

Homework #8- Processing data iteratively; Restructuring a data set

Directions: Please submit one program file, one output file, and one log file for the entire assignment. Use comment statements to separate your answers. For questions that do not require a SAS program use comment statements. For example:

```
/*
Question #1d: my answer
Question #2a: my answer
*/
/*Question #4b: */
--SAS program—
/*Question #5*/
```

Please make sure the log and output file contain only one run. For example, clear the screen for the log and output file and submit your program one last time before you upload your solutions to **Blackboard**. See lab 1 for the instructions on how to clear your output and log files.

Part I:

1. Using an Iterative DO Statement with a Conditional Clause

Orion's income last year was \$50,000,000 and expenses totaled \$38,750,000. Income is projected to increase at 1% per year and expenses are expected to increase at 2% per year.

- Create a SAS data set named **work.expenses** that contains each year's projected income and expenses.
 - Use an iterative DO statement with a conditional clause.
 - Stop the loop when expenses exceed income or after 30 years, whichever comes first.
- Print the results and format **Income** and **Expenses** with a dollar sign and two decimal places.

Hint: Recall that an iterative DO statement with a conditional clause produces different results with DO WHILE and DO UNTIL statements. The results below were generated using a DO UNTIL statement.

PROC PRINT Output

Obs	Income	Expenses	Year
1	\$64,762,815.75	\$64,844,951.93	26

2. Using Arrays for Repetitive Computations

Monthly customer order data for the first half of the year is stored in the data set **orion.orders_midyear**. Orion Star Sales management is considering a 10% price decrease during the first three months of the upcoming year. Management wants to see how such a discount would affect this year's sales.

Partial **orion.orders_midyear**

Customer_ID	Month1	Month2	Month3	Month4	Month5	Month6
5	213.10	.	478.0	525.80	394.35	191.79
10	188.10	414.09	2876.9	3164.59	2373.44	169.29
11	78.20	70.38
12	135.60	.	117.6	129.36	97.02	122.04
18	.	.	29.4	32.34	24.26	.

- Create a data set, **special_offer**, including the 10% discount in months 1 through 3.
 - Create an array, **Mon**, to access **Month1** through **Month3**.
 - Use a DO loop to adjust each customer's monthly data to include the 10% discount (**Month1** through **Month3** only).
 - Create three new variables:
 - Total_Sales** – the total of current sales over the six months
 - Projected_Sales** – the total of the adjusted sales over the six months (including the three discounted months)
 - Difference** – the difference between **Total_Sales** and **Projected_Sales**
 - Keep only the new variables: **Total_Sales**, **Projected_Sales**, and **Difference**.
- Print the resulting data set and verify your results.
 - Suppress the session start date, page number, and observation column.
 - Add an appropriate title.
 - Use the SUM statement to display a total for the **Difference** variable.
 - Use the DOLLAR. format for all variables.

Partial PROC PRINT Output (24 Total Observations)

Total Sales with 10% Discount in First Three Months			
	Total_ Sales	Projected_ Sales	Difference
	\$1,803.04	\$1,733.93	\$69.11
	\$9,186.41	\$8,838.50	\$347.91
	\$148.58	\$140.76	\$7.82
	\$601.62	\$576.30	\$25.32
...			
	\$1,518.95	\$1,467.02	\$51.93
			=====
			\$1,550.74

3. Using a Character Array for Table Lookup

The Public Safety Department at Orion Star wants all employees to be aware of the new policies and procedures regarding customer incidents in their retail stores.

- Each employee must participate in web-based training and then take a multiple-choice test that consists of 10 questions.
- Each question has five choices (A through E).
- The test results from each testing session are entered into the SAS data set **orion.test_answers** as shown below.
- Each observation in **orion.test_answers** contains a single person's answers.

Partial **orion.test_answers** (15 Total Observations)

Employee_ ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
121044	A	C	C	B	D	E	D	B	B	A
120145	B	C	C		E	E	D	B	A	A
120761	A	C	C	B	D	D	E	B	B	C
120656	B	C	C	A	D	B	B	C	A	D
121107	A	C	C	B	E	E	D	B	B	A

- The correct answers for the questions are shown below:

Question:	1	2	3	4	5	6	7	8	9	10
Answer:	A	C	C	B	E	E	D	B	B	A

- a. Read **orion.test_answers** and determine whether each person passed or failed the test.

- Compute a variable **Score** that contains the total correct answers for each person.



Create a temporary array for the answer key.

- If an employee scores 7 or higher, write the observation to a data set named **passed**.
- If an employee scores less than 7, write the observation to a data set named **failed**.

- b. Print the **passed** data set to verify that it contains 12 observations.

Partial PROC PRINT Output (12 Total Observations)

Passed												
Obs	Employee_ ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
1	121044	A	C	C	B	D	E	D	B	B	A	9
2	120145	B	C	C		E	E	D	B	A	A	7
3	121107	A	C	C	B	E	E	D	B	B	A	10
4	121038	B	C	C	B	D	D	D	B	B	A	7
5	120273	C	C	C	B	E	E	E	B	B	A	8

- c. Print the **failed** data set to verify that it contains three observations.

PROC PRINT Output

Failed												
Obs	Employee_ ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
1	120761	A	C	C	B	D	D	E	B	B	C	6
2	120656	B	C	C	A	D	B	B	C	A	D	2
3	120798		A	C	B	D	D	D	B	B	A	6

Part II-Restructing a data set

1. Rotating a Data Set and Using a Lookup Table

The data set **orion.travel_expense** contains an observation for each employee business trip and includes **Trip_ID**, **Employee_ID**, and up to five expenses, **Exp1-Exp5**. The table below shows the type of each expense:

Exp1	Airfare
Exp2	Hotel
Exp3	Meals
Exp4	Transportation
Exp5	Miscellaneous

Partial **orion.travel_expense** (10 Total Observations)

Obs	Trip_ID	Employee_ ID	Exp1	Exp2	Exp3	Exp4	Exp5
1	1044-1	121044	345.97	568.54	235.00	320.00	.
2	0145-1	120145	256.00	675.90	343.25	125.00	67.50
3	0656-1	120656	312.26	.	236.98	325.00	45.00
4	1119-1	121119	597.80	780.99	345.87	195.00	50.75
5	0812-1	120812	345.24	865.45	534.20	430.50	76.75

- Rotate **orion.travel_expense** to create an output data set named **travel** that contains one observation per nonmissing travel expense.
 - The new data set should contain the variables **Trip_ID**, **Employee_ID**, **Expense_Type**, and **Amount**.
 - Use two arrays in your solution: one to refer to each of the five expenses in an observation and a second array to store the expense types as a lookup table.
- Print the new data set.
 - Format **Amount** with dollar signs, commas, and two decimal places.
 - Compare your results to the partial listing below.

Partial PROC PRINT Output (44 Total Observations)

Obs	Trip_ID	Employee_ ID	Expense_Type	Amount
1	1044-1	121044	Airfare	\$345.97
2	1044-1	121044	Hotel	\$568.54
3	1044-1	121044	Meals	\$235.00
4	1044-1	121044	Transportation	\$320.00
5	0145-1	120145	Airfare	\$256.00

Part III- Supplemental exercises for STAT 625 and Honors credit

1. Using Other Loop Control Statements

Orion's income last year was \$50,000,000 and expenses totaled \$38,750,000. Income is projected to increase at 1% per year and expenses are expected to increase at 2% per year.

- a. Use an iterative DO statement to calculate projected income and expenses for the next 75 years.
 - Investigate SAS documentation for information about other loop control statements such as CONTINUE and LEAVE.
 - Include the appropriate loop control statement (CONTINUE/LEAVE) to stop the loop when expenses exceed income.
- b. Print the results and format the values of **Income** and **Expenses** with dollar signs and two decimal places.

Hint: Recall that the DO WHILE and DO UNTIL statements might produce different results.

PROC PRINT Output

Obs	Income	Expenses	Year
1	\$64,762,815.75	\$64,844,951.93	26

2. Terminating a DATA Step

Monthly order data for the first half of the year is stored in the **orion.orders_midyear** data set. This data set is updated monthly and can contain data for 1 to 12 months.

Partial **orion.orders_midyear**

Obs	Customer_ID	Month1	Month2	Month3	Month4	Month5	Month6
1	5	213.10	.	478.0	525.80	394.35	191.79
2	10	188.10	414.09	2876.9	3164.59	2373.44	169.29
3	11	78.20	70.38
4	12	135.60	.	117.6	129.36	97.02	122.04
5	18	.	.	29.4	32.34	24.26	.

- a. Orion Star decided to create a Frequent Shopper Program (FSP) if the customer has order data for at least three months. To be eligible for the invitation to the program, the customer has to have placed order in 50% of the months to date and has to have spent at least \$1,000 since the beginning of the year. Example: If data file has 6 months of orders from a customer, the customer must have spent over \$1,000 in three or more months. If the data file has 3 months of orders, the customer has to have spent over \$1,000 in two or more months to be eligible for the Frequent Shopper program.
 - Open the input data set and verify that it contains at least three months of data.
 - If there is less than three months, write a message to the SAS log and stop the DATA step immediately.
 - If there are at least three months of data in the data set, create an array, **Mon**, to access the **Monthn** variables, regardless of how many **Monthn** variables exist in the data set.
Hint: Consider using a SAS variable list to list the array elements.
 - Use a DO loop to examine each customer's data to determine whether the customer qualifies for the Frequent Shopper Program.
 - Create a new data set, **fsp**, that contains an observation for each qualifying customer.
 - Use the report below to determine which variables to drop or keep.

Hint: Use SAS documentation to investigate the use of the STOP statement and the DIM function.

- b. Print the resulting data set with an appropriate title and formats, and verify your results.

Partial PROC PRINT Output (11 Total Observations)

orion.orders_midyear: Frequent Shoppers			
Obs	Customer_ID	Total_ Order_ Amount	Months_ Ordered
1	5	\$1,803.04	5
2	10	\$9,186.41	6
3	27	\$1,373.23	3
4	31	\$3,113.47	3
5	34	\$1,473.18	5

- c. Test your program using **orion.orders_qtr1**.

PROC PRINT Output

orion.orders_qtr1: Frequent Shoppers			
Obs	Customer_ID	Total_ Order_ Amount	Months_ Ordered
1	10	\$3,479.09	3
2	27	\$1,093.60	2
3	31	\$1,777.60	2
4	171	\$1,849.99	3
5	2806	\$1,506.90	3

- d. Test your program using **orion.orders_two_months**.

Partial SAS Log

Insufficient data for Frequent Shopper Program	
NOTE: There were 1 observations read from the data set ORION.ORDERS_TWO_MONTHS.	
NOTE: The data set WORK.FSP has 0 observations and 3 variables.	
NOTE: DATA statement used (Total process time):	
1066	
1067 title 'orion.orders_two_months: Frequent Shoppers ';	
1068 proc print data=fsp;	
1069 format total_order_amount dollar10.2;	
1070 run;	
NOTE: No observations in data set WORK.FSP.	

3. Rotating a Data Set

The data set **orion.order_summary** contains monthly order information with multiple observations for each customer.

Partial **orion.order_summary** (101 Total Observations)

Customer_ID	Month	Order_ Sale_Amt
5	5	478.00
5	6	126.80
5	9	52.50
5	12	33.80
10	3	32.60

- a. Rotate **orion.order_summary** to create an output data set named **customer_orders** that contains one observation for each customer.

- Use an array in a DATA step.
- The new data set should contain **Customer_ID** and **Month1** through **Month12**.

- b. Print the new data set. Verify the results.

Partial PROC PRINT Output (37 Total Observations)

Customer_ID	Month1	Month2	Month3	Month4	Month5	Month6	Month7	...	Month 11	Month12
-------------	--------	--------	--------	--------	--------	--------	--------	-----	----------	---------

5	478.0	126.80	.	.	33.80
10	.	.	32.6	250.8	79.8	12.20	163.29	1894.60	143.30
11
12	.	117.6	.	.	.	48.40	.	.	.
18	.	29.4