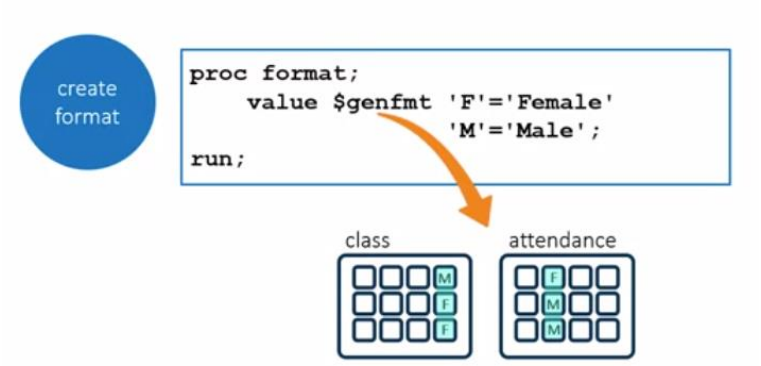
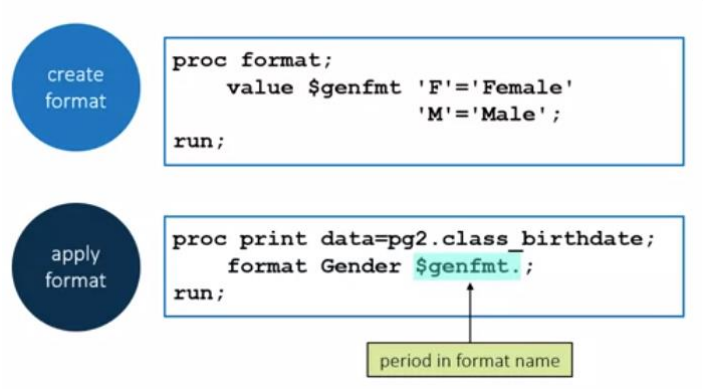
Doing more with SAS programming
Data Step Review

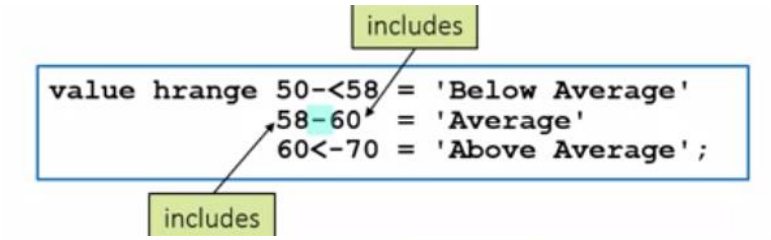


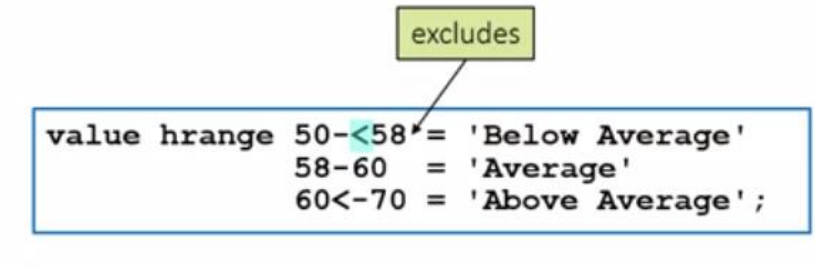
Create custom format

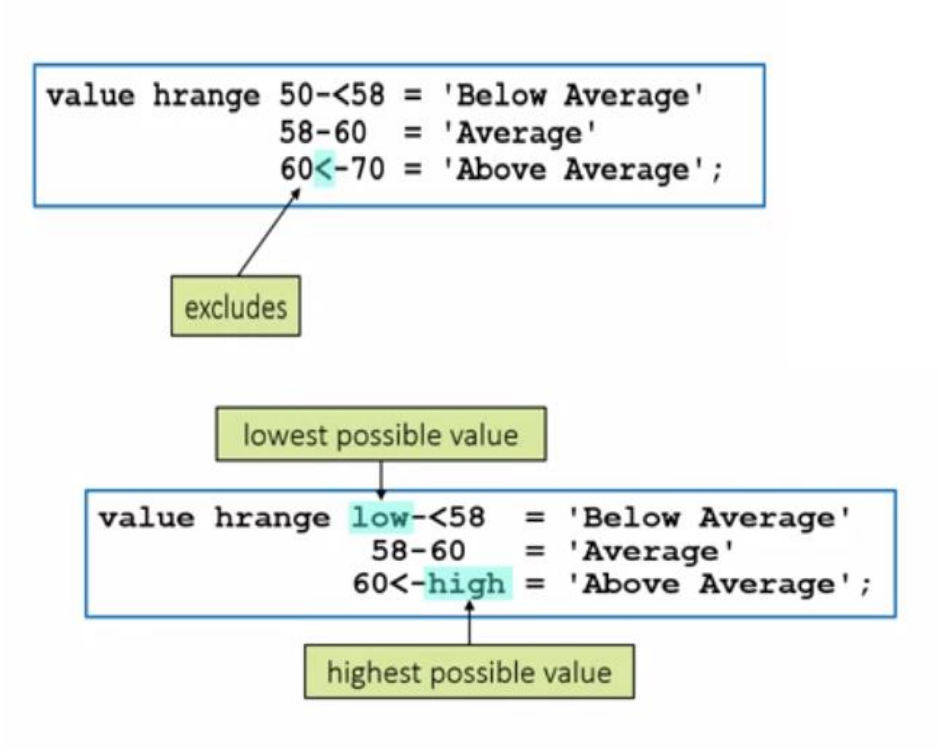


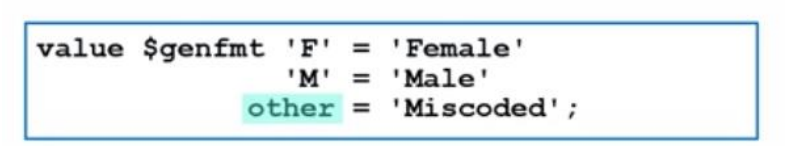

create format
proc format;
value $genfmt 'F'='Female'
'M'='Male';
run;
class
attendance
M
F
F
F
M
M


create format
proc format;
value $genfmt 'F'='Female'
'M'='Male';
run;
apply format
proc print data=pg2.class_birthdate;
format Gender $genfmt.;
run;
period in format name


includes
value hrange 50-<58 = 'Below Average'
58-60 = 'Average'
60<-70 = 'Above Average';
includes


excludes
value hrange 50-<58 = 'Below Average'
58-60 = 'Average'
60<-70 = 'Above Average';



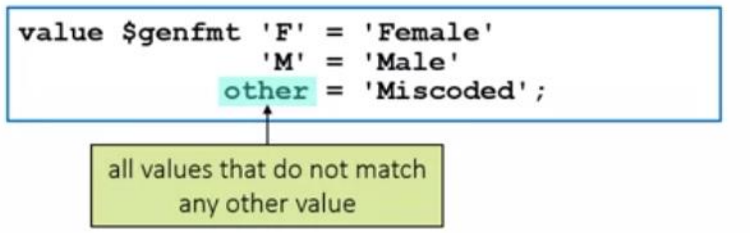



```
*****************************************************;
*  Creating and Using Custom Formats                *;
*****************************************************;
*  Syntax                                           *;
*                                                   *;
*    PROC FORMAT;                                   *;
*      VALUE format-name                            *;
*            value-or-range-1='formatted-value'     *;
*            value-or-range-2='formatted-value'     *;
*            ... ;                                  *;
*    RUN;                                           *;
*****************************************************;
```

```
proc format;
    value $genfmt 'F'='Female'
                  'M'='Male'
                  other='Miscoded';
    value hrange low-<58  = 'Below Average'
                 58-60    = 'Average'
```

```
                60<-high = 'Above Average';
run;

proc print data=pg2.class_birthdate noobs;
    where Age=12;
    var Name Gender Height;
    format Gender $genfmt. Height hrange.;
run;

*************************************************************;
*  Demo                                                     *;
*  1) Notice the syntax for creating the STDATE format in   *;
*     the PROC FORMAT step.                                 *;
*  2) Add a VALUE statement to the PROC FORMAT step to      *;
*     create the $REGION format with the following          *;
*     labels:                                               *;
*         NA => Atlantic                                    *;
*         WP, EP, SP => Pacific                             *;
*         NI, SI => Indian                                  *;
*         blank => Missing                                  *;
*         other => Unknown                                  *;
*  3) Highlight the PROC FORMAT step and run the selected   *;
*     code. Verify in the SAS log that the formats have     *;
*     been output.                                          *;
*  4) Add a FORMAT statement in the PROC FREQ step to       *;
*     format Basin with the $REGION format and StartDate    *;
*     with the STDATE format. Highlight PROC FREQ step      *;
*     and run the selected code.                            *;
*************************************************************;

proc format;
    value stdate low - '31DEC1999'd = '1999 and before'
                 '01JAN2000'd - '31DEC2009'd = '2000 to 2009'
                 '01JAN2010'd - high = '2010 and later'
                 . = 'Not Supplied';
    *Add a VALUE statement;
    value $region 'NA'='Atlantic'
                  'WP', 'EP', 'SP'='Pacific'
                  'NI', 'SI'='Indian'
                  ''='Missing'
                  other='Unknow';
run;

proc freq data=pg2.storm_summary;
    tables Basin*StartDate / norow nocol;
    format StartDate stdate. Basin $region.;
```

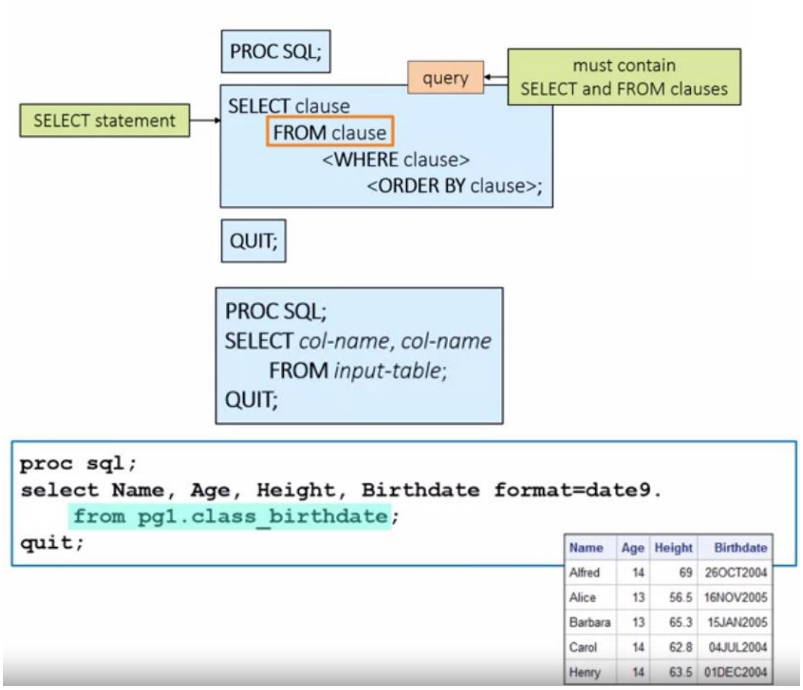
```
run;
```




```
*************************************************************;
*  Activity 4.03                                            *;
*  1) Review the PROC FORMAT step that creates the          *;
*     $REGION format that assigns basin codes into          *;
*     groups. Highlight the step and run the selected       *;
*     code.                                                 *;
*  2) Notice the DATA step includes IF-THEN/ELSE            *;
*     statements to create a new column named BasinGroup.   *;
*  3) Delete the IF-THEN/ELSE statements and replace it     *;
*     with an assignment statement to create the            *;
*     BasinGroup column. Use the PUT function with Basin    *;
*     as the first argument and $REGION. as the second      *;
*     argument.                                             *;
*  4) Highlight the DATA and PROC MEANS steps and run the   *;
*     selected code. How many BasinGroup values are in      *;
*     the summary report?                                   *;
*************************************************************;

proc format;
    value $region 'NA'='Atlantic'
                  'WP','EP','SP'='Pacific'
                  'NI','SI'='Indian'
                  ' '='Missing'
                  other='Unknown';
run;

data storm_summary;
    set pg2.storm_summary;
    Basin=upcase(Basin);
    *Delete the IF-THEN/ELSE statements and replace them with an
assignment statement;
    if Basin='NA' then BasinGroup='Atlantic';
    else if Basin in ('WP','EP','SP') then BasinGroup='Pacific';
    else if Basin in ('NI','SI') then BasinGroup='Indian';
```

```
    else if Basin=' ' then BasinGroup='Missing';
    else BasinGroup='Unknown';
run;

proc means data=storm_summary maxdec=1;
     class BasinGroup;
     var MaxWindMPH MinPressure;
run;
```

How well you use SAS?



expression AS *col-name*

```
proc sql;
select Name, Age, Height*2.54 as HeightCM format=5.1,
       Birthdate format=date9.
    from pg1.class_birthdate;
quit;
```

Name	Age	HeightCM	Birthdate
Alfred	14	175.3	26OCT2004
Alice	13	143.5	16NOV2005
Barbara	13	165.9	15JAN2005
Carol	14	159.5	04JUL2004
Henry	14	161.3	01DEC2004

WHERE *expression*

```
proc sql;
select Name, Age, Height, Birthdate format=date9.
    from pg1.class_birthdate
    where age > 14;
quit;
```

Name	Age	Height	Birthdate
Janet	15	62.5	02APR2003
Mary	15	66.5	26MAR2003
Philip	16	72	21NOV2002
Ronald	15	67	14OCT2003
William	15	66.5	28DEC2003

WHERE *expression*

```
proc sql;
select Name, Age, Height, Birthdate format=date9.
    from pg1.class_birthdate
    where age > 14;
quit;
```

Name	Age	Height	Birthdate
Janet	15	62.5	02APR2003
Mary	15	66.5	26MAR2003
Philip	16	72	21NOV2002
Ronald	15	67	14OCT2003
William	15	66.5	28DEC2003

WHERE *expression*

```
proc sql;
select Name, Age, Height, Birthdate format=date9.
    from pg1.class_birthdate
    where age > 14;
quit;
```

Name	Age	Height	Birthdate
Janet	15	62.5	02APR2003
Mary	15	66.5	26MAR2003
Philip	16	72	21NOV2002
Ronald	15	67	14OCT2003
William	15	66.5	28DEC2003

ORDER BY *col-name* <DESC>

```
proc sql;
select Name, Age, Height, Birthdate format=date9.
    from pg1.class_birthdate
    where age > 14
    order by Height desc;
quit;
```

default sort order is ascending

Name	Age	Height	Birthdate
Philip	16	72	21NOV2002
Ronald	15	67	14OCT2003
Mary	15	66.5	26MAR2003
William	15	66.5	28DEC2003
Janet	15	62.5	02APR2003

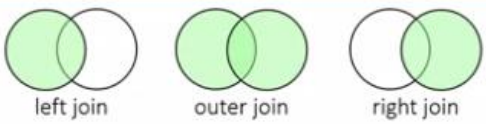
CREATE TABLE *table-name* AS

```
proc sql;
create table work.myclass as
    select Name, Age, Height
        from pg1.class_birthdate
        where age > 14
        order by Height desc;
quit;
```

FROM *table1* INNER JOIN *table2*
ON *table1.column = table2.column*

```
proc sql;
select Grade, Age, Teacher
    from pg1.class_update inner join pg1.class_teachers
    on class_update.Name = class_teachers.Name;
quit;
```

```
proc sql;
select Grade, Age, Teacher
    from pg1.class_update inner join pg1.class_teachers
    on class_update.Name = class_teachers.Name;
quit;
```



**********;

```
*  Activity 3.02                                                *;
*    1) Run the program. Examine the results and the log.       *;
*       Are the two WHERE statements applied?                   *;
*    2) Change the second WHERE statement to WHERE ALSO         *;
*       and rerun the code. Examine the results and the         *;
*       log. Are the two WHERE statements applied?              *;
************************************************************;

proc print data=pg1.storm_summary;
     where MaxWindMPH>156;
     where MinPressure>800 and MinPressure<920;
run;

************************************************************;
*  Activity 3.04                                                *;
*    1) Change the value in the %LET statement from NA to       *;
*       SP.                                                     *;
*    2) Run the program and carefully read the log.             *;
*       Which procedure did not produce a report?               *;
*       What is different about the WHERE statement in          *;
*       that step?                                              *;
************************************************************;

%let BasinCode=NA;

proc means data=pg1.storm_summary;
     where Basin="&BasinCode";
     var MaxWindMPH MinPressure;
run;

proc freq data=pg1.storm_summary;
     where Basin='&BasinCode';
     tables Type;
run;


************************************************;
*   SAS Programming Process                    *;
************************************************;
*   This program is an example of code that you  *;
*   learn in the class to analyze international  *;
*   storm data. The program follows the SAS      *;
*   programming process:                         *;
*    1) Access data                              *;
*    2) Explore data                             *;
*    3) Prepare data                             *;
```

```
*    4) Analyze and report on data                *;
*    5) Export results                            *;
************************************************;

***************;
* Section 1:  *;
* Access Data *;
***************;

options validvarname=v7;
ods graphics on;

*Path is assigned in the cre8data.sas program;
*%let path=s:/workshop;

libname pg1 base "&path/data";

proc import datafile="/home/elva40120/EPG194/data/storm.xlsx"
                dbms=xlsx out=storm_damage replace;
     sheet="Storm_Damage";
run;

****************;
* Section 2:   *;
* Explore Data *;
****************;

title "Explore Basin and Status Codes";
proc freq data=pg1.storm_summary;
     tables basin type;
run;

title "Summary Statistics for Maximum Wind(MPH) and Minimum Pressure";
proc means data=pg1.storm_summary;
     var MaxWindMPH MinPressure;
run;

title "First 5 Rows from Imported Storm Damage";
proc print data=storm_damage(obs=5);
run;

*******************;
* Section 3:      *;
* Prepare Data    *;
*******************;
```

```
data storm_summary2;
      set pg1.storm_summary pg1.storm_2017(drop=location);
      length OceanCode $ 7 BasinName $ 14;
      drop oceancode;
      Basin=upcase(basin);
      OceanCode=substr(basin,2,1);
      key=cats(season,name);
      StormLength=enddate-startdate;

      if oceancode="A" then Ocean="Atlanic";
      else if oceancode="P" then Ocean="Pacific";
      else if oceancode="I" then Ocean="Indian";

      if Basin="NA" then BasinName="North Atlantic";
      else if Basin="SA" then BasinName="South Atlantic";
      else if Basin="WP" then BasinName="West Pacific";
      else if Basin="EP" then BasinName="East Pacific";
      else if Basin="SP" then BasinName="South Pacific";
      else if Basin="NI" then BasinName="North Indian";
      else if Basin="SI" then BasinName="South Indian";
run;

data storm_damage2;
      set storm_damage;
      Name=upcase(scan(Event,-1));
      Season=Year(date);
      key=cats(season,name);
      drop Event Date;
      format Cost dollar16.;
run;

proc sql;
create table damage_detail as
select d.name, d.season, basinname, maxwindmph, minpressure,
stormlength, cost, deaths
    from storm_damage2 as D, storm_summary2 as S
    where d.key=s.key order by cost desc;
quit;

*****************************;
* Section 4:                *;
* Analyze and Report on Data  *;
* Export Results            *;
*****************************;
%let Year=2016;
%let basin=North Atlantic;
```

```
ods noproctitle;
ods excel file="&path/output/storm_report&year..xlsx"
          options(sheet_interval="proc"
          sheet_name="&Year Storms by Basin"
          embedded_titles="yes");

title1 "Number of Storms by Type and Basin";
title2 "&year Season";
proc freq data=storm_summary2 order=freq;
    tables basinname / nopercent nocum plots=freqplot;
     tables basinname*type / norow nocol crosslist ;
     where season=&year;
run;

ods excel options(sheet_name="&year Wind Statistics");
title1 "Wind Statistics by Storm";
title2 "Year &year";
proc means data=pg1.storm_detail mean min max maxdec=0 nonobs;
     class name;
     var wind;
     where season=&year;
     output out=hur_stats mean=AvgWind min=MinWind max=MaxWind;
run;

data map;
     set storm_summary2;
     length maplabel $ 20;
     where season=&year and basinname="&basin";
     if maxwindmph<100 then MapLabel=" ";
     else maplabel=cats(name,"-",maxwindmph,"mph");
     keep lat lon maplabel maxwindmph;
run;

title1 "Tropical Storms in &year Season";
title2 "&basin Basin";
footnote1 "Storms with MaxWind>100mph are labeled";

ods excel options(sheet_name="&year &Basin Basin");
proc sgmap plotdata=map;
    *openstreetmap;
    esrimap
url='http://services.arcgisonline.com/arcgis/rest/services/World_Physi
cal_Map';
    bubble x=lon y=lat size=maxwindmph /
              datalabel=maplabel datalabelattrs=(color=red size=8);
run;
```

```
************************************************************;
*  Understanding SAS Program Syntax                        *;
************************************************************;
*  Syntax                                                  *;
*    /*comment*/                                           *;
*    *comment;                                             *;
*;
************************************************************;
************************************************************;
*  Demo                                                    *;
*    1) Run the program. Does it run successfully?         *;
*    2) Use the Format Code feature to improve the         *;
*       program spacing.                                   *;
*       * Enterprise Guide: Select Edit => Format Code.    *;
*         You can also right-click in the program          *;
*         and select Format Code or press Ctrl and I.      *;
*       * SAS Studio: Click Format Code. You can also      *;
*         right-click in the program and select Format     *;
*         Code.                                            *;
*    3) Add the following text as a comment before the     *;
*       DATA statement:                                    *;
*           Program created by <your-name>                 *;
*    4) Comment out the first TITLE statement and the      *;
*       WHERE statement in PROC PRINT.                     *;
************************************************************;

data mycars;
     set sashelp.cars;
     AvgMPG=mean(mpg_city, mpg_highway);
run;

title "Cars with Average MPG Over 35";

proc print data=mycars;
     var make model type avgmpg;
     where AvgMPG > 35;
run;

title "Average MPG by Car Type";

proc means data=mycars mean min max maxdec=1;
     var avgmpg;
     class type;
RUN;
```

```
TITLE;

*************************************************;
*     Exploring Data with Procedures            *;
*************************************************;
*  Syntax                                       *;
*                                               *;
*    PROC PRINT DATA=input-table(OBS=n);        *;
*        VAR col-name(s);                       *;
*    RUN;                                       *;
*                                               *;
*    PROC MEANS DATA=input-table;               *;
*        VAR col-name(s);                       *;
*    RUN;                                       *;
*                                               *;
*    PROC UNIVARIATE DATA=input-table;          *;
*        VAR col-name(s);                       *;
*    RUN;                                       *;
*                                               *;
*    PROC FREQ DATA=input-table;                *;
*        TABLES col-name(s);                    *;
*    RUN;                                       *;
*************************************************;

proc print data=sashelp.cars(obs=10);
     var Make Model Type MSRP;
run;

proc means data=sashelp.cars;
     var EngineSize Horsepower MPG_City MPG_Highway;
run;

proc univariate data=sashelp.cars;
     var MPG_Highway;
run;

proc freq data=sashelp.cars;
     tables Origin Type DriveTrain;
run;

*************************************************************;
*  Demo                                                     *;
*    1) Complete the PROC PRINT statement to list the data in *;
*       PG1.STORM_SUMMARY. Print the first 10 observations.   *;
*       Highlight the step and run the selected code.         *;
*    2) Add a VAR statement to include only the following     *;
```

```
*       columns: Season, Name, Basin, MaxWindMPH, MinPressure,*;
*       StartDate, and EndDate. Add "list first 10 rows" as a *;
*       comment before the PROC PRINT statement. Run the step.*;
*    3) Copy the PROC PRINT step and paste it at the end of   *;
*       the program. Change PRINT to MEANS. Remove the OBS=   *;
*       data set option to analyze all observations. Modify   *;
*       the VAR statement to calculate summary statistics for *;
*       MaxWindMPH and MinPressure. Add "calculate summary    *;
*       statistics" as a comment before the PROC MEANS        *;
*       statement. Highlight the step and run the selected    *;
*       code.                                                 *;
*    4) Copy the PROC MEANS step and paste it at the end of   *;
*       the program. Change MEANS to UNIVARIATE. Add "examine *;
*       extreme values" as a comment before the PROC          *;
*       UNIVARIATE statement. Highlight the step and run the  *;
*       selected code.                                        *;
*    5) Copy the PROC UNIVARIATE step and paste it at the end *:
*       of the program. Change UNIVARIATE to FREQ. Change the *;
*       VAR statement to a TABLES statement to produce        *;
*       frequency tables for Basin, Type, and Season. Add     *;
*       "list unique values and frequencies" as a comment     *;
*       before the PROC FREQ statement. Highlight the step    *;
*       and run the selected code.                            *;
************************************************************;

proc print data=PG1.STORM_SUMMARY (obs=10);
          var  Season Name Basin MaxWindMPH MinPressure StartDate
EndDate;
run;


proc means data=PG1.STORM_SUMMARY;
          var  MaxWindMPH MinPressure;
run;

proc univariate data=PG1.STORM_SUMMARY;
          var  MaxWindMPH MinPressure;
run;

/* list unique values and frequencies */
proc freq data=PG1.STORM_SUMMARY;
          tables Basin Type Season;
run;

************************************************************;
*  Filtering Rows with Basic Operators                     *;
```

```
************************************************************;
*  Syntax and Example                                      *;
*                                                          *;
*    WHERE expression;                                     *;
*                                                          *;
*    Basic Operators:                                      *;
*         = , EQ                                           *;
*         ^= , ~= , NE                                     *;
*         > , GT                                           *;
*         < , LT                                           *;
*         >= , GE                                          *;
*         <= , LE                                          *;
*    SAS Date Constant                                     *;
*         "ddmmmyyyy"d ("01JAN2015"d)                      *;
************************************************************;

proc print data=sashelp.cars;
     var Make Model Type MSRP MPG_City MPG_Highway;
     where Type="SUV" and MSRP <= 30000;
run;


************************************************************;
*  Filtering Rows Using Macro Variables                    *;
************************************************************;
*  Syntax and Example                                      *;
*                                                          *;
*    %LET macrovar=value;                                  *;
*                                                          *;
*    Usage:                                                *;
*    WHERE numvar=&macrovar;                               *;
*    WHERE charvar="&macrovar";                            *;
*    WHERE datevar="&macrovar"d;                           *;
************************************************************;

%let CarType=Wagon;

proc print data=sashelp.cars;
     where Type="&CarType";
     var Type Make Model MSRP;
run;

proc means data=sashelp.cars;
     where Type="&CarType";
     var MSRP MPG_Highway;
run;
```

```
proc freq data=sashelp.cars;
    where Type="&CarType";
     tables Origin Make;
run;

************************************************************;
*  Demo                                                    *;
*    1) Highlight the demo program and run the selected    *;
*       code.                                              *;
*    2) Write three %LET statements to create macro        *;
*       variables named WindSpeed, BasinCode, and Date.    *;
*       Set the initial values of the variables to match   *;
*       the WHERE statement.                               *;
*    3) Modify the WHERE statement to reference the macro  *;
*       variables. Highlight the demo program and run the  *;
*       selected code. Verify that the same results are    *;
*       produced.                                          *;
*    4) Change the values of the macro variables to        *;
*       values that you select. Possible values for Basin  *;
*       include NA, WP, SP, WP, NI, and SI. Highlight the  *;
*       demo program and run the selected code.            *;
************************************************************;

proc print data=pg1.storm_summary;
     where MaxWindMPH>=156 and Basin="NA" and StartDate>="01JAN2000"d;
     var Basin Name StartDate EndDate MaxWindMPH;
run;

proc means data=pg1.storm_summary;
     where MaxWindMPH>=156 and Basin="NA" and StartDate>="01JAN2000"d;
     var MaxWindMPH MinPressure;
run;



************************************************************;
*  Formatting Data Values in Results                       *;
************************************************************;
*  Syntax and Example                                      *;
*                                                          *;
*    FORMAT col-name(s) format;                            *;
*                                                          *;
*    <$>format-name<w>.<d>                                 *;
*                                                          *;
*    Common formats:                                       *;
```

```
*       dollar10.2 -> $12,345.67                                *;
*       dollar10.  -> $12,346                                   *;
*       comma8.1   -> 9,876.5                                   *;
*       date7.     -> 01JAN17                                   *;
*       date9.     -> 01JAN2017                                 *;
*       mmddyy10.  -> 12/31/2017                                *;
*       ddmmyy8.   -> 31/12/17                                  *;
************************************************************;

proc print data=pg1.class_birthdate;
     format Height Weight 3. Birthdate date9.;
run;

************************************************************;
*  Identifying and Removing Duplicate Values               *;
************************************************************;
*  Syntax and Example                                      *;
*                                                          *;
*    Remove duplicate rows:                                *;
*    PROC SORT DATA=input-table <OUT=output-table>         *;
*        NODUPRECS <DUPOUT=output-table>;                  *;
*        BY _ALL_;                                         *;
*    RUN;                                                  *;
*                                                          *;
*    Remove duplicate key values:                          *;
*    PROC SORT DATA=input-table <OUT=output-table>         *;
*        NODUPKEY <DUPOUT=output-table>;                   *;
*        BY <DESCENDING> col-name (s);                     *;
*    RUN;                                                  *;
************************************************************;

************************************************************;
*  Demo                                                    *;
*    1) Modify the first PROC SORT step to sort by all     *;
*       columns and remove any duplicate rows. Write the   *;
*       removed rows to a table named STORM_DUPS.          *;
*       Highlight the step and run the selected code.      *;
*       Confirm that there are 107,821 rows in             *;
*       STORM_CLEAN and 214 rows in STORM_DUPS.            *;
*    2) Run the second PROC SORT step and confirm that     *;
*       the first row for each storm represents            *;
*       the minimum value of Pressure.                     *;
*       Note: Because storm names can be reused in         *;
*             multiple years and basins, unique storms     *;
*             are grouped by sorting by Season, Basin,     *;
*             and Name.                                    *;
```

```
*    3) Modify the third PROC SORT step to sort the       *;
*       MIN_PRESSURE table and keep the first row for     *;
*       each storm. You do not need to keep the removed   *;
*       duplicates. Highlight the step and run the        *;
*       selected code.                                    *;
*********************************************************;

*Step 1;
proc sort data=pg1.storm_detail out=storm_clean;
     by _ALL_ ;
run;

*Step 2;
proc sort data=pg1.storm_detail out=min_pressure;
     where Pressure is not missing and Name is not missing;
     by descending Season Basin Name Pressure;
run;

*Step 3;
proc sort data=min_pressure;
    where Pressure is not missing and Name is not missing;
     by descending Season Basin Name Pressure;
run;

*********************************************************;
*  Identifying and Removing Duplicate Values            *;
*********************************************************;
*  Syntax and Example                                   *;
*                                                       *;
*    Remove duplicate rows:                             *;
*    PROC SORT DATA=input-table <OUT=output-table>      *;
*        NODUPRECS <DUPOUT=output-table>;               *;
*        BY _ALL_;                                      *;
*    RUN;                                               *;
*                                                       *;
*    Remove duplicate key values:                       *;
*    PROC SORT DATA=input-table <OUT=output-table>      *;
*        NODUPKEY <DUPOUT=output-table>;                *;
*        BY <DESCENDING> col-name (s);                  *;
*    RUN;                                               *;
*********************************************************;

*********************************************************;
*  Demo                                                 *;
*    1) Modify the first PROC SORT step to sort by all  *;
*       columns and remove any duplicate rows. Write the *;
```

```
*       removed rows to a table named STORM_DUPS.          *;
*       Highlight the step and run the selected code.      *;
*       Confirm that there are 107,821 rows in             *;
*       STORM_CLEAN and 214 rows in STORM_DUPS.            *;
*    2) Run the second PROC SORT step and confirm that     *;
*       the first row for each storm represents            *;
*       the minimum value of Pressure.                     *;
*       Note: Because storm names can be reused in         *;
*             multiple years and basins, unique storms     *;
*             are grouped by sorting by Season, Basin,     *;
*             and Name.                                    *;
*    3) Modify the third PROC SORT step to sort the        *;
*       MIN_PRESSURE table and keep the first row for      *;
*       each storm. You do not need to keep the removed    *;
*       duplicates. Highlight the step and run the         *;
*       selected code.                                     *;
*************************************************************;

*Step 1;
proc sort data=pg1.storm_detail out=storm_clean;
     by _ALL_ ;
run;

*Step 2;
proc sort data=pg1.storm_detail out=min_pressure;
     where Pressure is not missing and Name is not missing;
     by descending Season Basin Name Pressure;
run;

*Step 3;
proc sort data=min_pressure;
    where Pressure is not missing and Name is not missing;
     by descending Season Basin Name Pressure;
run;

*************************************************************;
*  Using Expressions to Create New Columns                  *;
*************************************************************;
*  Syntax and Example                                       *;
*                                                           *;
*    DATA output-table;                                     *;
*        SET input-table;                                   *;
*        new-column = expression;                           *;
*    RUN;                                                   *;
*************************************************************;
```

```
data cars_new;
      set sashelp.cars;
      where Origin ne "USA";
      Profit=MSRP-invoice;
      Source="Non-US Cars";
      format Profit dollar10.;
      keep Make Model MSRP Invoice Profit Source;
run;

*************************************************************;
*  Demo                                                     *;
*    1) Add an assignment statement to create a numeric     *;
*       column named MaxWindKM by multiplying MaxWindMPH    *;
*       by 1.60934.                                         *;
*    2) Add a FORMAT statement to round MaxWindKM to the    *;
*       nearest whole number.                               *;
*    3) Add an assignment statement to create a new         *;
*       character column named StormType that is equal to   *;
*       Tropical Storm. Highlight the DATA step and run     *;
*       the selected code.                                  *;
*************************************************************;

data tropical_storm;
      set pg1.storm_summary;
      drop Hem_EW Hem_NS Lat Lon;
      where Type="TS";
      *Add assignment and FORMAT statements;
run;


*************************************************************;
* Using Character Functions                                 *;
*************************************************************;
*   Syntax and Example                                      *;
*     DATA output-table;                                    *;
*         SET input-table;                                  *;
*         new-column=function(arguments);                   *;
*     RUN;                                                  *;
*                                                           *;
*  Numeric Functions:                                       *;
*    SUM(num1, num2, ...)                                   *;
*    MEAN(num1, num2, ...)                                  *;
*    MEDIAN(num1, num2, ...)                                *;
*    RANGE(num1, num2, ...)                                 *;
*                                                           *;
*  Character Functions:                                     *;
```

```
*    UPCASE(char)                                                *;
*    PROPCASE(char, <delimiters>)                                *;
*    CATS(char1, char2, ...)                                     *;
*    SUBSTR(char, position, <length>)                            *;
************************************************************;

data cars_new;
     set sashelp.cars;
     MPG_Mean=mean(MPG_City, MPG_Highway);
     Type=upcase(Type);
     format MPG_Mean 4.1;
     keep Make Model MSRP Invoice MPG_Mean Type;
run;

************************************************************;
*  Demo                                                    *;
*    1) Add an assignment statement to convert Basin to    *;
*       all uppercase letters using the UPCASE function.   *;
*    2) Add an assignment statement to convert Name to     *;
*       proper case using the PROPCASE function.           *;
*    3) Add an assignment statement to create Hemisphere,  *;
*       which concatenates Hem_NS and Hem_EW using the     *;
*       CATS function.                                     *;
*    4) Add an assignment statement to create Ocean,       *;
*       which extracts the second letter of Basin using    *;
*       the SUBSTR function. Highlight the DATA step and   *;
*       run the selected code.                             *;
************************************************************;

data storm_new;
     set pg1.storm_summary;
     drop Type Hem_EW Hem_NS MinPressure Lat Lon;
     *Add assignment statements;
run;


************************************************************;
*  Enhancing Reports                                       *;
************************************************************;
*  Syntax and Example                                      *;
*                                                          *;
*    TITLEn "title-text";                                  *;
*    FOOTNOTEn "footnote-text";                            *;
*                                                            *;
*    LABEL col-name="label-text"                           *;
*          col-name="label-text";                          *;
```

```
*                                                                   *;
*    Grouped Reports (sort first):                                  *;
*    PROC procedure-name;                                           *;
*        BY col-name;                                               *;
*    RUN;                                                           *;
*******************************************************************;

*Titles and Footnotes;
title1 "Class Report";
title2 "All Students";
footnote1 "Report Generated on 01SEP2018";

proc print data=pg1.class_birthdate;
run;

*Using macro variables;
%let age=13;

title1 "Class Report";
title2 "Age=&age";
footnote1 "Report Generated on %sysfunc(today(),date9.)";

proc print data=pg1.class_birthdate;
     where age=&age;
run;

*Labels;
proc means data=sashelp.cars;
     where type="Sedan";
     var MSRP MPG_Highway;
     label MSRP="Manufacturer Suggested Retail Price"
          MPG_Highway="Highway Miles per Gallon";
run;

*Grouped Report;
proc sort data=sashelp.cars out=cars_sort;
     by Origin;
run;

proc freq data=cars_sort;
     by Origin;
     tables Type;
run;

*******************************************************************;
*  Creating Frequency Reports and Graphs                          *;
```

```
*************************************************************;
*  Syntax and Example                                       *;
*                                                           *;
*    ODS GRAPHICS ON;                                       *;
*    PROC FREQ DATA=input-table <proc-options>;             *;
*        TABLES col-name(s) / options;                      *;
*    RUN;                                                   *;
*                                                           *;
*    PROC FREQ statement options:                           *;
*        ORDER=FREQ|FORMATTED|DATA                          *;
*        NLEVELS                                            *;
*    TABLES statement options:                              *;
*        NOCUM                                              *;
*        NOPERCENT                                          *;
*        PLOTS=FREQPLOT                                     *;
*           (must turn on ODS Graphics)                     *;
*        OUT=output-table                                   *;
*************************************************************;

ods graphics on;

proc freq data=sashelp.heart order=freq nlevels;
     tables Chol_Status / nocum plots=freqplot(orient=horizontal
scale=freq);
run;

*************************************************************;
*  Creating Two-Way Frequency Reports                       *;
*************************************************************;
*  Syntax and Example                                       *;
*                                                           *;
*    PROC FREQ DATA=input-table;                            *;
*        TABLES col-name*col-name </ options>;              *;
*    RUN;                                                   *;
*                                                           *;
*    PROC FREQ statement options:                           *;
*        NOPRINT                                            *;
*    TABLES statement options:                              *;
*        NOROW, NOCOL, NOPERCENT                            *;
*        CROSSLIST, LIST                                    *;
*        OUT=output-table                                   *;
*************************************************************;

title "Blood Pressure by Cholesterol Status";
proc freq data=sashelp.heart;
     tables BP_Status*Chol_Status;
```

```
run;
title;

************************************************************;
*  Demo (Highlight the PROC FREQ step and run              *;
*        the selected code after each step.)               *;
*    1) Highlight the PROC FREQ step, run the selected     *;
*       code, and examine the default results.             *;
*    2) Add the NOPERCENT, NOROW, and NOCOL options in     *;
*       the TABLES statement.                              *;
*    3) Delete the options in the TABLES statement and     *;
*       add the CROSSLIST option.                          *;
*    4) Change the CROSSLIST option to the LIST option in  *;
*       the TABLES statement.                              *;
*    5) Delete the previous options and add                *;
*       OUT=STORMCOUNTS. Add NOPRINT to the PROC FREQ      *;
*       statement to suppress the report.                  *;
************************************************************;


proc freq data=pg1.storm_final;
     tables BasinName*StartDate;
     format StartDate monname.;
     label BasinName="Basin"
           StartDate="Storm Month";
run;

************************************************************;
*  Creating Frequency Reports and Graphs                   *;
************************************************************;
*  Syntax and Example                                      *;
*                                                          *;
*    ODS GRAPHICS ON;                                      *;
*    PROC FREQ DATA=input-table <proc-options>;            *;
*        TABLES col-name(s) / options;                     *;
*    RUN;                                                  *;
*                                                          *;
*    PROC FREQ statement options:                          *;
*        ORDER=FREQ|FORMATTED|DATA                         *;
*        NLEVELS                                           *;
*    TABLES statement options:                             *;
*        NOCUM                                             *;
*        NOPERCENT                                         *;
*        PLOTS=FREQPLOT                                    *;
*           (must turn on ODS Graphics)                    *;
*        OUT=output-table                                  *;
```

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

ods graphics on;

proc freq data=sashelp.heart order=freq nlevels;
     tables Chol_Status / nocum plots=freqplot(orient=horizontal
scale=freq);
run;

************************************************************;
*  Demo                                                    *;
*    1) Highlight the PROC FREQ step and run the selected  *;
*       code. Examine the default results.                 *;
*    2) In the PROC FREQ statement, add the ORDER=FREQ     *;
*       option to sort results by descending frequency.    *;
*       Add the NLEVELS option to include a table with     *;
*       the number of distinct values.                     *;
*    3) Add the NOCUM option in the TABLES statement to    *;
*       suppress the cumulative columns.                   *;
*    4) Change Season to StartDate in the TABLES           *;
*       statement. Add a FORMAT statement to display       *;
*       StartDate as the month name (MONNAME.).            *;
*    5) Add the ODS GRAPHICS ON statement before PROC      *;
*       FREQ. Use the PLOTS=FREQPLOT option in the TABLES  *;
*       statement to create a bar chart. Add the chart     *;
*       options ORIENT=HORIZONTAL and SCALE=PERCENT.       *;
*    6) Add the title Frequency Report for Basin and       *;
*       Storm Month. Turn off the procedure title with     *;
*       the ODS NOPROCTITLE statement. Add a LABEL         *;
*       statement to display BasinName as Basin and        *;
*       StartDate as Storm Month. Clear the titles and     *;
*       turn the procedure titles back on.                 *;
************************************************************;

ods graphics on;
proc freq data=pg1.storm_final;
     tables BasinName Season;
run;

************************************************************;
*  Exporting Results to Excel                              *;
************************************************************;
*  Syntax and Example                                      *;
*                                                          *;
*    ODS EXCEL FILE="filename.xlsx" <STYLE=style>          *;
*              <OPTIONS (SHEET_NAME='label')>;             *;
```

```
*        /* SAS code that produces output */              *;
*    ODS EXCEL OPTIONS (SHEET_NAME='label');              *;
*        /* SAS code that produces output */              *;
*    ODS EXCEL CLOSE;                                     *;
************************************************************;

************************************************************;
*  Demo                                                    *;
*  1) Add an ODS statement to create an Excel file named   *;
*     wind.xlsx in the output folder of the course files.  *;
*     Close the excel desination at the end of the         *;
*     program. Highlight the demo program and run the      *;
*     selected code.                                       *;
*  2) Open the Excel file.                                 *;
*     * SAS Studio: Navigate to the output folder in the   *;
*       Files and Folders section of the navigation pane.  *;
*       Select wind.xlsx click Download.                   *;
*     * Enterprise Guide: Click the Results -> Excel tab   *;
*     and click Download.                                  *;
*  3) Examine the Excel workbook. Notice the light blue    *;
*     background in the results generated by the default   *;
*     style. Also notice the default spreadsheet names.    *;
*     Close the Excel file.                                *;
*  4) Examine the available style options.                 *;
*     * SAS Studio: Submit the following program:          *;
*        proc template;                                    *;
*            list styles;                                  *;
*        run;                                              *;
*     * Enterprise Guide: Select Tools -> Style Manager.   *;
*  5) Change the style by adding the STYLE=SASDOCPRINTER   *;
*     option in the first ODS statement.                   *;
*  6) Use the SHEET_NAME= option on the first ODS EXCEL    *;
*     statement to name the first worksheet Wind Stats.    *;
*     Add another ODS EXCEL statement with the SHEET_NAME=*;
*     option before the second TITLE statement and SGPLOT  *;
*     step. Name the second worksheet Wind Distribution.   *;
*     Highlight the demo program and run the selected      *;
*     code. Open the Excel file to view the results.       *;
************************************************************;

*Add ODS statement;

title "Wind Statistics by Basin";
ods noproctitle;
proc means data=pg1.storm_final min mean median max maxdec=0;
    class BasinName;
```

```
    var MaxWindMPH;
run;

title "Distribution of Maximum Wind";
proc sgplot data=pg1.storm_final;
    histogram MaxWindMPH;
    density MaxWindMPH;
run;
title;
ods proctitle;
*Add ODS statement;

*************************************************************;
*  Exporting Results to PDF                                 *;
*************************************************************;
*  Syntax                                                   *;
*                                                           *;
*    ODS PDF FILE="filename.xlsx" STYLE=style               *;
*            STARTPAGE=NO PDFTOC=1;                         *;
*    ODS PROCLABEL "label";                                 *;
*        /* SAS code that produces output */                *;
*    ODS PDF CLOSE;                                         *;
*************************************************************;

*************************************************************;
*  Demo                                                     *;
*    1) Run the program and open the PDF file to examine    *;
*       the results. Notice that bookmarks are created,     *;
*       and they are linked to each procedure's output.     *;
*    2) Add the STARTPAGE=NO option to eliminate page       *;
*       breaks between procedures. Add the STYLE=JOURNAL    *;
*       option.                                             *;
*    3) To customize the PDF bookmarks, add the PDFTOC=1    *;
*       option to ensure that bookmarks are expanded only   *;
*       one level when the PDF is opened. To customize      *;
*       the bookmark labels, add the ODS PROCLABEL          *;
*       statement before each PROC with custom text. Run    *;
*       the program and open the PDF file.                  *;
*************************************************************;

ods pdf file="&outpath/wind.pdf";
ods noproctitle;

title "Wind Statistics by Basin";
proc means data=pg1.storm_final min mean median max maxdec=0;
    class BasinName;
```

```
    var MaxWindMPH;
run;

title "Distribution of Maximum Wind";
proc sgplot data=pg1.storm_final;
    histogram MaxWindMPH;
    density MaxWindMPH;
run;
title;

ods pdf close;

************************************************************;
*  Reading and Filtering Data with SQL                     *;
************************************************************;
*  Syntax and Example                                      *;
*                                                          *;
*    PROC SQL;                                             *;
*        SELECT col-name, col-name FORMAT=fmt              *;
*        FROM input-table                                  *;
*        WHERE expression                                  *;
*        ORDER BY col-name <DESC>;                         *;
*    QUIT;                                                 *;
*                                                          *;
*    New column in SELECT list:                            *;
*    expression AS col-name                                *;
************************************************************;

proc sql;
select Name, Age, Height*2.54 as HeightCM format 5.1,
       Birthdate format=date9.
    from pg1.class_birthdate
    where age > 14
    order by Height desc;
quit;

************************************************************;
*  Reading and Filtering Data with SQL                     *;
************************************************************;
*  Syntax and Example                                      *;
*                                                          *;
*    PROC SQL;                                             *;
*        SELECT col-name, col-name FORMAT=fmt              *;
*        FROM input-table                                  *;
*        WHERE expression                                  *;
*        ORDER BY col-name <DESC>;                         *;
```

```
*    QUIT;                                                      *;
*                                                               *;
*    New column in SELECT list:                                 *;
*    expression AS col-name                                     *;
***************************************************************;

proc sql;
select Name, Age, Height*2.54 as HeightCM format 5.1,
       Birthdate format=date9.
    from pg1.class_birthdate
    where age > 14
    order by Height desc;
quit;

***************************************************************;
*  Demo                                                       *;
*    1) Add a SELECT statement to retrieve all columns        *;
*       from PG1.STORM_FINAL. Highlight the step and run      *;
*       the selected code. Examine the log and results.       *;
*    2) Modify the query to retrieve only the Season,         *;
*       Name, StartDate, and MaxWindMPH columns. Format       *;
*       StartDate with MMDDYY10. Highlight the step and       *;
*       run the selected code.                                *;
*    3) Modify Name in the SELECT clause to convert the       *;
*       values to proper case.                                *;
*    4) Add a WHERE clause to include storms during or        *;
*       after the 2000 season with MaxWindMPH greater         *;
*       than 156.                                             *;
*    5) Add an ORDER BY clause to arrange rows by             *;
*       descending MaxWindMPH, and then by Name.              *;
*    6) Add TITLE statements to describe the report.          *;
*       Highlight the step and run the selected code.         *;
***************************************************************;

Title "Max Wind Speed after 2000";
proc sql;
*Add SELECT statement;
select Season, propcase(Name) as Name, StartDate format=mmddyy10.,
MaxWindMPH
from PG1.STORM_FINAl
where Season > 2000 and MaxWindMPH>156
order by MaxWindMPH desc;
quit;


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

```
*  Joining Tables with PROC SQL                                  *;
*****************************************************************;
*  Syntax and Example                                            *;
*                                                                *;
*    PROC SQL;                                                   *;
*        SELECT col-name, col-name                               *;
*        FROM input-table1 INNER JOIN input-table2               *;
*        ON table1.col-name=table2.col-name;                     *;
*    QUIT;                                                       *;
*****************************************************************;

proc sql;
select class_update.Name, Grade, Age, Teacher
    from pg1.class_update inner join pg1.class_teachers
    on class_update.Name=class_teachers.Name;
quit;

*****************************************************************;
*  Demo                                                          *;
*    1) Open PG1.STORM_SUMMARY and PG1.STORM_BASINCODES          *;
*       and compare the columns. Identify the matching           *;
*       column.                                                  *;
*    2) Add PG1.STORM_BASINCODES to the FROM clause to           *;
*       perform an inner join on Basin. Remember to              *;
*       qualify the columns as table-name.col-name in the        *;
*       ON expression.                                           *;
*    3) Add the BasinName column to the query after              *;
*       Basin. Highlight the step, run the selected code,        *;
*       and examine the log. Why does the program fail?          *;
*    4) Modify the query to qualify the Basin column in          *;
*       the SELECT clause. Highlight the step and run the        *;
*       selected code.                                           *;
*****************************************************************;

proc sql;
select Season, Name, Basin, MaxWindMPH
    from pg1.storm_summary
    order by Season desc, Name;
quit;
```