**DOCUMENTATION OF SAS PROGRAMMING ESSENTIALS 1:**

**10 January, 2021**

**Lessons:**

1. **Course Overview**
2. **Essentials**
3. **Accessing the data**
4. **Exploring & validating the data**
5. **Preparing data**
6. **Analyzing & Reporting on Data**
7. **Exploring Results**
8. **Using SQL in SAS**

**Lesson:1 Essentials & Overview:**

Programming Interfaces:

1. SAS studio
2. SAS windowing environment
3. SAS enterprise Guide

Features in SAS studio & enterprise

1. Code implementaion
2. Formating
3. Syntax Coloring

**SAS program structure:**

Consists of two main steps

1. Data step  
   a. read datas from the input source

b. filter rows, computing new columns, join tables, data manipulation

1. Procedure step:

Example:

data myclass;

set sashelp.class;

run;

proc print data=myclass;

run;

**Basic code statement layout:**

Data statement

Set statement

Assignemnt statement

Run statement

ALL STEPS MUST BE ENDED WITH SEMICOLONS

Global statements: title, options, libname;

Basics:

1. Commenting using \*

/\* codes \*/

The comments need to be ended with semicolon

1. To format the code: can press Ctrl /

Different types of syntax errors

1. Misspelled
2. Unmatched quotes
3. Mssing semicolons
4. Not in valid options

**Lesson:2 ACCESSING THE DATA**

**Types of datas:**

1. Structured datas:

Defined rows & columns

Ex: sas, microcsfot access & ecel fiules

engines enables sas to read

1. Unstructured datas

no columns

text, delimited files, json, weblogs

must be imported

**SAS table**

1. has defined rows & columns
2. Column: variables
3. Row: observation

Required column attributes:

1. Name:

1-32 characters

Starts with letter or underscores

Continues with letter number or underscores

1. Type:

Numeric (0-9),(-) (.) E –(8 bytes—16 digits)

Character: Letter, Numbers, sp. Characters, blanks, dates  
(1-32 char)

Notes:

To give the summary of the data:

proc contents data="path of the file";

run;

**But what is the problem with such hard coded paths:**

1. Change in data location
2. Id the program is long—difficult to provide the path
3. Can have other data types

**To create libraries:**

libname pg1 "path"

(in log)

Libref PG1 was successfully assigned as follows:

Engine: V9

Physical Name: /folders/myfolders/coursefiles/EPG1V2/data

SAS libray: collection of data files with same type & location

Libname libref engine “path”;

Libref:

1. Name of library – (8 char max)
2. Starts with letter or underscore

Engine: set of instructions to read the data

EX: base, excel, teradata

Since it’s a global statement: \*IT DOESN’T NEED run; to execute it\*

How to use it in the path:

Proc contents data=libref.tablename

#Work Library

It’s a temporary library

Contents are deleted at the end of the sas session

It’s a default library

Data=work.test

#sashelp:

Includes sample datas

Data=sashelp.sas

**#use the libraries to read the excel files:xlsx engine reads data from the excel book**

\*to create library;

libname np xlsx "/folders/myfolders/coursefiles/EPG1V2/data/np\_info.xlsx";

\*to make the excel folow the naming convention of sas7;

options validvarname=v7;

\*to read the parks table and display;

proc contents data=np.parks;

run;

\*to clear Np library;

libname np clear;

importing the datas from the unstructred data:

proc import datafile=”path/file” dbms=filtype out=outputtable; replace;

run;

EX:

proc import datafile="/folders/myfolders/coursefiles/EPG1V2/data/storm\_damage.tab"

dbms=tab out=storm\_damage\_tab replace;

run;

proc import creates a copy of the excel file:

data must be reimported if it changes

**LESSON 3: EXPLORING AND VALIDATING**

1. Name of column
2. Types of values the have
3. Evaluate: the need to add

**These procedures help subset, format & sort the datas**

**Exploring data with procedures:**

1. **Proc contents**

**Procedure:**

Print: creates listing of all the rows and columns

Mean: calaculate the summary statistics: Default ( mean, frequency, std dev, min, max)

Univariate: includes more detailed stats of the distribution

The report includes:

1. 5 lowest & highest values, no of ibservations
2. Freq, %, cf, cp

Examples:

/\*list first 20 rows\*/

proc print data=pg1.np\_summary(obs=20);

var Reg Type ParkName DayVisits TentCampers RVCampers;

run;

/\*calculate summary statistics\*/

proc means data=pg1.np\_summary;

var DayVisits TentCampers RVCampers;

run;

/\*examine extreme values\*/

proc univariate data=pg1.np\_summary;

var DayVisits TentCampers RVCampers;

run;

/\*list unique values and frequency counts\*/

proc freq data=pg1.np\_summary;

tables Reg Type;

run;

**#to filter the rows:**

**Proc precedure-name …;**

**Where expression;**

**Run;**

**//expression=column, operator(=,<,>), values**

Character values—case sensitive

Where basin (“si”,”ni”);

Where a>=”01jan2010”d;

Where type=”ts” and hem=”w”;

Special where operators:

where statement use--task

proc print data=pg1.storm\_summary(obs=50);

/\*where MinPressure is missing; /\*same as MinPressure = .

where Type is not missing; /\*same as Type ne " "

where MaxWindMPH between 150 and 155;

where Basin like "\_I";\*/

where Name like "Z%";

run;

**Macrovariables:**

%let BasinCode=SP;

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;

\_\_\_\_\_

proc print data=pg1.np\_summary;

var Type ParkName;

where ParkName like '%Preserve%';

run;

**Formating the columns:**

**Proc print data=inputtable**

**Format col-name format;**

**Run;**

<$>format-name<w>.<d>

$ -character format

<w>total width including decimals & sp. Characters

. period

<d> no of decimal places

Ex: format height weight 3. Birthdate dates9.;

\_\_\_\_

\* Activity 3.05 \*;

\* 1) Highlight the PROC PRINT step and run the \*;

\* selected code. Notice how the values of Lat, Lon, \*;

\* StartDate, and EndDate are displayed in the \*;

\* report. \*;

\* 2) Change the width of the DATE format to 7 and run \*;

\* the PROC PRINT step. How does the display of \*;

\* StartDate and EndDate change? \*;

\* 3) Change the width of the DATE format to 11 and run \*;

\* the PROC PRINT step. How does the display of \*;

\* StartDate and EndDate change? \*;

\* 4) Highlight the PROC FREQ step and run the selected \*;

\* code. Notice that the report includes the number \*;

\* of storms for each StartDate. \*;

\* 5) Add a FORMAT statement to apply the MONNAME. \*;

\* format to StartDate and run the PROC FREQ step. \*;

\* How many rows are in the report? \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc print data=pg1.storm\_summary(obs=20);

format Lat Lon 4. StartDate EndDate date11.;

run;

proc freq data=pg1.storm\_summary order=freq;

tables StartDate;

\*Add a FORMAT statement;

format StartDate MONNAME.;

run;

#sorting the datas & removing the duplicates:

1.improve visual arrangementof the datas:

2. identify & remove duplicate rows:

Proc sort data=input-table <out=output-table>

Nodupkey <dupout=output-table>

by desc col-names;

Run;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Activity 3.06 \*;

\* 1) Modify the OUT= option in the PROC SORT statement \*;

\* to create a temporary table named STORM\_SORT. \*;

\* 2) Complete the WHERE and BY statements to answer \*;

\* the following question: Which storm in the North \*;

\* Atlantic basin (NA or na) had the strongest \*;

\* MaxWindMPH? \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc sort data=pg1.storm\_summary out=storm\_sort;

where Basin in ("NA","na");

by descending MaxWindMPH;

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* LESSON 3, PRACTICE 8 \*;

\* a) Modify the PROC SORT step to read PG1.NP\_SUMMARY \*;

\* and create a temporary sorted table named \*;

\* NP\_SORT. \*;

\* b) Add a BY statement to order the data by Reg and \*;

\* descending DayVisits. \*;

\* c) Add a WHERE statement to select Type equal to NP. \*;

\* Submit the program. \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc sort data=pg1.np\_summary out=np\_sort;

by Reg descending DayVisits;

where type="NP";

run;

**LESSON 4. PREPARING THE DATA**

**Data manipulations like:**

1. Filtering rows & columns
2. Computing columns
3. Conditional processing

Compilation----------------------🡪>>>>>>>>>>>>>> execution

Compilation:

1. Check the syntax for the errors
2. Identify colum attributes
3. Establish new table metadata

Execution:

1. Read & unit data
2. Perform data manipulation

Data step processing:

DATA step execution is like an automatic loop. The first time through the DATA step, the SET statement reads row number one from the input table and then processes any other statements in sequence, manipulating the values within that row. When SAS reaches the end of the DATA step, there is an implied OUPUT action, and the new row is written to the output table.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* LESSON 4, PRACTICE 1 \*;

\* a) Open the PG1.EU\_OCC table and examine the column \*;

\* names and values. \*;

\* b) Modify the code to create a temporary table named \*;

\* EU\_OCC2016 and read PG1.EU\_OCC. \*;

\* c) Complete the WHERE statement to select only the \*;

\* stays that were reported in 2016. Notice that \*;

\* YearMon is a character column and the first four \*;

\* positions represent the year. \*;

\* d) Complete the FORMAT statement in the DATA step to \*;

\* apply the COMMA17. format to the Hotel, \*;

\* ShortStay, and Camp columns. \*;

\* e) Complete the DROP statement to exclude Geo from \*;

\* the output table. \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

data eu\_occ2016;

set pg1.eu\_occ;

where YearMon like "2016%";

format Hotel ShortStay Camp COMMA17.;

drop Geo;

run;

**Using assignments to create new column:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Activity 4.04 \*;

\* 1) Add an assignment statement to create StormLength \*;

\* that represents the number of days between \*;

\* StartDate and EndDate. \*;

\* 2) Run the program. In 1980, how long did the storm \*;

\* named Agatha last? \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

data storm\_length;

set pg1.storm\_summary;

drop Hem\_EW Hem\_NS Lat Lon;

\*Add assignment statement;

StormLength=EndDate-StartDate;

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Activity 4.06 \*;

\* 1) Add a WHERE statement that uses the SUBSTR \*;

\* function to include rows where the second letter \*;

\* of Basin is P (Pacific ocean storms). \*;

\* 2) Run the program and view the log and data. How \*;

\* many storms were in the Pacific basin? \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Syntax \*;

\* SUBSTR (char, position, <length>) \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

data pacific;

set pg1.storm\_summary;

drop Type Hem\_EW Hem\_NS MinPressure Lat Lon;

\*Add a WHERE statement that uses the SUBSTR function;

where substr(Basin,2,1)="P";

run;

**Using numeric functions:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* LESSON 4, PRACTICE 4 \*;

\* a) Create a new column named SqMiles by multiplying \*;

\* Acres by .0015625. \*;

\* b) Create a new column named Camping as the sum of \*;

\* OtherCamping, TentCampers, RVCampers, and \*;

\* BackcountryCampers. \*;

\* c) Format SqMiles and Camping to include commas and \*;

\* zero decimal places. \*;

\* d) Modify the KEEP statement to include the new \*;

\* columns. Run the program. \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

data np\_summary\_update;

set pg1.np\_summary;

keep Reg ParkName DayVisits OtherLodging Acres SqMiles Camping;

SqMiles=Acres\*.0015625;

Camping=sum(OtherCamping,TentCampers,

RVCampers,BackcountryCampers);

format SqMiles comma6. Camping comma10.;

run;

libname pg1 "/folders/myfolders/coursefiles/EPG1V2/data";

data storm\_cat;

set pg1.storm\_summary;

keep Name Basin MinPressure StartDate PressureGroup;

\*add ELSE keyword and remove final condition;

if MinPressure=. then PressureGroup=.;

else if MinPressure<=920 then PressureGroup=1;

else PressureGroup=0;

run;

proc freq data=storm\_cat;

tables PressureGroup;

run;

**Conditional Processing:**

**If… then…;**

\* Activity 4.08 \*;

\* 1) Run the program and examine the results. Why is \*;

\* Ocean truncated? What value is assigned when \*;

\* Basin='na'? \*;

\* 2) Modify the program to add a LENGTH statement to \*;

\* declare the name, type, and length of Ocean \*;

\* before the column is created. \*;

\* 3) Add an assignment statement after the KEEP \*;

\* statement to convert Basin to uppercase. Run the \*;

\* program. \*;

\* 4) Move the LENGTH statement to the end of the DATA \*;

\* step. Run the program. Does it matter where the \*;

\* LENGTH statement is in the DATA step? \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Syntax \*;

\* LENGTH char-column $ length; \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

data storm\_summary2;

set pg1.storm\_summary;

\*Add a LENGTH statement;

length Ocean $8;

keep Basin Season Name MaxWindMPH Ocean;

\*Add assignment statement;

Basin=upcase(Basin);

OceanCode=substr(Basin,2,1);

if OceanCode="I" then Ocean="Indian";

else if OceanCode="A" then Ocean="Atlantic";

else Ocean="Pacific";

run;

if-then/do

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

data front rear;

set sashelp.cars;

if DriveTrain="Front" then do;

DriveTrain="FWD";

output front;

end;

else if DriveTrain='Rear' then do;

DriveTrain="RWD";

output rear;

end;

run;

if else

LESSON 4, PRACTICE 7 \*;

\* a) Submit the program and view the generated output. \*;

\* b) In the DATA step, use IF-THEN/ELSE statements to \*;

\* create a new column, ParkType, based on the value \*;

\* of Type. \*;

\* NM -> Monument \*;

\* NP -> Park \*;

\* NPRE, PRE, or PRESERVE -> Preserve \*;

\* NS -> Seashore \*;

\* RVR or RIVERWAYS -> River \*;

\* c) Modify the PROC FREQ step to generate a frequency \*;

\* report for ParkType.

data park\_type;

set pg1.np\_summary;

length ParkType $ 8;

if Type='NM' then ParkType='Monument';

else if Type='NP' then ParkType='Park';

else if Type in ('NPRE', 'PRE', 'PRESERVE') then

ParkType='Preserve';

else if Type in ('RVR', 'RIVERWAYS') then ParkType='River';

else if Type='NS' then ParkType='Seashore';

run;

proc freq data=park\_type;

tables ParkType;

run;

title "Storm Analysis";

title2 "Summary Statistics for MaxWind and MinPressure";

proc means data=pg1.storm\_final;

var MaxWindMPH MinPressure;

run;

title2 "Frequency Report for Basin";

proc freq data=pg1.storm\_final;

tables BasinName;

run;

**LESSON 5: ANALYSING & REPORTING DATA:**

1. **Enhancing the reports with titles, footnotes & labels**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Activity 5.03 \*;

\* 1) Modify the LABEL statement in the DATA step to \*;

\* label the Invoice column as Invoice Price. \*;

\* 2) Run the program. Why do the labels appear in the \*;

\* PROC MEANS report but not in the PROC PRINT \*;

\* report? Fix the program and run it again. \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

data cars\_update;

set sashelp.cars;

keep Make Model MSRP Invoice AvgMPG;

AvgMPG=mean(MPG\_Highway, MPG\_City);

label MSRP="Manufacturer Suggested Retail Price"

AvgMPG="Average Miles per Gallon";

Invoice="Invoice Price";

run;

proc means data=cars\_update min mean max;

var MSRP Invoice;

run;

proc print data=cars\_update label;

var Make Model MSRP Invoice AvgMPG;

run;

Write a PROC FREQ step to analyze rows from pg1.np\_species.

Use the TABLES statement to generate a frequency table for Category.

Use the NOCUM option to suppress the cumulative columns.

Use the ORDER=FREQ option in the PROC FREQ statement to sort the results by descending frequency.

Use Categories of Reported Species as the report title.

Submit the program and review the results.

title1 "Categories of Reported Species";

proc freq data=pg1.np\_species order=freq;

tables Category / nocum;

Modify the PROC FREQ step to make the following changes:

Include only the rows where Species\_ID starts with EVER and Category is not Vascular Plant.

Note: EVER represents Everglades National Park.

Turn on ODS Graphics before the PROC FREQ step and turn off the procedure title.

Add the PLOTS=FREQPLOT option to display frequency plots.

Add in the Everglades as a second title.

Submit the program and review the results.

ods graphics on;

ods noproctitle;

title1 "Categories of Reported Species";

title2 "in the Everglades";

proc freq data=pg1.np\_species order=freq;

tables Category / nocum plots=freqplot;

where Species\_ID like "EVER%" and

Category ne "Vascular Plant";

run;

title;

run;

proc means data=pg1.storm\_final maxdec=0 n mean min;

var MinPressure;

where Season >=2010;

class Season Ocean;

ways 1;

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Activity 5.07 \*;

\* Run the program and examine the results to \*;

\* see examples of other procedures that \*;

\* analyze and report on the data. \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

%let Year=2016;

%let basin=NA;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Creating a Map with PROC SGMAP \*;

\* Requires SAS 9.4M5 or later \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*Preparing the data for map labels;

data map;

set pg1.storm\_final;

length maplabel $ 20;

where season=&year and basin="&basin";

if maxwindmph<100 then MapLabel=" ";

else maplabel=cats(name,"-",maxwindmph,"mph");

keep lat lon maplabel maxwindmph;

run;

\*Creating the map;

title1 "Tropical Storms in &year Season";

title2 "Basin=&basin";

footnote1 "Storms with MaxWind>100mph are labeled";

proc sgmap plotdata=map;

\*openstreetmap;

esrimap url='http://services.arcgisonline.com/arcgis/rest/services/World\_Physical\_Map';

bubble x=lon y=lat size=maxwindmph / datalabel=maplabel datalabelattrs=(color=red size=8);

run;

title;footnote;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Creating a Bar Chart with PROC SGPLOT \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

title "Number of Storms in &year";

proc sgplot data=pg1.storm\_final;

where season=&year;

vbar BasinName / datalabel dataskin=matte categoryorder=respdesc;

xaxis label="Basin";

yaxis label="Number of Storms";

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Creating a Line PLOT with PROC SGPLOT \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

title "Number of Storms By Season Since 2010";

proc sgplot data=pg1.storm\_final;

where Season>=2010;

vline Season / group=BasinName lineattrs=(thickness=2);

yaxis label="Number of Storms";

xaxis label="Basin";

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Creating a Report with PROC TABULATE \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc format;

value count 25-high="lightsalmon";

value maxwind 90-high="lightblue";

run;

title "Storm Summary since 2000";

footnote1 "Storm Counts 25+ Highlighted";

footnote2 "Max Wind 90+ Highlighted";

proc tabulate data=pg1.storm\_final format=comma5.;

where Season>=2000;

var MaxWindMPH;

class BasinName;

class Season;

table Season={label=""} all={label="Total"}\*{style={background=white}},

BasinName={LABEL="Basin"}\*(MaxWindMPH={label=" "}\*N={label="Number of Storms"}\*{style={background=count.}}

MaxWindMPH={label=" "}\*Mean={label="Average Max Wind"}\*{style={background=maxwind.}})

ALL={label="Total" style={vjust=b}}\*(MaxWindMPH={label=" "}\*N={label="Number of Storms"}

MaxWindMPH={label=" "}\*Mean={label="Average Max Wind"})/style\_precedence=row;

run;

title;

footnote;

#output the summary table:

Output out= output-table <stat=col-name>;

**LESSON6: EXPORTING THE DATA:**

Syntax:

Proc export data=input-table outfile=”out” <dbms=identifier> <replace>;

Run;

Identifier can be csv, tab, xlsx etc)

**# exporting data to excel sheet:**

Libname xlout xlsx “&outpath/sp.xlsx”;

Data xlout.s\_p;

Set pg1.storm\_final;

Where basin=”SP”;

Run;

**#output delievery system: (ods)**

Ods scvall file=”&outpath/cars.csv”;

Proc print data=sashelp.cars ;

….

Ods csvall close;

**#exporting to powerpoint:**

Ods powerpoint file=”filename.pptx” style=style;

/\* -----------code---------\*?

Ods powerpoint close;

#exporting into document

Ods rtf file=”name” startpage=no;

/\*\*\*\*\*\*code\*\*\*\*\*\*\*/

Ods rtf close;

**LESSON 7: STRUCTURED QUERY LANGUAGE(SQL in SAS)**

PREPARE DATA & ANALYZE THEM

1. Sql syntax:

**Proc sql;**

**Select a, b, c\*2 as c, d format=5.1 birthdate format=date9.**

**From libref.name**

**Where b>10**

**Order by c desc, a;**

**Quit;**

**# creating innerjoins & alias:**

Alias**: from table 1 as alias1 innerjoin table2 as alias 2**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Activity 7.03 \*;

\* 1) Define aliases for STORM\_SUMMARY and \*;

\* STORM\_BASINCODES in the FROM clause. \*;

\* 2) Use one table alias to qualify Basin in the \*;

\* SELECT clause. \*;

\* 3) Complete the ON expression to match rows when \*;

\* Basin is equal in the two tables. Use the table \*;

\* aliases to qualify Basin in the expression. Run \*;

\* the step. \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Syntax \*;

\* FROM table1 AS alias1 INNER JOIN table2 AS alias2 \*;

\* ON alias1.column = alias2.column \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc sql;

select Season, Name, u.Basin, BasinName, MaxWindMPH

from pg1.storm\_summary as u inner join pg1.storm\_basincodes as t

on u.basin=t.basin

order by Season desc, Name;

quit;

alias---innerjoin--sql

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Activity 7.03 \*;

\* 1) Define aliases for STORM\_SUMMARY and \*;

\* STORM\_BASINCODES in the FROM clause. \*;

\* 2) Use one table alias to qualify Basin in the \*;

\* SELECT clause. \*;

\* 3) Complete the ON expression to match rows when \*;

\* Basin is equal in the two tables. Use the table \*;

\* aliases to qualify Basin in the expression. Run \*;

\* the step. \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Syntax \*;

\* FROM table1 AS alias1 INNER JOIN table2 AS alias2 \*;

\* ON alias1.column = alias2.column \*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc sql;

select Season, Name, u.Basin, BasinName, MaxWindMPH

from pg1.storm\_summary as u inner join pg1.storm\_basincodes as t

on upcase(u.basin)=t.basin

order by Season desc, Name;

quit;

**Important: while comparing check the cases.**