Homework, Processing Data Iteratively

**Exercise 1**. Performing Computations with DO Loops

The Payroll department wants to project total employee costs (wages, retirement benefits, and medical benefits) through future years based on assumed increases:

* wages will increase by 3% each year
* retirement benefit costs will increase by 1.4% each year
* medical benefit costs will increase by 9.5% each year.

The present total costs for these items is:

* Wages total $12,874,000
* Retirement benefit costs total is $1,765,000
* Medical benefit costs total is $649,000

1. Use a DATA step to create a file: work.future\_costs.

(a) initialize the variables wages, retire, medicine with the present values of wages, retirement benefits, and medical benefits cost.

(b) Use a do loop to calculate the total costs for each of the next 10 years. For each year include a variable, total cost that is the sum of the three variables, wages, retire, and medicine.

2. Use a PROC PRINT step to display the data portion of work.future\_costs. (There should be 10 observations in the data set.)

3. The present corporate income is $50,000,000 and is expected to grow at a rate of 1% per year. Write another data step that modifies the one above and examines how long it is before total\_costs exceeds income.

**Exercise 2.** Using an Iterative DO Statement with a Conditional Clause

A company’s income last year was $50,000,000 and expenses totaled $38,750,000. Income is projected to increase at one percent per year and expenses are expected to increase at two percent per year.

1. Use a data step to create a SAS data set named work.expenses that contains each years 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.
2. Use a PROC PRINT step to display the results and format Income and Expenses with a dollar sign and two decimal places.

**Exercise 3**. 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 one percent per year and expenses are expected to increase at two percent per year.

1. Use a DATA step to create a data set work.income that projects income and expenses for the next 75 years.

(a) Use an iterative DO statement to calculate projected income and expenses for the next 75 years.

(b) Investigate SAS documentation for information on the loop control statements CONTINUE and LEAVE. (<http://support.sas.com/documentation/94/index.html>)

(c) Include the appropriate loop control statement (CONTINUE/LEAVE) to stop the loop when expenses exceed income.

2. Use a PROC PRINT step to display and print the results and format the values of Income and Expenses with dollar signs and two decimal places.

**Exercise 4**. Using Arrays for Repetitive Computations

The SAS data set orders\_midyear is in the STA5066 subdirectory. The file contains monthly customer order data for the first half of the year. The Sales manager is considering a five-percent price decrease next year. She wants to see how such a discount would affect this year’s income by applying it to the midyear data.

1. Use a PROC CONTENTS step to examine the descriptor portion of the data set.

2. Use a DATA step to read orders\_midyears and create a new data set, discount\_sales, that reflects the five-percent discount.

(a) Create an array, Mon, to access Month1 through Month6 using a SAS variable list construction.

(b) Use a DO loop to adjust each customer’s monthly data by applying the five-percent discount.

3. Use a PROC PRINT step to display the resulting data set and verify your results. Use the DOLLAR. format for the monthly sales amounts.

**Exercise 5**. Using Arrays for Repetitive Computations 2

Monthly customer order data for the first half of the year is stored in the data set orders\_midyear in the STA5066 sub directory. Sales management is considering a 10-percent price decrease during the first three months of the upcoming year. They want to see how this discount would affect this year’s sales.

1. Use a DATA step to read orders\_midyear and create a data set, special\_offer, including the 10-percent discount in months 1 through 3.

(a) Create an array, Mon, to access Month1 through Month3.

(b) Use a DO loop to calculate each customers monthly data to include the 10-percent discount (Month1 through Month3 only).

(c) Create three new variables:

i. Total\_Sales: the total of current sales over the six months

ii. Projected\_Sales: the total of the adjusted sales over the six months (including the three discounted months)

iii. Difference: the difference between Total\_Sales and Projected\_Sales

iv. Keep only the new variables: Total\_Sales, Projected\_Sales, and Difference.

2. Use a PROC PRINT step to print the resulting data set and verify your results.

(a) Display a total for the Difference variable. (Check SAS documentation of PROC PRINT for the SUM statement.)

(b) Use the DOLLAR. format for all variables.

**Exercise 6.** Using an Array for Table Lookup

The SAS data set orders\_midyear is in the STA5066 subdirectory. The manager of the Sales department wants to identify preferred customers for an upcoming promotion. Use the orders\_midyear data set and a lookup table to create a new data set, preferred\_cust.

1. Use an array to refer to the values in the variables month1 to month6.

2. Create a temporary array, Target, to use as a lookup table. The array Target, should hold the target sales amount for each month: 200, 400, 300, 100, 100, and 200

3. Create new variables, Over1, Over2, . . . ,Over6, to hold the amount that a customer spent above the corresponding months target.

4. Use a DO loop to calculate the values of Over1 through Over6 when the corresponding month’s sales amount exceeds the target.

5. Store the sum of Over1 through Over6 in another new variable, Total\_Over.

6. Write an observation only if Total Over is greater than 500.

7. . The new data set, preferred cust, should include only the variables Customer ID, Over1 through Over6, and Total\_Over.

8. Use a PROC PRINT step to print preferred\_cust and verify your results.

**Exercise 7**. Using a Character Array for Table Lookup

The SAS data set test\_answers is in the STA5066 sub directory.

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

Each employee participated in Web-based training and then took a multiple-choice test that consists of 10 questions. Each question has five choices (A-E).

The test results from each testing session are entered into the SAS data set test\_answers. Each observation in test answers contains a single persons answers.

1. Use a PROC PRINT step to examine the contents of test\_answers.

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

2. Use the data set test\_answers to determine whether each person passed or failed the test and to create two new SAS data sets, passed, and failed.

(a) Compute a variable Score that contains the total number of correct answers for each person.

(b) Use a temporary array for the answer key. (You will need $ after array name)

(c) If an employee scores 7 or higher, write the observation to a data set named passed.

(d) If an employee scores less than 7, write the observation to a data set named failed.

3. Use PROC PRINT steps to verify that passed contains 12 observations and failed contains three observations.

**Exercise 8**. The SAS dataset labsubset is in the STA5066 sub directory

This file contains the following variables:

Variable Name Description

seqn sequence number

hgp hemoglobin (g/dl)

htp hematocrit (%)

tcp cholesterol (mg/dl)

tgp triglycerides (mg/dl)

lcp low density lipoprotein (mg/dl)

hdp high density lipoprotein (mg/dl)

fbpsi fibrinogen (mg/dl)

crp C reactive protein (mg/dl)

sgp plasma glucose (mg/dl)

urp urinary creatinine (mg/dl)

1. Use a PROC MEANS step to determine which variables on the dataset have fill values (whole number consisting of 8s, such as 8, 888, etc.)

2. Use a DATA step that creates a file work.examsub2 and recodes all values that correspond to fill values to SAS unknown values (.).

(a) Use an array to represent the variables that have 8 ”fills”.

(b) Use a temporary array, unknown, to hold the values that correspond to fill values for each variable.

(c) Use a do loop to replace fill values with unknowns.

3. Use a PROC MEANS step to check that your program has run correctly.