SAS Report

Safalya Pal - A90555919003 STAT329 - Lab Report

Creating Tables

CODE:

- data keyword is used to define the table name (table names must not contain spaces)
- input keyword is used to define the column names.
- IdNumber 1-4 is used to state that the characters in position 1-4 of each row would fall under the column IdNumber and similarly for Name \$ 6-24. This eliminates the need of delimiters.

- The \$ after Name and Team indicates that values of the Name and Team columns are strings.
- Loss = StartWeight EndWeight is used to defined a new columnamed Loss derived from StartWeight and EndWeight.
- dataline or cards is used to indicate the beginning of the table.
- All command lines end with semicolons ;.

Importing Excel File

```
*Command for importing excel file;

proc import datafile = "/home/u59242738/Data Files/SAS Data1.xlsx"

out=SASOD_data1 dbms = xlsx replace;

run;

*modifing the given data file (creating variables);

Data SASOD_data1modified;

set SASOD_data1;

weightloss=StartWeight - EndWeight;

run;

proc print data=SASOD_data1modified;

run;
Copy

OUTPUT:
```



- We can add a comment line inside the SAS code starting with star (*) and ending with semi-colon (;) in the same line.
- "Proc import datafile=" is used to import a datafile and the position of the file which we want to import is written next to it inside "". We can get the position from datafile properties.
- Out = we give the name to the datafile.
- **Dbms** = for excel file we write xlsx, for csv file we write csv.
- Then we include **replace** so that if by the same name another datafile is saved this new datafile will replace that dataset and then we use semi-colon to end this particular program.
- Then we write **run**; to run this particular lines of codes.
- Then for modifying this data, at first we give a datafile name with the code data new name;
- Then we call the datafile which we want to modify by the code **set** old name.
- In this program we had StartWeight and EndWeight columns and we want a new program by the name **weightloss** and then we will simply just put the formula/equation like **weightloss** = **Startweight EndWeight**.
- Then we put **run**; and run this particular lines of codes.
- We can also print the data in beautiful form with this code **proc print** data = datafile name which we want to see.

CODE:

/* importing file*/

proc import datafile="/home/u59242738/Data Files/SAS Data2.xlsx"

out=data2

```
dbms=XLSX replace;
run;
/* modifying the data*/
data data2_modified;
set data2;
total_NCI = (NCI_2021 + NCI_2020 + NCI_2019 + NCI_2018 + NCI_2017);
total_NCI_2 = sum(NCI_2021, NCI_2020, NCI_2019, NCI_2018, NCI_2017);
average_NCI = total_NCI / 5;
average_NCI_2 = mean(NCI_2021, NCI_2020, NCI_2019, NCI_2018, NCI_2017);
run;
proc print data= data2_modified;
run;
Copy
OUTPUT:
```

Obs	Sr. No	Company	NCI_2021	NCI_2020	NCI_2019	NCI_2018	NCI_2017	total_NCI	total_NCI_2	average_NCI	average_NCI_
- 1	1	RIL	74257	-143583	-53949	-59109	-54949	-237333.00	-237333.00	-47466.60	-47466.6
2	2	ICICI Bank	-54185.5	-36945.4	-24040.8	-38965.6	7000.3	-147137.00	-147137.00	-29427.40	-29427.4
3	3	Axis Bank	-54179.7	-9667.6	-18748.5	-10252.7	-12632.7	-105481.20	-105481.20	-21096.24	-21096.2
- 4	4	IOCL	-22154	-26882.4	-20771.5	-15778.7	-14733.9	-100320.50	-100320.50	-20064.10	-20064.1
5	5	Tata Steel	-13008.5	-17634.7	-16350	-12273.4	-3956.4	-63223.00	-63223.00	a12644.60	a12644.6
61	6	JSW/Steell	-2609-	-49092L	-17432L	-613%	6284	45554 00	46551.00	-9110120	-901002
	2	0000000	-03700002	APRILITY	40000000			4689443	W0000000	41020380	-1100 0.00
		191935 p.	35553-55	_			19999	6662A2556	632052-08	3271.72	22754
24		2000025	336580 SS		.5860E 2	365566	A127 27E	20080150	28(882)37	696868	-676%.C
el/a	66	Nacel Fouds	500,000	J. (87) S.	2666	.0929 C	40024	235W08.5W	250000 SW	400036032	400854
10.00		N82		69295A6	50304.69	DAME NO.	54606.04	CAMPORTO	2000000	0075.73	40000
1260		The ball of the same	40.500	Second	ALCOHOLO.	2000	46080408	70.00000	GOSGS-GO	5000000	2006
200	10	Design Course	4509.85	TO SERVE	AMARINA CO	600000	5960		20000000000000000000000000000000000000	-9678.76	4605.2
	48	Ulital tem			-8094.08	-					
4.5		Hederal Burk	-479/8/28	-4675000TF				A1800 A180		-397E-3F	2000
99		Adjust Marky	75000	-34	3663 C					-5028044	-5028.A
17.		Distributions.	-2000040			-74(0)/43			A\$70.16,000	-8018/91	-60100
et i	-	Àâŋs.	-6269	-100	11 .55%			1000000	40004907	-200	-20003
	- 10	Miser Remerk	200000	-0000000	-34 min	ASS MES	400,000,00	713KB/B	- 18 San Alia	-86 9060	480,006
490		Makek	-1393 67	3825.60	80000	2814.21	49000400	1,000,000,000	4000000	-87875/09	-23104
	20	20	58648A	48%。百	4001.75	469126	3780,33	1056.0	- 7344/9	-2003,06	-9703,0
	21	STATIONS	880926	-070-99	-8685	435634	-5582-19	13/10/20	4000000	-2694/19	-8696.8
22	22	Scottice.	-500007	4195.98	-8575.SE	-2275.5	2572.9	200000	-980079 844	-265631	-9166.5
20	23	Private.	2578.96	1888.88	8370.21	400.33	2626.68	-44600.00	1970010	3360.28	25602
200	26	20808 Lees	-1394.97	8362.05	-00/0.c	-5865.0b.	2000,00	#25692,00	-1900 to (0)	-220020	-22002
1900	367	BESSEL	-73335E	-4576/Z ² 1	-1816/80	2000	456,07	42307.86	4350000	2053.EE	46600
186	288	Mater	503362	-257575;	47.582	5959.50°	2353.25	-590000	490566	5800 TH	-1950/77
-			,	,				,			
25	25	M&M Fin	-7382.59	-2572.71	-1316.85	211.04	-256.47	-11317.58	-11317.58	-2263.52	-2263.5
26	26	MRF	-5063.42	-167.08	-1309.83	-1969.64	-1393.89	-9903.86	-9903.86	-1980.77	-1980.7
27	27	<u> Анцелине</u>	-1155.48	-4689.58	-897.62	422.49	-2474.63	-8894.82	8894.82	-1778.86	-1778.6
1	28-11	Ath Militia	-5980.6	20169	-1580-4	4818	-36656		- 地區級 4 億	476260	4762
20	- 29	letwe	465740	49922	- WAR	-209-65	400000	- PER 1945	47792.90	-1046-57	-KENET
68	23	den	-279259	35057	2000	-202-02	-2542.94	1000.00	-7000 to	496999	-19725
8	221	Charge C	20000	JESSE CO.	.930034iP	-2453 441	-21116ir	280236	-2606.06	4909.00	-49581
124.1	38	Mela	1986.90	4112/195	-012339	462.30	4560 300	-WEI 60	4860.00	-6459/14	-0.656.5
ž.		tarin	-2/202.52	200. Fd	2000	705,32	-24/6/65	.5095.5.1	7383.54	0.006,01	14250
,		keleEdecto	1 .288656	9750.7 F	37204	18892.07		4992.12	4992.53	4592.68	4382.6
			-2000000	20000-04	669.70		207-24	4049.00		-4080-02	
		intel Sometime		The second second					-6846(08		-13665
8		Mad (brees	-2274	-589	894.62	2245100	4541.00	679879	4792,74	-9363:74	-4363.7
ii 💮		\$2.0mg2	-\$69.74	A297 HE	-290A59	-819.95	550.60	-5897.00	5698.69	-2972.78	-21727
*		275 State	-1251.50	-000.00	-03077	-000 M	-5894.54	-600271	all shift	-9978-94	-3678/
	35-1	had be Health	97130	569 (720	20.	2579.7	5093,59	-5003,50	1004.70	46667
	₩.		4770.72	_	-7480 SS	462.88	-275.50	8893	550242	2000	-20075
37	40	Tetrien Irá	-0693	2264	-845	2205.07	7225.97	443.0kg	-5653.4%	4688.68	-4930.0
	67	Афрей Реукца	380,4	9889	4005(4)	3565.06	66479	533.60	多级数	-4008,30	-20004
1	63	Accept Physics	824.28	269,48	-873/36	-4208.99	-580.6	-6269.77	-6169.77	-2632.96	-2002
2		VB Wolse	-920.95	-4200.00	-4002 20	-9539.36	769.10	4025-27	-900122	-1918.26	-49482
*	66		205.6	207,03	6957.95	-593,35	-026,06	6Me21	5274,21	-strate	-1914
**	_	Sta Comme			7460,13	41.23	-584.61	4902.59	4983.59	000.54	400
2//	_	State Comme	2760	243	4629,86	-179/39	0002.64	401818	492.%	922,65	-8600
60-			273.55	47523	-6305:32	-2181	86.02	-0257.28	-4207.39	-207,68	-207.6
-		AT SIG									
38		Special People	1229.25	729.84	4001,83	-695,84	-579.83	-6937.29	4787.29	827,48	-837,4 -829,0
10 7	50		597.39	4836.81	2505.93	7788.27	-08.66	4600.00	-6200.00	829.02	

In this program the things we learnt new are pointed below:

- In case of modifying a datafile, we can simply put the **new_variable_name** = **equation** like previous work or we can also use **new_variable_name** = **formula**.
- Here we used

which is the equation.

but in this case we simply put the \mathbf{sum} formula

• Also in case of finding average we used **mean** formula.

Infile Statement

```
CODE:

/*reading data separated by blank*/

data sdata_blank;

infile "/home/u59242738/Data Files/DATA_blanks.txt";

input Names $ Gender $ Age Weight;

run;

proc print data=sdata_blank;

run;

/*reading Data separated by comma(csv file)*/

Data sdata_commas;

infile "/home/u59242738/Data Files/DATA_commas.csv" dsd;

input Names $ Gender $ Age Weight;

run;

proc print data=sdata_commas;
```

```
run;
/*reading data separated by other deliminators like colon in this case*/
data sdata_otherdel;
infile "/home/u59242738/Data Files/other_del_data.txt" dlm= ":";
input Names $ Gender $ Age Weight;
run;
proc print data= sdata_otherdel;
run;
Copy
OUTPUT:
```



• At first, we give the datafile name we want to give the data for this par-

ticular notebook as data name.

- Next, we write infile "datafile location" to import the datafile.
- In case of this datafiles imported here the variables names were not defined in the text/csv files. So, we used **input** and wrote the variable names one by one.
- In case of character type variables we have to use \$ sign after the variable name to define it as character type and in case of numeric values we don't have to put any sign after the variable names.
- After that, when we imported the same datafile from a csv file instead of a text file we had to include **dsd** after the infile "location" statement. This dsd specifies that when data values are enclosed in quotation marks, delimiters within the value and are treated as character data. The DSD option changes how SAS treats delimiters when you use LIST input and sets the default delimiter to a comma. When we specify DSD, SAS treats two consecutive delimiters as a missing value and removes quotation marks from character values.
- In case of importing a text file where the values are separated with any delimiters like semi-colon, comma, colon, or any others then we simply have to define the delimiter like dlm = ":" after the infile "location" statement.

Importing Text Files with Header

```
CODES:
```

```
/*import txt file with data infile statement(blank delimiter), first row as header*/
data bank_full_blank;
infile "/home/u59242738/Data Files/Bank_full_Blank.txt" firstobs= 2
delimiter= " ";
input age job $ marital $ education $ default $ balance housing $ loan $ contact $ day montrun;
/*import txt file with proc import (blank delimiter) first row as header*/
proc import datafile="/home/u59242738/Data Files/Bank_full_Blank.txt"
```

out=data_full_blank_2_sasdata dbms=dlm replace;

```
delimiter=" ";
getnames=yes; /*including first line as heading in the data*/
run;
Copy
OUTPUT:
```

	marital education	on default	halance housi	no loan contr	ct day month	duration car	nnainn arbus nniann	revious noutcome a
T SS messages	e meded teday	RP	2142 355	PO (2000)	wa S mey	287	and the second	© ⇔Reens se
2. 66 100,000	\$200 \$600000		27 549	50 1585		959	2	9 postopio so
3 33 eccesso	married seconds		2 - 598	yes make		78	2 2	6 missess so
A 47 Sheeps	meded polynom		1806 yes	no min		22	2 4	Q conjugates Q
S SS SECTIONS	STORY STORY		1 50	59 W/675		196	1 .1	9 (15/15/00) 59
S 35 measure		- E-0	237 399	no make		139	2 42	O misseum Do
2) 26 monogon		ne	call yes	300 0000		217	1 1	o sergregate to
62 extregre	directed Sections	yes	2 yes	Dig TOO		360	2 2	g estata
5 57 sefeed	passied places	200	727 760	20 1089		52	2 4	9 primores po
40 Seekald	skyle sessoda		999 - ves	po unko		95	4	0 takeers no
05 05 pens	Grecos secodo		279 266	20 1202		222	5 X 40 F	P protected so
Total age agency	single seconds		200 200	00 9000		200	i st	9 percent to
1 08 Sections	macried seconds		9 799	Dog Talkon	mm S mey	927	8	O miles on
55 mass	marked waterway		79 yes-	50 9000		77	4	9 Williams to
S S services	medical necocia		952 yes	t no makes		1776	8 m	C michagem Do
S 97 refred	named primary	EQ.	722 759	no make	E	353	2 2	0 microsm no
the state of			ag			-88		Access
S Stands	and Alleger			oo marakasida	5	22	e in the last	750000 - 00
	7.5	A 100	60 Supplement	C		3 × 5 =	3 - 2	Total Control
an office and another	estanto selecto	Z- 20	TO VOC	-	and the same		2000	V mikanya sa
TOTAL SHARE	The Section	-	720 16	- The section		20	t	= = ⁷
1 00 market	a market later	-	799 998		200	X84		The monage was
- M- W 300.00	-clade stores	-	25 400	7.	ing S mu	diga.	10.0	9 Wilneston So
de 25 velies	marged recent		200 200	par min	and the same	920		0 Anglesian 20
the sections	DATES CORRECTOR	W. T. 200	The second second	100	Marie A. Salar	400		A ASSOCIATE DE
Obs age job	marital educar	tion default		sing loan con	tact day month	duration co	mpaign pdays p	revious poutcome y
4 th managem	est mouths today	22	2505 793	ಕ್ಕೂ ಪ್ರತಿ	anno S may	228	4 4	y substante se
to the representation	20000 200000	ker o	29 YSS	50 TO	ense & say	989	1 4	g sassans to
Sections of		šey i ze	2 783	yes Euska	saves S may	76	7 4	0 unicema es
4 of thereta	medias victoria	mt * 50	3838 ARE	## E	ense 5 essy	92	1 4	9 sessent co
\$ \$5 \$7505KY	STORY WERE		9 539	D(r \$550)	sees \$ may	35	4 -2	© spanson se
25 management	20,000							Acceptable and
	ect married terming	200	231 yea	00 #20ks	ಞ ಕ್ಷಮ್ಮ ಕ್ಷ ಕ್ಷಮ್ಮ ಕ್ಷಮ್ಮ ಕ್ಷಮ	188	€ 42	6 minous of
2) 2/ 26 mmagan	est married terriery			Anne Control		130 217		
	ent marked terdent mit single terdent	200	23°2 yea	yes * costs			4 42	g cologono og
2) 26 messgen	ent mached tectory mit single tectory	200	231 yea	yes * 0000	aren d'anne	297	1 4 1 1	0 minum ou
1/ 26 mengen 6/ 42 estepas 6/ 50 mind 10/ 43 technique	ent mached tectory mit single tectory	yea y so	237 yea 467 yea 2 yes	00 750 00 750	sem 4 mer ovo 5 mer	217 330	4 42 4 42 4 44	0 minero oc 0 minero oc 0 minero oc 0 minero to 0 minero to
1/ 26 mengen 6/ 42 estepas 6/ 50 mind 10/ 43 technique	ent mached tectory mit single tectory	yea y do	221 yes 2 yes 221 yes 222 yes 223 yes	900 0000 00 9000 00 9000 00 9000	sero S mer	217 352 52	1 4	0 minores on 0 minores on 0 minores on 0 minores on
7 26 manager S 42 manager S 50 minut M 42 minutes	mende setter sept setter sept setter	yea y so	221 yes 2 yes 221 yes 222 yes 223 yes	900 0000 00 9000 00 9000 00 9000	serio 5 may como 5 may	217 300 50 50	1 42 2 -2 1 42 4 45 1 45	9 minores no 2 minores no 3 min
7 25 may	en merket between some one of the control of the co	7 20 20 20 20 20 20 20 20 20 20 20 20 20	221 yas 467 yas 2 yas 221 yas 223 yas 233 yas 233 yas	900 0000 00 9000 00 9000 00 9000	sero 5 mer como 5 mer como 5 mer como 5 mer	297 352 59 55 220	1 42 2 -2 1 42 4 45 1 45	9 minores no 2 minores no 3 min
1/ 26 managem	eri manind redacti ret singe enterpri demond sedacti retario demond single second single second single second	yes y so secy so secy so	22 yes 2 yes 22 yes 22 yes 23 yes 33 yes	905 0000 00 9000 00 9000 00 9000 00 9000	serva 6 men como 5 men como 5 men como 5 men como 5 men como 5 men como 6 men como 6 men	207 822 92 93 94 202 104 105 105 105 105 105 105 105 105 105 105	1 42 2 -2 1 42 4 45 1 45	C malanta and a contract of the contract of th
7 26 manager 6 22 manager 6 25 manager 70 42 manager 70 man	ent market technique rott singe technique met demonst technique method offense strates second demonst second de	7 20 20 20 20 20 20 20 20 20 20 20 20 20	22 yes 22 yes 22 yes 23 yes 23 yes	193 0000 00 750 00 7	serva 6 men como 5 men como 5 men como 5 men como 5 men como 5 men como 6 men como 6 men		4 4 4 1 4 4 1 4 4 1 4 1 4 1 4 1 4 1 4 1	C malaness on on one of the control
7/ 26 management of 22 minutes of 22 minutes of 23 minutes	ent marched ferfacer rest single tention; control of tention; cont	sociation to sociation to sociation	22 Yes 22 Yes 23 Yes 24 Yes 25 Yes 26 Yes 27 Yes 27 Yes	193 0000 00 753 00 230 00 2	serio 4 men como 5 men como 5 men como 5 men como 5 men como 5 men como 5 men como 6 men	277 282 29 20 20 20 20 20 20 21 21 21 21 21 21 21 21 21 21 21 21 21		0 minores ou co
y 26 composition of the composit	medical periodical per	TOTAL	22 yes 22 yes 22 yes 23 yes 23 yes 25 yes 26 yes 27 yes 28 yes 28 yes 28 yes 28 yes 28 yes 28 yes	1905 COSE COSE COSE COSE COSE COSE COSE COSE	serio di meny como 6 many como 6 many como 7 many como	207 202 202 203 203 204 207 207 207 207 207 207 207 207 207 207		C microno co
7 25 CONTROL OF THE PARTY OF TH	control of the contro	y 20 Security 20	22 155 22 155 22 155 23 155 23 155 25	100 100 100 100 100 100 100 100 100 100	sen 4 may com 5 may com 5 may com 5 may com 5 may com 6 may com 6 may com	207		C minores oo
	The second secon	y 200 y 200 Security 200 Se	20 yes 2 yes 20	905 C000 00 905 00 9	Service Company Compan	207		C minores on control of minores o
	and control of control control of control control of control control of control contro	Ten to the control of	20 100 20	905 C000 00 905 00 9	sens 4 may com 5 may com 5 may com 5 may com 6 may	277 222 223 223 223 223 223 223 223 223		C malcona co co construir co co construir co co co construir co co co construir co co co construir co co construir construi
7 25 company	ent marched fections continued to the co	Tool Tool Tool Tool Tool Tool Tool Tool	20 yes 20	900 000 100 100 100 100 100 100 100 100	Service Control of the control of th	277 222 222 222 222 222 222 222 222 222		C minoran co
	and market sections of the control o	Total State of the Control of the Co	722 YES	200 200 200 200 200 200 200 200 200 200	Section 6 may come 5 m	277 222 222 222 222 222 222 222 222 222		C minores co co co construir co co co construir co
	The second secon	Total Section	20 yes 22 yes 20	200 200 200 200 200 200 200 200 200 200	Series 6 may series 5 may series 6 may se	277 202 202 202 203 203 203 203 203		C minores oo
	The state of the s	Teles (1998) Te	20 yes 2 yes 20	900 000 100 100 100 100 100 100 100 100	Sense de margonia de la composición del composición del composición de la composición del composición dela	277 222 222 222 222 222 222 222 222 222		C minores so se

- In case of comment lines inside the codes we can also use / comment / .
- In case of **infile** statement we have to define the names of the variables ourselves and as the first row is our header in the datafile, so we have to define **firstobs** = 2 so that the sas environment takes the values of the variables from the second row ignoring the first row.
- As there is no delimiter used, we will simply define **delimiter** = "" putting a blank inside the quotations.
- Then we will have to manually put the variable names in the input statement.
- In case of importing the same file by proc print statement, we will just simply put a code **getnames** = **yes**; and the sas environment will understand and take that first line as heading or simply variable names.

Various Datafiles

```
CODE:
```

```
/* Reading datafiles in different format using informat statement
SAS has different formats but we'll work with only two kinds
i. standard numeric which is in the format w.d
ii. character value which is in the format $w.
eg, in w.d format, 145 kg is 3. format and 2.1 is 3.1 format*/
data sdata_column;
infile "/home/u59242738/Data Files/DATA_column.txt";
input
@1 Name $5.
@6 Gender $1.
@7 Weight 3.
@10 DOB mmddyy10.;
run;
```

```
/* 1 Jan 1960 = 1, 2 Jan 1960 = 2 and so on for date format*/
/* display DDMonth YYYY format */
proc print data= sdata_column;
format DOB date9.;
run;
/* display MM/DD/YYYY format */
proc print data= sdata_column;
format DOB mmddyy10.;
run;
Copy
OUTPUT:
Obs Name Gender Weight DOB
```

Obs	Name	Gender	Weight	DOB
- 1	Tim	М	145	210CT1978
2	Sara		130	20SEP1964
3	Mike	M	180	23NOV1965
4	Laura	F	130	06NOV1980
5	Sean	M	167	07APR2000

Obs	Name	Gender	Weight	DOB
1	Tim	M	145	10/21/1978
2	Sara		130	09/20/1964
3	Mike	M	180	11/23/1965
4	Laura	F	130	11/06/1980
5	Sean	M	167	04/07/2000

• Torming shapedrting a file through infile statement from a text file

un;

it is a character variable and this 5. Will define that it will take 5 places from 1 till 5.

- Similarly @6 Gender \$1. Will define that Gender variable will start taking values from 6th space till only 1 space from that which means only 6th space and it will be a character type variable.
- Date has many formats in SAS environment.

Normally if we don't define any format it will take 1st January 1960 as 1, 2nd January 1960 as 2 and so on for date format.

But, we can change it according to our needs.

Like, here we used 2 formats.

- i) Format DOB date9. which will show us the date as 01JAN1960 or 11NOV2021 like that.
- ii) Format DOB mmddyy10. which will show us the date as 01/01/1960 or 11/11/2021 like that.

Importing Multiple Excel Sheets

```
/* Importing different sheet other than the first one from Excel to Sas*/
proc import datafile= "/home/u59242738/Data Files/excel_data.xlsx"

out= dataset1 dbms= XLSX replace;
sheet= "name_class";

run;
/* Importing some selected ranges of a sheet from Excel to SAS*/
proc import datafile= "/home/u59242738/Data Files/excel_data.xlsx"

out= dataset1 dbms= XLSX replace;
sheet= "name_class";

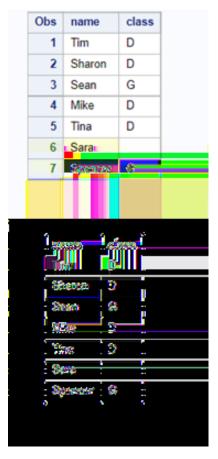
range= "A1:B8";

run;
proc print data= dataset1;
```

```
run;
/* by using noobs we are ignoring the obs no column*/
proc print data= dataset1 noobs;
```

run; Copy

OUTPUT:



EXPLANATION:

• Here, we imported the data file by the same process and just added an extra code line which is

Sheet= "name_class";

By this way, no matter how many sheets are there in that particular excel file, only the sheet named as name_class will be imported.

• Similarly, if we want to take only some particular values/observations from a particular sheet of the excel file we will simply define the range we want to take like

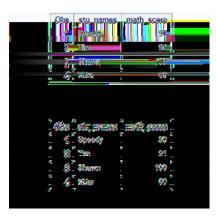
Range= "A1:B8" like that.

• No matter the dataset, sas environment always put observation number by default in the imported dataset. If we want to ignore that observation number column from our output we will add a command **noobs** after the proc print data= dataset_name. That way, when we print our results at the end, the observation number column is not printed.

CREATE A DATAFILE

```
/* Create datafile*/
libname scores "/home/u59242808";
/* Permanent Data File*/
data scores.student_math;
input stu_names $ math_score;
datalines;
Speedy 90
Tim 91
Shawn 100
Mike 60
;
run;
proc print data= scores.student_math;
run;
```

```
/*Temporary data file (data go away the next time i login)*/
data student_math2;
input stu_names $ math_score;
datalines;
Speedy 90
Tim 91
Shawn 100
Mike 60
;
run;
proc print data = student_math2;
run;
Copy
```



OUTPUT:

- $\bullet\,$ We can create a data file by ourselves in SAS environment as well
- i) Permanent datafile
- ii) Temporary datafile.

• In case of creating a permanent datafile we will have to save the datafile in a library.

Here, I saved it in my **scores** library by and saved the datafile as **student_math** by using the code data scores.student_math.

- Then, I have to put the variable names as input
- Then, by defining datalines, we will just simply put the values and observations.
- In case of a temporary datafile, we don't have to define any library name
 and we can simply just start by defining data dataset name as data student_math2 and then put the input variable names and observations.

SAS function

```
CODE:

proc import datafile= "/home/u59242738/Data Files/score_data.xlsx"

out= score_data1 dbms= XLSX replace;

run;

/* converting Gender variable from small to upper letter

another function is used to find out the length of name variable observations*/

data score_data_new;

set score_data1;

gender_func = upcase(gender);

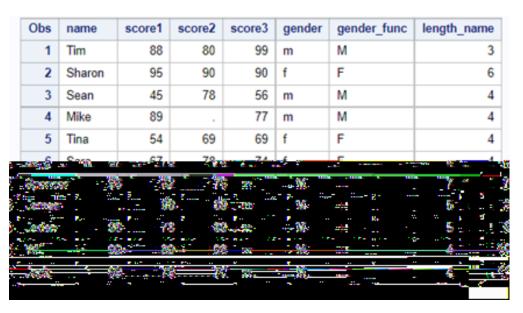
length_name = length(name);

run;

proc print data= score_data_new;

run;
Copy

OUTPUT:
```



- At first we imported the data.
- Then we saw that the gender variable was in lowercase we want the uppercase values and we also need the length of the names.
- So, we created a new variable as **gender_func** and used the upcase statement to convert the alphabets present in the gender to uppercase using the statement

$Gender_func = upcase(gender);$

• We also used length statement to find out the length of the name observations in our data and put the output in a new variable by the name length_name like

 $Length_name = length(name);$

IF-THEN STATEMENT

```
/* using if-then statement*/
data score_data1_ifthen;
set score_data1;
if gender = "m" then gender_num = 1;
```

```
else gender_num = 2;
run;
proc print data= score_data1_ifthen;
run;
Copy
OUTPUT:
```

Obs	name	score1	score2	score3	gender	gender_func	length_name
1	Tim	88	80	99	m	M	3
2	Sharon	95	90	90	f	F	6
3	Sean	45	78	56	m	M	4
4	Mike	89		77	m	M	4
5	Tina	54	69	69	f	F	4
- S	Some	57	Z9 ₂	74	f.	F	4
ed see	Sec. 10	67.	70 a	33-	M.	- Care 2	7. "
22300	7		0. 0.	- Trees	36	±±± Re	
erner erien	8	B 70		A Later	% - W	±i± Re ±i±	5

- We used the previous dataset here and that's why we didn't have to import the datafile again here.
- Here we want to convert the gender m-f values to binary form.

So, we used if then statement here and made a new variable as gender_num which will give us value 1 if gender is mail and 2 if gender is female.

• By using if then statement we wrote here if gender = "m" then gender_num = 1 else gender_num=2 and then printed the output.

IF ELSE THEN STATEMENT

CODE:

proc import datafile= "/home/u59242738/Data Files/score_data_miss.xlsx"

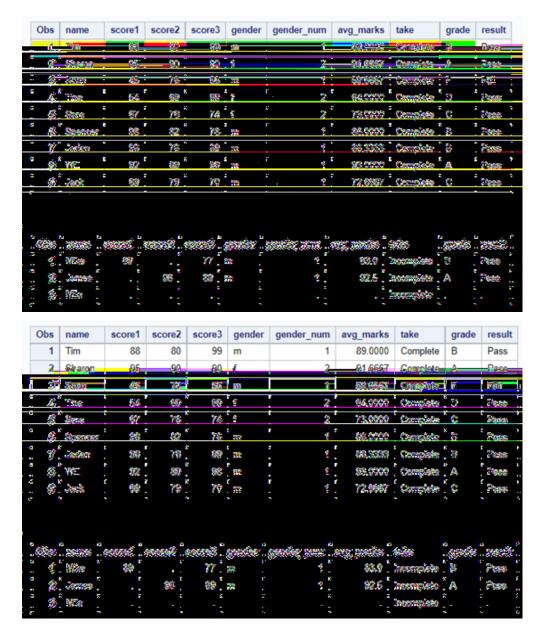
```
out= score_data_miss dbms= xlsx replace;
run;
data score_data_miss_cond;
set score_data_miss;
if gender = "m" then gender_num = 1;
else if gender = "f" then gender_num = 2;
else gender_num = .;
avg_marks = mean(score1, score2, score3);
run;
proc print data= score_data_miss_cond;
run;
data score_data_miss_cond2;
set score_data_miss_cond;
length take $12;
if avg_marks >= 90 then do
 grade = "A";
 result = "Pass";
 end;
else if avg_marks >= 80 then do
 grade = "B";
 result = "Pass";
 end;
else if avg_marks >= 70 then do
```

```
grade = "C";
 result = "Pass";
 end;
else if avg_marks >= 60 then do
 grade = "D";
 result = "Pass";
 end;
else if 0 <= avg_marks < 60 then do
 grade = "F";
result = "Fail";
 end;
else do
grade = ".";
result = ".";
 end;
/* create two variables i) complete when all the marks are given and ii) pass if the average
if score1 ~=. and score2 ~=. and score3 ~=. then take = "Complete";
else take = "Incomplete";
run;
proc print data= score_data_miss_cond2;
run;
/* Divide the file into two parts using if statement where part 1= who complete the assignment
data subset_1;
```

```
set score_data_miss_cond2;
if take = "Complete";
run;
data subset_2;
set score_data_miss_cond2;
if take = "Incomplete";
run;
proc print data= subset_1;
proc print data= subset_2;
run;
/* Divide the file into two parts using delete statement where part 1= who complete the ass.
data subset_3;
set score_data_miss_cond2;
if take = "Incomplete" then delete;
run;
data subset_4;
set score_data_miss_cond2;
if take = "Complete" then delete;
run;
proc print data = subset_3;
proc print data = subset_4;
run;
Сору
```

OUTPUT:

Obs	name	score1	score2	score3	gender	gender_num	avg_marks
1	Tim	88	80	99	m	1	89.0000
2	Sharon	95	90	90	f	2	91.6667
3	Sean	45	78	56	m	1	59.6667
4	Mike	89		77	m	1	83.0000
5	Tina	54	69	69	f	2	64.0000
6	Sara	67	70	.74	f asser	2	72.0000
أزئي	· · ·	Incorrect	- 36	- 20 C	78		⇒ *. 5
lin	S	S. S		76	men.	- T	
8≢.	- William	ACCOUNTS OF		80	- 68	N -	
ВĖ,	. 4	2000	38	78,	339 -	775	3-1
9	. eg 1	ever "	92 -	822	98	:	6
	3 495 5	. x	-	See 5	200	*	
٠		*	- W.				
	* 930 L	VEa ×					
	e			*	*	•	
	me scor		7 7	nder gend		g marks take	grade resi
<u> </u>		88 <u>89</u>	. 100 m		9	COLUMNY DESIGNATION	1 Pen
		95 90	\$0 F	:	2	21.5961 Complete	
3 8 8 8 €	977 ·	45 78			4 .	59.5557 Complete	F Fed
₩. WE	Ch.	(R)	77 500		ē.	83,9990 Incomplet	eo 27 Peo
🥳 😘	* .	54 99	500 f		2	84,0000 Complete	D Fee
6 30	Mi .	67 76	74 9		2.	75.0000 Complete	O Per
11 80	\$ TERROTE	36 82	76 . 50		g *	85,9599 Complete	29 Peo
	200	98	89 m	è	9	92,5000 Incomple	F .
	R	28 78			2	38.3853 Complete	
Ø. ≪8					::		
		92 59	\$6 m		Y	95,0000 Complete	A Per
39 . 11	Ř.	59 79	70 50			72.9997 Complete	C Pea



- Here, at first we imported the datafile which is the first output.
- Then, we used if then statement to convert gender to binary in a new variable **gender_num** and made another variable avg_marks which implies the average of score1, score2, score3 and printed the first output.
- Then, we made 3 new variables take, grade and result.

We take the length of take variable as 12 as by default it takes only 8 length.

We defined if avg_marks \geq 90 then grade will show A and result will show pass

If avg_marks \geq 80 then grade will show B and result will show pass and so on.

If avg_marks lies between 0 to 60 then grade will show F and result will show Fail and at last, when there is no avg_marks given, then grade and result will also show blank.

Later, in case of take variable where there is at least 1 missing value in score1, score2, score3, take will show incomplete and if for any student, all 3 marks are given, it will show complete.

And this result is shown in output2.

- Then we made 2 subset of this output by 2 different processes where 1 subset will show all the student who has take = complete and another output will show who has take = incomplete.
- i) We simply took the dataset and checked if take = complete given, then it will be printed in the subset1.
- ii) If take = incomplete, then those observations will be printed on subset2
- iii) Then again we took the entire dataset and checked if take = incomplete then deleted those values and saved it as subset3.
- iv) Lastly, if take = complete, then deleted those observations and saved it as subset4.

1D ARRAY

CODE:

• ARRAY:

```
proc import datafile= "/home/u59242738/Data Files/score\_data\_miss999.xlsx" \\ out= score\_data\_miss999 \ dbms=xlsx \ replace;
```

run;

 $proc print data = score_data_miss999;$

run;

• One-dimensional ARRAY, converting the 999 values to period(. / space);

data dataset1;

```
set score_data_miss999;
array score_var(3) score1 score2 score3;
do i = 1 to 3;
if score_var(i) = 999 then score_var(i) = .;
end;
run;
proc print data = dataset1;
run;
OUTPUT:
```

Obs	name	score1	score2	score3	gender
1	Tim	88	80	99	m
2	Sharon	95	90	90	f
3	Sean	45	78	56	m
4	Mike	89	999	77	m
5	Tina	54	69	69	f
6	Sara	67	78	74	f
7	Spencer	98	82	78	m
8	James	999	96	89	m
9	Jaden	98	78	89	m
10	Will	92	89	98	m
11	Jack	69	79	70	m
12	Mila	999	999	999	

Obs	name	score1	score2	score3	gender	i
1	Tim	88	80	99	m	4
2	Sharon	95	90	90	f	4
3	Sean	45	78	56	m	4
4	Mike	89		77	m	4
5	Tina	54	69	69	f	4
6	Sara	67	78	74	f	4
7	Services	83	32	<u>7</u> 5	16	d
	STATES.		98=	89	W	4
\$87.	1000	(6) -	K,	86	W	4
	195	92	629	985 -	M	Ą.
55	liggle	98	. 10	70	77	ø
VE a		. *		-	- <u></u>	

- At first, we imported the datafile.
- As we can see that there are some 999 values there which are wrong and we want to convert those to . null values and we used 1Dimensional array here for that purpose.
- We defined a array as score_var here which has 3 variables score1, score2, score3.
- $\bullet\,$ Then for every ith variable of score_var we checked if any value is 999 or

not and if it is 999 then we changed it to "." (null value) and if it is not 999 then we stopped for that particular value and checked the next one.

GIVING TITLE TO A DATASET

```
CODE:
```

```
proc import datafile= "/home/u59242738/Data Files/score_data_miss777.xlsx"

dbms = xlsx out = score_data0 replace;

run;

proc print data= score_data0;

title "missing values are shown as 777";

run;
Copy
```

OUTPUT:

-	missing values are shown as 777								
Obs	name	score1	score2	score3	gender				
- 1	Tim	88	80	99	m				
2	Sharon	95	90	90	f				
3	Sean	45	78	56	m				
4	Mike	89	777	77	m				
5	Tina	54	69	69	f				
6	Sara	67	78	74	f				
7	Spencer	98	82	78	m				
8	James	777	96	89	m				
9	Jaden	98	78	89	m				
10	Will	92	89	777	m				
-11	Jack	69	79	70	m				
12	Mila								

- Here, the dataset is imported through proc print statement.
- Then while print the output data we used a **title** code after the proc print data statement and before running it. That way we can give any title to any dataset.

ADDING LABELS TO THE VARIABLES

- We generally add labels for listing and reporting purpose.
- Syntax for label:

```
label var_name = "Variable name";
or
label var_1 = "variable_label1"
var_2 = "variable_label2"
var_3 = "variable_label3";
```

- there are generally permanent & temporary label
- i) if label statements are included in data step then labels are permanent, whereas,
- ii) if the label statements are included in proc step then labels are temporary.

```
proc import datafile= "/home/u59242738/Data Files/score_data.xlsx"

out = adding_labels dbms=xlsx replace;

run;

data adding_labels_mod;

set adding_labels;

LABEL score1 = "Math score"

    score2 = "Stats score"

    score3 = "English score" ;

run;

proc print data= adding_labels_mod LABEL;

run;

/* using label in proc print statement */

proc print data= adding_labels_mod LABEL;
```

```
LABEL name = "Students name";
run;
proc print data= adding_labels_mod;
run;
/* Splitting the labels */
proc print data= adding_labels_mod label split= "*";
label name = "student*name";
run;
Copy
OUTPUT:
```

Obs	name	Math score	Stats score	English score	gender
1	Tim	88	80	99	m
2	Sharon	95	90	90	f
3	Sean	45	78	56	m
4	Mike	89		77	m
5	Tina	54	69	69	f
6	Sara	67	78	74	f
7	Spencer	98	82	78	m
8	James		96	89	m
9	Jaden	98	78	89	m
10	Will	92	89	98	m
11	Jack	69	79	70	m

Obs	Students name	Math score	Stats score	English score	gender
1	Tim	88	80	99	m
2	Sharon	95	90	90	f
3	Sean	45	78	56	m
4	Mike	89		77	m
5	Tina	54	69	69	f
6	Sara	67	78	74	f
7	Spencer	98	82	78	m
8	James		96	89	m
9	Jaden	98	78	89	m
10	Will	92	89	98	m
11	Jack	69	79	70	m

Obs	name	score1	score2	score3	gender
1	Tim	88	80	99	m
2	Sharon	95	90	90	f
3	Sean	45	78	56	m
4	Mike	89		77	m
5	Tina	54	69	69	f
6	Sara	67	78	74	f
7	Spencer	98	82	78	m
8	James		96	89	m
9	Jaden	98	78	89	m
10	Will	92	89	98	m
11	Jack	69	79	70	m

Obs	student name	Math score	Stats score	English score	gender
1	Tim	88	80	99	m
2	Sharon	95	90	90	f
3	Sean	45	78	56	m
4	Mike	89		77	m
5	Tina	54	69	69	f
6	Sara	67	78	74	f
7	Spencer	98	82	78	m
8	James		96	89	m
9	Jaden	98	78	89	m
10	Will	92	89	98	m
11_	Jack	69	79	70	

 $\bullet~$ At first, we imported the data.

- Then we used the label syntax for sas in our program and gave the label for Math score, Stats score, English score for score1, score2, score3 respectively through the datastep and printed it keeping in mind that we have to add LABEL statement in while printing the dataset.
- Later, we gave the label for name as Students name and printed it.
- Later, we only printed the data not using the LABEL statement in the proc print statement and got the original variable names in our output.
- We can also split the label name in two different lines in the variable column by defining some concatenation as label split and using it in the label name code itself. We used as concatenation here and used label name = studentname and got student and name splitted in two different lines in the variable name space.

ASSIGNING FORMAT INTO VARIABLE

```
proc import datafile= "/home/u59242738/Data Files/score_data_miss.xlsx"

out = data0 dbms = xlsx replace;

run;

data dataset1;

set data0;

total_score = sum(score1, score2, score3);

average_score = mean(score1, score2, score3);

run;

data dataset2;

set dataset1;

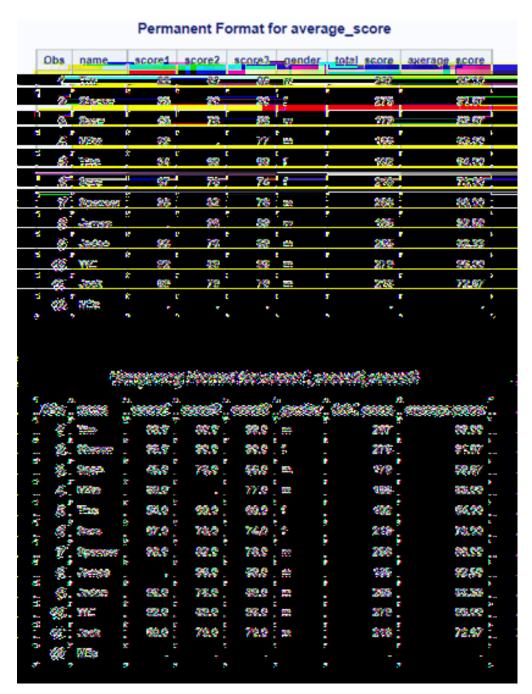
format average_score 5.2;

run;

proc print data = dataset2;

title "Permanent Format for average_score";
```

```
run;
/* Using temporary format in proc print statement */
proc print data= dataset2;
format score1 4.1;
format score2 4.1;
title "Temporary Format for score1, score2, score3";
Copy
RUN;
OUTPUT:
```



• At first, we imported the dataset in SAS environment.

- Later, from score1, score2, score3 we made 2 new variables as total_score and average_score which defines the sum and mean of these 3 score values respectively.
- There we got some average values like 91.66666667, 59.6666667 like that and that's why we used formatting here.
- At first, we permanently formatted the average_score in 5.2 rule by which it will take only 2 decimal places and can take 5 values before the point and printed the first dataset with a title permanent format for average_score.
- Later, we temporary formatted the 3 scores available (score1, score2, score3) in our dataset by 4.1 rule by which it can take 4 number values before point and can take only 1 decimal place and printed it with the title temporary format for score1, score2, score3.

USER DEFINE FORMAT

```
CODE:
```

```
proc import datafile= '/home/u59242738/Data Files/score_data_miss.xlsx'

out= score_data0 dbms= xlsx replace;

run;

data score_data1;

set score_data0;

total_score = sum(score1, score2, score3);

average_score = mean(score1, score2, score3);

run;

proc format;

value $genderf 'f' = 'female'

'm' = 'male';

/* this $genderf variable will be created having the values female and male from f and m resulue as_group 0 - <60 = 'F'

60 - <70 = 'D'</pre>
```

```
70 - <80 = 'C'
80 - <90 = 'B'
90 - High = 'A'
other = 'Missing';
run;
proc print data = score_data1;
format gender $genderf. average_score as_group.;
title "User Define Formal Table"
run;
Copy
OUTPUT:</pre>
```

	U	ser Defi	ine Forr	mal Tab	le run	
name	score1	score2	score3	gender	total_score	average_score
Tim	88	80	99	male	267	В
Sharon	95	90	90	female	275	A
Sean	45	78	56	male	179	F
Mike	89		77	male	166	В
Tina	54	69	69	female	192	D
Sara	67	78	74	female	219	С
Spencer	98	82	78	male	258	В
James		96	68	risile	485	4.
Jeden	99	788	860	DESTR.	2025	29
和产	92	89	*	1112	279	A
2000s	- W	73	79	(1) (2)	260	ě.
Wile.		_	_			A FEMALES
	Tim Sharon Sean Mike Tina Sara Spencer James James	name score1 Tim 88 Sharon 95 Sean 45 Mike 89 Tina 54 Sara 67 Spencer 98 James 98 James 98 James 98	name score1 score2 Tim 88 80 Sharon 95 90 Sean 45 78 Mike 89 . Tina 54 69 Sara 67 78 Spencer 98 82 James 98 32 James 98 28 James 98 32 James 98 32	name score1 score2 score3 Tim 88 80 99 Sharon 95 90 90 Sean 45 78 56 Mike 89 . 77 Tina 54 69 69 Sara 67 78 74 Spencer 98 82 78 James 98 83 Jacks 28 28 Jacks 28 28 Jacks 28 28	name score1 score2 score3 gender Tim 88 80 99 male Sharon 95 90 90 female Sean 45 78 56 male Mike 89 . 77 male Tina 54 69 69 female Sara 67 78 74 female Spencer 98 82 78 male James 98 38 male <td>Tim 88 80 99 male 267 Sharon 95 90 90 female 275 Sean 45 78 56 male 179 Mike 89 . 77 male 166 Tina 54 69 69 female 192 Sara 67 78 74 female 219 Spencer 98 82 78 male 258 James 98 82 78 male 258 James 98 82 78 male 258 James 98 82 78 male 258</td>	Tim 88 80 99 male 267 Sharon 95 90 90 female 275 Sean 45 78 56 male 179 Mike 89 . 77 male 166 Tina 54 69 69 female 192 Sara 67 78 74 female 219 Spencer 98 82 78 male 258 James 98 82 78 male 258 James 98 82 78 male 258 James 98 82 78 male 258

- At first, we imported the dataset and find out the total score and average score in numerical values.
- Now, we had 'm' and 'f' values in gender variable and numerical mean

values in average_score variable.

- We used proc format here and defined gender f value as female and m value as male.
- Then we took value from average_score and took them in group values like if the average_score is between 0 to 60 then it will return F, if its between 60 to 70 it will return D, like that we converted the average scores to grades and if there is any null value in average—score it will return missing.
- Then we printed our data with proc print statement and used format statement to change these given gender values to formatted genderf character values and changed average_score to as_group which we defined and also gave a title to our dataset and printed it.

STORING AND REFERING AND USING USER DEFINE FORMAT

```
libname myfmts "/home/u59242738/SAS_code";
proc format library= myfmts;
value $genderf 'm' = 'Male'
  'f' = 'Female';
value asgroup 0-<60 = 'F'
  60-<70 = 'D'
  70-<80 = 'C'
  80-<90 = 'B'
  90- High = 'A'
  other = 'Missing';
run;
proc format library= myfmts fmtlib;
run;</pre>
```

```
proc import datafile= "/home/u59242738/Data Files/score_data_miss.xlsx"

out = scoredata0 dbms= xlsx replace;

run;

data scoredata1;

set scoredata0;

total_score = sum(score1, score2, score3);

avg_score = mean(score1, score2, score3);

libname myfmts "/home/u59242738/SAS_code";

option fmtsearch = (myfmts work library);

proc print data= scoredata1;

format gender $genderf. avg_score asgroup.;

run;
Copy
OUTPUT:
```

```
FORMAT NAME: ASGROUP LENGTH: 7 NUMBER OF VALUES: 6
| MIN LENGTH: 1 MAX LENGTH: 40 DEFAULT LENGTH: 7 FUZZ: STD
|-----
START
      END
          |LABEL (VER. V7|V8 17NOV2021:02:47:24)|
       60<F
     01
      60 I
            70<D
      701
            80<C
     801
            90<B
     90|HIGH
      |**OTHER**
| **OTHER**
             Missing
.....
   TORMAT NAME: SCENARS LENGTH: C NUMBER OF HALLESS.
 8350 (30)
Section 1
        74
```

Obs	name	score1	score2	score3	gender	total_score	avg_score
- 1	Tim	88	80	99	Male	267	Е
2	Sharon	95	90	90	Female	275	A
3	Sean	45	78	56	Male	179	F
4	Mike	89		77	Male	166	E
5	Tina	54	69	69	Female	192	
6	Sara	67	78	74	Female	219	(
7	Spencer	98	82	78	Male	258	E
8	James		96	89	Male	185	
9	Jaden	98	78	89	Male	265	
10	Will	92	89	98	Male	279	
11	Jack	69	79	70	Male	218	(
12	Mila						Missing

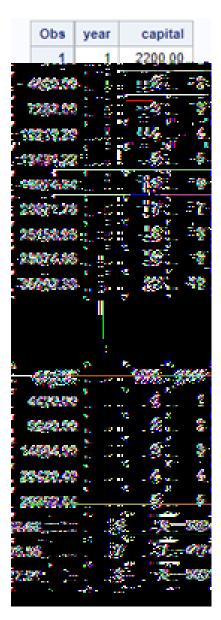
- If we have a lot of datasets and there are some primary formatting which has to been done in all of these datasets, we can use this method of storing the user define format and can use for all of the datasets in very simple 2 steps and don't have to define again and again.
- For this we have to call out our library to store the data and have to make a library for storing it.
- Then we can define the values and grades as we did previously.
- Then we printed the stored data in which we can check what exactly we defined.
- Later, we imported a data and find out total and mean of the dataset.
- Later, we called out the data where we stored our user defined format and refered to it.
- Then we printed our data and just used the format statement so that sas environment can understand exactly in which dataset we want to use the use define format and for exactly which variables we want to use it and got our output.

USING CONDITIONAL CLAUSE WHILE IT-TERATING DO-WHILE STATEMENT

```
CODE:
/* Using conditional clause while itterating do-while statement */
/* In this dataset suppose we want to limit the number of years to invest in the capital to
and add the UNTIL expression to determine years it take for investment to reach 50 thousand
number of years */
data invest;
do year = 1 to 10 until (capital >= 50000);
capital + 2000;
capital + capital * 0.10; /* if the interest rate is 10% */
output;
end;
/* if year = 11 then year = 10; */
run;
data invest;
do year = 1 to 10 until (capital \geq 50000);
capital + 4000;
capital + capital * 0.10; /* if the interest rate is 10% */
output;
end;
if year = 11 then year = 10;
run;
```

Сору

OUTPUT:



- Here we tried to find out at 10% interest rate after investing some exact amount each year + gained amount how long will it take to reach that capital amount to 50000 in maximum of 10 years time.
- $\bullet\,$ Here at first out investment amount was 2000 each year and we can see

that we couldn't make 50000 at the end of 10th year and it seems like we need more time.

• So, later we tried to invest 4000 each year and got the result with the help of do-while loop that we can gain 50000 capital plus some amount at the end of 8th year and don't have to wait till 10th year.

CREATING VARIABLES IN ARRAY STATE-MENT

- array array_name {dimension}
- if variables are not specified then new variables name will be created like array_name1, array_name2 ... till dimension specification
- if we want to specify the character array then syntax will be:

array array_name {5}\$;

- if we don't specify the length of the character it becomes 8
- if we want to increase the length of the assigned variable then syntax will be:

array array_name {5}\$ 24;

Assigning initial value in array & creating temporary array element Syntax:

```
array Goal{3} G1 G2 G3 (50 32 41)
```

• if array is string,

then, array string{3} \$ str1-str3 ('red', 'green', 'blue')

• temporary array element which is not written in output,

```
array var{4}temporary(40 41 42 43) */
```

```
CODE:
```

```
proc import datafile= "/home/u59242738/Data Files/score_data_miss999.xlsx"
dbms = xlsx out = data0 replace;
run;
data score_data1(drop= i);
set data0;
array score_variable(3) score1 score2 score3;
```

```
do i = 1 to 3;
if score_variable(i) = 999 then score_variable(i) = .;
end;
average_score = mean(score1, score2, score3);
run;
proc print data= score_data1;
run;
data score_data2(drop= i);
set score_data1;
array score(3) score1 score2 score3;
array score_diff(3);
do i = 1 to 3;
score_diff{i} = score{i} - average_score;
end;
run;
proc print data= score_data2;
title "Creating variables in an Array statement";
run;
proc means data= score_data1;
var score1 score2 score3;
run;
data score_data3(drop= i);
set score_data1;
```

```
array score(3) score1 score2 score3;
array avg{3} (79.5 81.9 80.8); /*Assigning initital values*/
array score_diff(3);
do i = 1 to 3;
score_diff{i} = score{i} - AVG{i};
end;
run;
proc print data= score_data3 (keep= name score_diff1 score_diff2 score_diff3);
title "Finding our X - Xbar using arrays";
run;
Copy
OUTPUT:
```

Obs	name	score1	score2	score3	gender	average_score
- 1	Tim	88	80	99	m	89.0000
2	Sharon	95	90	90	f	91.6667
3	Sean	45	78	56	m	59.6667
4	Mike	89		77	m	83.0000
5	Tina	54	69	69	f	64.0000
6	Sara	67	78	74	f	73.0000
7	Spencer	98	82	78	m	86.0000
8	James		96	89	m	92.5000
9	Jaden	98	78	89	m	88.3333
10	Will	92	89	98	m	93.0000
11	Jack	69	79	70	m	72.6667
12	Mila					

Creating variables in an Array statement

Obs	name	score1	score2	score3	gender	average_score	score_diff1	score_diff2	score_diff3
1	Tim	88	80	99	m	89.0000	-1.0000	-9.0000	10.0000
2	Sharon	95	90	90	f	91.6667	3.3333	-1.6667	-1.6667
			_				The second secon		



Creating variables in an Array statement

The MEANS Procedure

Variable	Label	N	Mean	Std Dev	Minimum	Maximum
score1	score1	10	79.5000000	19.2945012	45.0000000	98.0000000
score2	score2	10	81.9000000	7.7380733	69.0000000	96.0000000
score3	score3	11	80.8181818	13.3627705	56.0000000	99.0000000

Finding our X - Xbar using arrays

Obs	name	score_diff1	score_diff2	score_diff3
1	Tim	8.5	-1.9	18.2
2	Sharon	15.5	8.1	9.2
3	Sean	-34.5	-3.9	-24.8
4	Mike	9.5		-3.8
5	Tina	-25.5	-12.9	-11.8
6	Sara	-12.5	-3.9	-6.8
7	Spencer	18.5	0.1	-2.8
8	James		14.1	8.2
9	Jaden	18.5	-3.9	8.2
10	Will	12.5	7.1	17.2
11	Jack	-10.5	-2.9	-10.8
12	Mila			

- At first, we imported the datafile and converted all the 999 values to missing values (.) and also found out the average score of all the students.
- Then we again used array statement and found out the difference between each score and the average (per person wise) score for those individual students.
- Then we found out the subject wise number of available values of students, average score, std deviation, minimum and maximum values of all the 3 subjects available.
- Then we again used array statement and found out the difference between each score and the average (per subject wise) score for those students.

• Also, while printing this last output we ignored to print the rest of the columns like score1, score2, score3, gender, and person wise average score.

CREATING TEMPORARY ARRAY ELE-MENTS WHICH WE DONOT NEED TO PRINT IN THE FINAL RESULT

```
/* Creating temporary array elements which need not be printed in the final output*/
proc import datafile="/home/u59242738/Data Files/score_data_miss999.xlsx"
out=data dbms=xlsx replace;
run;
data data1 (drop = i);
set data;
array score_array(3) score1 score2 score3;
do i=1 to 3;
if score_array(i)=999 then score_array(i)= . ;
end;
average_score = mean(score1,score2, score3);
run;
data data2 (drop = i);
set data1;
array score(3) score1 score2 score3;
array average {3}_temporary_(79.5 81.9 80.8); /*We are assigning values to average 1 2 and 3
temporarily and does not require in final output*/
```

```
/* array average{3}_temporary_(79.5 81.9 80.8); */
array score_diff(3);
do i = 1 to 3;
score_diff{i} = score{i} - average{i};
end;
run;
proc print data=data2;
title "Assingning temporary array elements";
run;
Copy
OUTPUT:
```

Obs	name	score1	score2	score3	gender	average_score	score_diff1	score_diff2	score_diff3
1	Tim	88	80	99	m	89.0000	8.5	-1.9	18.2
2	Sharon	95	90	90	f	91.6667	15.5	8.1	9.2
3	Sean	45	78	56	m	59.6667	-34.5	-3.9	-24.8
4	Mike	89		77	m	83.0000	9.5		-3.8
5	Tina	54	69	69	f	64.0000	-25.5	-12.9	-11.8
6	Sara	67	78	74	f	73.0000	-12.5	-3.9	-6.8
7	Spencer	98	82	78	m	86.0000	18.5	0.1	-2.8
8	James		96	89	m	92.5000		14.1	8.2
9	Jaden	98	78	89	m	88.3333	18.5	-3.9	8.2
10	Will	92	89	98	m	93.0000	12.5	7.1	17.2
11	Jack	69	79	70	m	72.6667	-10.5	-2.9	-10.8
12	Mila								

- $\bullet\,$ At first we imported the dataset and found out the person wise average score in the 3 scores given.
- Then we find out the subject wise average score but didn't store it permanently in the sas dataset but rather found out the difference in subject wise average score and the scores obtained by the students with the help of array statement.

MERGING TWO DATASETS (ONE TO ONE)

```
/* Combining SAS dataset (one-one mail merging) */
/* one-one merging */
proc import datafile= "/home/u59242738/Data Files/score_data_id_partial.xlsx"
out= data_a dbms= xlsx replace;
run;
proc import datafile= "/home/u59242738/Data Files/score_data_id.xlsx"
out= data_b dbms= xlsx replace;
run;
data one2one;
set data_a;
set data_b;
run;
proc print data= one2one;
title "One to One marging";
run;
proc print data= data_a;
title "Partial dataset";
proc print data= data_b;
title "Complete dataset";
run;
Сору
OUTPUT:
```

One to One marging

Obs	name	score1	score2	gender	stu_id	score3
1	Tim	88	80	m	1	99
2	Sharon	95	90	f	2	90
3	Sean	45	78	m	3	56
4	Mike	89		m	4	77
5	Tina	54	69	f	5	69
6	Sara	67	78	f	6	74

Partial dataset

Obs	name	score1	score2	gender	stu_id
1	Tim	88	80	m	1
2	Sharon	95	90	f	2
3	Sean	45	78	m	3
4	Mike	89		m	4
5	Tina	54	69	f	5
6	Sara	67	78	f	6

Complete dataset

Obs	name	score1	score2	score3	gender	stu_id
1	Tim	88	80	99	m	1
2	Sharon	95	90	90	f	2
3	Sean	45	78	56	m	3
4	Mike	89	_	77	m	4
5	Tina	54	69	69	f	5
6	Sara	67	78	74	f	6
7	Sam	98	82	78	m	7
8	James		96	89	m	8
9	Jaden	98	78	89	m	9
10	Jack	69	79	70	m	11
11	Will	92	89	98	m	10

EXPLANATION:

- There are 2 datasets given.
- i) Full complete dataset given.
- ii) Partial dataset given, like details of only 6 students are given out of 11 and score3 variables scores are not present in the dataset.
 - At first, we imported both of these 2 datasets.
 - Then just made a new dataset and set both of the datasets in there and simply we got the one-to-one merged dataset which was required here.
 - For convenience, the 2 given datasets are also printed here.

CONCATENATING TWO DATASETS

• general form of CONCATENATING (syntax):

data <file name>;
set <file name1> <file name 2>;
run;

```
CODE:
/* Concatenating (combining)*/
proc import datafile= "/home/u59242738/Data Files/score_data_id_partial.xlsx"
out= data_a dbms= xlsx replace;
run;
proc import datafile= "/home/u59242738/Data Files/score_data_id.xlsx"
out= data_b dbms= xlsx replace;
run;
data concat;
set data_a data_b;
run;
proc print data= concat;
title "Concatenated dataset";
run;
proc print data= data_a;
title "partial dataset";
proc print data= data_b;
title "complete dataset";
run;
Сору
OUTPUT:
```



partial dataset

Obs	name	score1	score2	gender	stu_id
1	Tim	88	80	m	1
2	Sharon	95	90	f	2
3	Sean	45	78	m	3
4	Mike	89	_	m	4
5	Tina	54	69	f	5
6	Sara	67	78	f	6

complete dataset

Obs	name	score1	score2	score3	gender	stu_id
1	Tim	88	80	99	m	1
2	Sharon	95	90	90	f	2
3	Sean	45	78	56	m	3
4	Mike	89		77	m	4
5	Tina	54	69	69	f	5
6	Sara	67	78	74	f	6
7	Sam	98	82	78	m	7
8	James		96	89	m	8
9	Jaden	98	78	89	m	9
10	Jack	69	79	70	m	11
11	Will	92	89	98	m	10

- There are 2 datasets given.
- i) Full complete dataset given.
- ii) Partial dataset given, like details of only 6 students are given out of 11 and score3 variables scores are not present in the dataset.
 - At first, we imported both of these 2 datasets.
 - Then just made a new dataset and set both of the datasets in there in the same line instead of taking 2 datasets in 2 different sets in 2 different lines and simply we got the concatenated (combined) dataset which was required here.
 - For convenience, the 2 given datasets are also printed here.

APPENDING TWO DATASETS

• syntax:

```
proc append base = <filename_1>
data = <filename_2>;
run;
```

- Base is the dataset to which observations are added.
- Data is the name of the dataset containing observation that have to be added in base dataset.
- We use force command where there is unlike structure of datasets.

```
/* Appending */
proc import datafile= "/home/u59242738/Data Files/score_data_id_partial.xlsx"
out= data_a dbms= xlsx replace;
run;
proc import datafile= "/home/u59242738/Data Files/score_data_id.xlsx"
out= data_b dbms= xlsx replace;
run;
proc append base= data_a
data= data b force;
```

```
run;
proc print data= data_a;
title "Appending dataset";
run;
Copy
OUTPUT:
```

Appending dataset

Obs	name	score1	score2	gender	stu_id
1	Tim	88	8.0	m	1
2	Sharon	95	90	f	2
3	Sean	45	78	m	3
4	Mike	89	-	m	4
5	Tina	54	69	f	5
6	Sara	67	78	f	6
7	Tim	88	80	m	1
8	Sharon	95	90	f	2
9	Sean	45	78	m	3
10	Mike	89	-	m	4
11	Tina	54	69	f	5
12	Sara	67	78	f	6
13	Sam	98	82	m	7
14	James		96	m	8
15	Jaden	98	78	m	9
16	Jack	69	79	m	11
17	Will	92	89	m	10

EXPLANATION:

• There are 2 datasets given.

- i) Full complete dataset given.
- ii) Partial dataset given, like details of only 6 students are given out of 11 and score3 variables scores are not present in the dataset.
 - At first, we imported both of these 2 datasets.
 - Here the appending was not so simple as merging and concatenating.
 - Here, we have to use append command in proc statement taking first data as the base data and second data to add it after the base data and then printed it and got the appending data which was required here.
 - As there is unlike structure of datasets, that's why we had to use the force command or else there would be the observations from the second dataset used in the command here as there is an extra variable here then the base dataset.