

INDEX

~~SAS Programming 1~~

NAME: Chetan STD: _____ SEC: _____ ROLL NO. _____ SUB: _____

| S. No. | Date | THE Essentials | Page No. | Teacher's Sign / Remarks |
|--------|------|-------------------------------|----------|--------------------------|
| 1) | | Essentials | get | |
| 2) | | Accessing Data | get | |
| 3) | | Exploring & Validating Data | get | |
| 4) | | Preparing Data | get | |
| 5) | | Analyzing & Reporting on Data | get | |
| 6) | | Exporting Results | get | |
| 7) | | Using SQL in SAS | get | |

many mistakes were

(classmate's work) true

| S. No. | Date | Title No. | Page No. | Teacher's Sign / Remarks |
|--------|------|-----------|----------|--------------------------|
|--------|------|-----------|----------|--------------------------|

* proc contents data = nadiyal
proc print data = nadiyal
run;

classmate

* RUN, QUIT, DATA and PROC
stat events function as step boundaries which determine when SAS statements take effect and indicate the end of the current step or the beginning of a new step.

* A SAS program will always create a log. A program can create o/p data & results as well depending on steps included.

* All SAS statements must end with a semicolon, but they are free form. You can begin any line anywhere, separate steps and steps with ; or at RUN statement.

* The program contains several statements (a few semi-colons)

proc
contents
data = nadiyal;
proc
print
data = nadiyal;
run;

proc
print
data = nadiyal;
run;

DATA STEP

CLASSNOTE
Date _____
Page _____

* data national

(sit)

* askelp · baseball;

data sit with a
DATA step.

* All numeric columns are 8 bytes by default.

Bat Avg = n hits / n At Bat;

DATA statement indicates data table or an underscore and contains weight sheet will be created or updated

proc. means data = NATIONAL;

var BatAvg;

* The programming interface include

> SAS Enterprise Guide (client application)
> SAS Studio (web-based)
> SAS programming environment.

hands-on

askelp · class

* The library must start with a letter
shares maximum.

The path must also be in quotation marks.

* The SAS help library contains sample and resource tables provided by SAS. The library is automatically available when SAS starts.

CLASSNOTE
Date _____
Page _____

Missing numeric values are represented with a period.

Date _____
Page _____

libname (mydata)

filepath / field data .xlsx"

Date _____
Page _____

SAS dates

- * SAS dates represent the number of days from Jan 1, 1960.

- * SAS date values can be positive or negative.

- * SAS date values can be used in integers, must be provided in quotations marks.

The file header due Excel file in including the .xlsx file quotation marks.

Case 2010

- * SAS columns are either character or numeric. SAS date values are numeric values that represent the number of days before or after Jan 1, 1960. It must have a length of one to eight characters and must begin with a letter or underscore.

The remaining characters must be letters, numbers or underscores.

To dissociate a library

that you previously assigned,

you can use

CLEAR

option in

the LIBNAME statement do
disassociate our assigned library.

```
proc import datafile="d:/collect817/  
bird_count.csv"
```

dbms=csv

out=bird817

replace,

svv;

It creates a SAS table named
Bird817 in the work library from
the CSV file bird_count.

This PROC IMPORT statement
creates a sas table from a
CSV file.

Tip:

When the code runs, it replaces
the SAS table if it already exists.

The descriptor or PROPTION of a
SAS table includes the table

- > name of the table ~~not~~ for now
- > type ~~of~~ the column Salary
- > creation ~~date~~ date of the table

Factory

character format must start with
a dollar sign followed by a letter
or underscore.

A format name does not end
with a period.

The period is a required
delimiter when using a format in
a **FORMAT** statement.

dateT → 25SEP20

dateT → 09/25/20

mmddyy → 09/25/20

The period is a required syntax element
in a format name within a **FORMAT** statement

~~format~~ Order Date date? Delivery date
muddyg8;

where

Style in ('RANCH' IS POSSIBLY)
'TWO STORY'

In the WHERE statement, the

IN operator enables you to select rows based on several values.

by descending. Salary Manager_ID, name,

A ~~multiple~~ ~~allowable~~ constraint

appear after the input table name.
in PROC SORT.

This step sorts Orion.staff by Salary in descending order and by Manager_ID in ascending order.

% let flowers = Plumeria;

The ~~with temporary~~ data set staff that contains the sorted rows will have the name of the new variable is followed by an equal sign & the unquoted value.

The unquoted is added when you use the macro variable.

You use the **VAR** at statement to specify the numeric columns to analyze in **PROC MEANS**. If you don't specify the VAR statement, all numeric columns are analyzed.

You can use the LIBNAME engine to read an Excel worksheet directly or process the data with the DATA step to output data set.

PROC FREQ output includes the distinct values for the column & a frequency count, percent cumulative frequency & percent.

Checking for duplicate errors in the first step in the compilation phase done in the **DATA** step.

Set **onion_sales**, where country = 'US';

Input dataset

The **SET** statement indicates the table that will be used. The **DATA** statement indicates the table that will be created or updated.

$\text{Avg Exp} = \text{mean}(\text{Exp1}, \text{Exp2}, \text{Exp3}, \text{Exp4})$

| Exp1 | Exp2 | Exp3 | Exp4 |
|------|------|------|------|
| 10 | . | 5 | 9 |

The MEAN function ignores missing values, so the calculation is $(10 + 5 + 9) / 3 = 8$

MONTH (SAS-date-value)

The MONTH function returns the month number (1-12) extracted from a SAS date value.

Only three found.
 The character conditions are case sensitive. The first two IF conditions are false. Therefore, the final ELSE statement assigns a edit a value of zero.

data work.compt;
 set onion; sales
 keep emp_id gender
 job_title clwfr;
 run;

DATA CAR-TYPE

CLASSMATE
Date _____
Page _____

```
set as help.caso;
if m$xp > 80000 then
  car-type = "lunux";
else car-type = "regular";
length car-type $ 8;
run;
```

Use a DO group in a DATA step when you want to execute multiple statements for a true IF-THEN expression. To execute more than one statement if a condition is true, you must use IF- THEN / DO groups.

TITLE is the same as TITLE1.

The TITLE statement for the last PROC PRINT step cancels out the higher TITLE statement.

When the DATA step is compiled, the first mention of car-type determines the column name, type and length.

The length is determined by the value in the assignment statement.

The value "lunux" has title1 'The First line';
title2 'The Second line';
proc print data=sales;
run;

The value "regular" has title1 'The Next line';
run; so the length is proc print data=sales;
run;

title 'The Top line';

6.

proc print data= sales;
run;

%let year = 2018;

classmate
Date _____
Page _____

footnote "#Year Sales";

The macro variable Year will be used in double quotes on the FOOTNOTE statement in order for the value to be substituted. If single quotes are used, the value will not be substituted.

proc print data= baseball;
var Name Team BatAvg;

proc means data= baseball;
var BatAvg;
class Team;

The label for BatAvg will appear in the MEANS report.

proc means report.
label WORK.BASEBALL
report running on the PROC PRINT statement.
(not the input table. SASHELP.BASEBALL contains a permanent label for the report by the specified columns.)

label BatAvg = "Batting Average".

run;

classmate
Date _____
Page _____

BatAvg

The BY statement in PROC

CLASSmate
Page:

The NOPROCTITLE option

CLASSmate
Page:

reporting procedure is responsible for grouping the report by the specified columns.

One or multiple columns can be on the BY statement.

The BY statement can be placed in any order within a PROC step.

The BY statement in PROC

~~NOCOM~~ options in the TABLES statement (goes after the forward slash) suppress the display of cumulative frequencies & cumulative levels for all TABLES columns.

SORT is responsible for sorting the table. The NLEVELS option in the PROC MEANS statement creates a table displaying no. of levels for all TABLES columns.

ods noprocditlles;

NO COL

→ removes due
column
per cent.

NO ROW

→ removes the
Row Percent

CROSS LIST

→ statistics values in
columns instead of
stacked in a cell

MEANS

procedure

- the VAR statement is not required
- The MAXDEC= option goes on the PROC MEANS statement.

PROC MEANS steps

- FREQ, BY, and CLASS are automatically included in the proc means statement, the columns would be produced by the OUT= option of the OUTPUT statement.

The INAYS statement

specifies the number of ways to make unique combinations

of class values.

The input table does not have to be pre-coded by the columns

CLASS

statement of due

Output OUT=work.summary Mean(WT)=70.75

classmate
Date _____
Page _____

PROC EXPORT DATA = SASHELP.CARS

DBMS = CSV REPLACE

OUTFILE = "C:\TEMP\CAUSES.CSV";

SUM;

The OUTPUT statement writes statistics to an output table.

OUT = option names the output table.



statistic (INPUT-Variable) - output-variable
can be specified on the OUTPUT statement.



DBMS = option specifies the file type

for the file being created.

libname sales also 'C:\MYDATA\MYDATA.DS1';

data salesq1_2018;
set sasdata.qtr1_2018;

run;

specifies the path to the filename of the external data file being created.

sum;

The DATA statement references

CLASS LIST.

A library is not specified as work in assumed.

WORK.CLASS_LIST is a temporary SAS table.

```
ods excel file="s:/workshop/output/class.xlsx"  
proc print data=sashelp.class;  
run;  
ods excel close;
```

```
The CLEAR option on the  
libname mylib mlib "s:/workshop/output/class.xlsx";  
data mylib.classlist;  
LIBNAME ad statement libname mylib classlist;  
set sashelp.class;
```

```
run;
```

one or more currently assigned
librarys.

for the ODS statement.

The RTF destination will create
a file that can be opened by word
processors.

(c:\mydata\midyear.xlsx';
libname sales clear

SHEET-NAME = is a sub-option
 that goes in a set of parameters
 for due options option.

```
ods csvall file='c:\temp\myfile.csv';
ods pdf file='c:\temp\myfile.pdf';
```

```
ods powerpoint file='c:\temp\myfile.ppt';
```

```
ods rtf file='c:\temp\myfile.rtf';
```

```
ods pdf close;
```

The CLOSE argument closes the destination of the file that is associated with it.

Select in the SELECT clause after the WHERE clause after the column name.

order by

Height / Weight or Ratio

The STYLE = option names the style to use in the output file.

The style controls visual CREATE TABLE NEW-TABLE-NANT AS

aspects such as colors & fonts.

Same before the SELECT

To perform an inner join,

CLASSMATE
Date _____
Page _____

data one;

input Name \$ ID;

Jael 111

Mary 333

Jane 555

specify INNER JOIN between two table names & specify the matching condition in ON clause (not OR).

data two;

input IDNO Salary;
data lines;

111 75000

222 83000

333 82000

Select ID, Name, Salary

from one inner join two

one ID = IDNO;

proc sql

select ID, Name, Salary

from emp saw as e inner join two

phones as p.

SAS Programming

classmate
Date _____
Page _____

20.

Data Manipulation Techniques

LENATH

WHERE

are not processed at execution time.

During the compilation phase of the DATA step (PDV) processed in the compilation phase so that their impact will be observed in the output table.

The DATA step is checked for syntax errors.

The descriptor portion of the output table is created.

During execution, data manipulation occurs in the PDV.

Initial values are assigned to the columns at the beginning of the execution phase of the DATA step.

An implied OUTPUT and RETURN (not REINITIALIZE) occurs at the bottom of the DATA step.

At execution time, the SET statement is proceeded toward data into the PDV.

In the SAS routine to the top of the DATA step, columns read from the input table are reviewed & computed columns are set to missing.

Keep classmate
Date _____
Page _____

The DATA step debugging classification
SAS Cut expire Guide works only
with DATA steps.

data Shipping_Zones; p b r
set Shipping;
Zone=1;

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 3 | 4 |
| 3 | 4 | 5 |
| 4 | 5 | |
| 5 | | |

putlog all; putlog Ratio=;

PUTLOG statement
creates the following
results in the SAS log
run:

Name=Alfred Height=69 Weight=112.5
Ratio=0.61 ERROR=0 N=1

Ratio=0.61

Zone=2;

Rate = (Rate * 1.5);

The explicit OUTPUT statement is

after ZONE=2, so those rows are not
making it to the output table.

An implicit OUTPUT is not at the
bottom of the DATA step due to
the explicit OUTPUT.

ALL is a keyword to show all
of the contents of the PDY.
Ratio= works out the column
name, an equal sign & the
value of Ratio.

Ratio writes out only the value

The KEEP statement controls
columns are in the output table.

| P | B | R | Z |
|---|---|---|---|
| 1 | 2 | 1 | |
| 2 | 3 | 1 | |
| 3 | 4 | 1 | |
| 4 | 5 | | |
| 5 | | | |

Select by year 2018

Date _____
Page _____

The WHERE statement ~~restricts~~ data as it is being read into PDV.

Date _____
Page _____

Ratio is not in the input table so Ratio cannot be specified in the WHERE statement.

To subset Ratio, the condition must be specified in an IF statement.

The BY statement

produces the columns First.Name and Last.Name in the PDV, assuming the input table is sorted by Name. These columns will not be in the output table.

If First.Name & Last.Name are both equal to 1, the Name

value is unique so that row.

It is the first time & the last time that the Name value appears.

→ indicates to continue processing

due last row of a BY group

- ignores missing values
- initially sets the accumulator column to 0, adds a numeric value to the accumulator column
- audibly retains the value of the accumulator column
- of the accumulator column.

The WHERE statement subsets data

at. To do little

If $\text{First} \cdot \text{Gender} = 1$

$\text{TotalInt} = 0;$

~~classmate~~
Page

Mean (of Wednesday)

Numeric columns list are

represents due beginning
of each new B Y q wap. of
regarding to + value of
the accumulation column needs
to be set down pd Jana.

the first 2 last columns are
the range.

A function returns a
value that must be used in
an assignment statement or
expression but as
keyword OF must be
used if a column list is
used has an argument in
a function.

round (Sales, 0)

and continue after
using column names or
permits other define functions
of Sales do two decimal

places.

~~classmate~~
Page

COMPRESS

CLASSmate
Date _____
Page _____

cat(" ", " ") City count

function used to remove the non-numeric symbols in phone

phone = compress(phone, "-.()[]")

The second argument of compress specifies all characters that are defined as the first argument.

phone = compress(phone, " -.-(.)[]")

removes leading & trailing blanks and removes the separator that is defined as the first argument.

use word do specify all

args bala do removed from the values of phone.

phone = compress(phone, " -.-(.)"),

keep only digits

Counting digit (City count, 2, 1),
The result is: no digits in file

divide arguments in value FIND

SCAN of millions of lines.
return the second word using function readLines
only due comma as a delimiter

Manipulating Data with Functions

format Quiz -- AvgQuiz 3.1;

can we a double dash do
represent a physical range of
columns as they are ordered in
the data.

format numeric 3.1; -ALL

to include all numeric
columns with the 3.1 format.

You don't need to use

the OF keyword in the FORMAT
statement. That is a special requirement

when you use column lists as
arguments in a format or

case routine.

data quiz_summary;

set pg2.class_quiz;

AvgQuiz = mean(of Quiz);

format Quiz -- AvgQuiz 3.1;

format numeric 3.1;

run; /* Using a CALL Routine

to Modify Data;

set pg2.class_quiz;

call sortn(of Quiz1-Quiz5);

QuizAvg = mean(of Quiz3-Quiz5);

run;

CALL SORTN sorts the values of
the columns in ascending order.

SUM (n_1, n_2, \dots) calculate classmate
MEAN (n_1, n_2, \dots) date
YEAR (SAS-date) extract info from Date page
MONTH (SAS-date) SAS date values do not include decimal values

TODAY (SAS-date) create date value
MDY (monthly, day, year) integer

NDY (month, day, year) lower limit upper limit

RAND ('distribution', parameters) floor (number)

LARGEST (t, value-1, value-2, ...) round (number, rounding-unit)

SMALLEST (t, value-1, value-2, ...) of days since Jan 1, 1960.

ROUND (number, rounding-unit)

QUIZ2nd = LARGEST (2, QUIZ1 - QUIZ5).

QUIZ3rd = LARGEST (3, QUIZ1 - QUIZ5).

QUIZ3rd = RAND ('integer', 1000, 9999).

TOP3 Avg =

sound (mean (QUIZ1st, QUIZ2nd,

QUIZ3rd)).

Changing Numeric Primitives

Date page

classmate

Date page

classmate

A time value in SAS is stored as the number of seconds from midnight

A datetime value in SAS is stored as the number of seconds from midnight on Jan 1, 1960.

Just like SAS date

Date
CLASSmate

This numeric storage method

> Calculating Date

Date
CLASSmate

enable you to calculate time
between two events or
add by time or date.

INTCK

DATERPART (date/time-value)

> Shifting Date Values

TIMEPART (date/time-value)

INTNX (interval)

Month2Pay = intck('month')

at act

in current,

SerialDate, PayDate)

alignment)

Month2Pay = intck('month')

The INTNX function
shifts dates or times based
on an interval.

SerialDate, PayDate, 'h')

Character Functions

data storm_damage;

Set pg2_storm_damage;

UPCASE

PROPCASE

Keep Event Date AssessmentDate

SUBSTR

AssessmentDate

= intnx('month', Date, -1);

Removing Characters from a string

COMPBL (string)

COMPRESS (string, characters)

STRIP (string)

format Date AssessmentDate

Anniversary =

intnx('year', Date, 10, 'same');

These functions

remove unnecessary characters
from a string.