

Unit 2-2 Exercises

1. Outputting Multiple Observations

The **orion.prices** data set contains price information for Orion Star products.

Partial Listing of **orion.prices** (50 Total Observations)

Product_ID	Unit_Price	Factor
210200100009	\$34.70	1.01
210200100017	\$40.00	1.01
210200200023	\$19.80	1.01
210200600067	\$67.00	1.01
210200600085	\$39.40	1.01
210200600112	\$21.80	1.01
210200900033	\$14.20	1.01
210200900038	\$20.30	1.01
210201000050	\$19.60	1.01
210201000126	\$6.50	1.01

- a. Write a DATA step to create a new data set that forecasts unit prices for the next three years. This data set will contain three observations for each input observation read from **orion.prices**.
 - Open file **p202e01**. It reads **orion.prices** and creates a new data set named **work.price_increase**.
 - Use explicit OUTPUT statements to forecast unit prices for the next three years, using **Factor** as the annual rate of increase.
- b. Print the new data set.
 - Include only **Product_ID**, **Unit_Price** and **Year** in the report.
 - Verify your results.

Partial PROC PRINT Output (150 Total Observations)

Obs	Product_ID	Unit_Price	Year
1	210200100009	\$35.05	1
2	210200100009	\$35.40	2
3	210200100009	\$35.75	3
4	210200100017	\$40.40	1
5	210200100017	\$40.80	2
6	210200100017	\$41.21	3
7	210200200023	\$20.00	1
8	210200200023	\$20.20	2
9	210200200023	\$20.40	3
10	210200600067	\$67.67	1

2. Outputting Multiple Observations

The data set **orion.discount** contains information about various discounts that Orion Star runs on its products.

Partial Listing of **orion.discount**

Product_ID	Start_ Date	End_Date	Unit_Sales_ Price	Discount
210100100027	01MAY2007	31MAY2007	\$17.99	70%
210100100030	01AUG2007	31AUG2007	\$32.99	70%
210100100033	01AUG2007	31AUG2007	\$161.99	70%
210100100034	01AUG2007	31AUG2007	\$187.99	70%
210100100035	01MAY2007	31MAY2007	\$172.99	70%
210100100038	01JUL2007	31JUL2007	\$59.99	60%
210100100039	01JUN2007	31AUG2007	\$21.99	70%
210100100048	01AUG2007	31AUG2007	\$13.99	70%

- a. Due to excellent sales, all discounts from December 2007 will be repeated in July 2008. Both the December 2007 and the July 2008 discounts will be called the Happy Holidays promotion.
 - Create a new data set named **work.extended** that contains all discounts for the Happy Holidays promotion.
 - Use a WHERE statement to read only observations with a start date of 01Dec2007.
 - Create a new variable, **Promotion**, which has the value Happy Holidays for each observation.
 - Create another new variable, **Season**, that has a value of Winter for the December observations and Summer for the July observations.
 - July 2008 discounts should have a start date of 01Jul2008 and an end date of 31Jul2008.
 - Drop the **Unit_Sales_Price** variable.
 - Use explicit OUTPUT to write two observations for each observation read.
- b. Print the new data set.
 - Add an appropriate title
 - Verify the results.

Partial PROC PRINT (332 Total Observations)

All discount ranges with the Happy Holidays promotion						
Obs	Product_ID	Start_ Date	End_Date	Discount	Promotion	Season
1	210200100007	01DEC2007	31DEC2007	50%	Happy Holidays	Winter
2	210200100007	01JUL2008	31JUL2008	50%	Happy Holidays	Summer
3	210200300013	01DEC2007	31DEC2007	50%	Happy Holidays	Winter
4	210200300013	01JUL2008	31JUL2008	50%	Happy Holidays	Summer
5	210200300025	01DEC2007	31DEC2007	50%	Happy Holidays	Winter

3. Using Conditional Logic to Output Multiple Observations

The data set **orion.country** contains information on country names as well as various lookup codes.

Listing of **orion.country**

Country	Country_Name	Population	Country_ ID	Continent_ ID	Country_Former Name
AU	Australia	20,000,000	160	96	
CA	Canada	.	260	91	
DE	Germany	80,000,000	394	93	East/West Germany
IL	Israel	5,000,000	475	95	
TR	Turkey	70,000,000	905	95	
US	United States	280,000,000	926	91	
ZA	South Africa	43,000,000	801	94	

- Create a new data set that contains one observation for each current country name as well as one observation for each former country name.
 - Use conditional logic and explicit OUTPUT statements to create a data set named **work.lookup**.
 - If a country has a former country name, write two observations: one with the current name in the **Country_Name** variable and another with the former country name in the **Country_Name** variable.
 - Drop the variables **Country_FormerName** and **Population**.
 - Create a new variable named **Outdated** with values of either Y or N to indicate whether the observation represents the current country name.
- Print the new data set with an appropriate title.

PROC PRINT Output

Current and Outdated Country Name Data						
Obs	Country	Country_Name	Country_ ID	Continent_ ID	Outdated	
1	AU	Australia	160	96	N	
2	CA	Canada	260	91	N	
3	DE	Germany	394	93	N	
4	DE	East/West Germany	394	93	Y	
5	IL	Israel	475	95	N	
6	TR	Turkey	905	95	N	
7	US	United States	926	91	N	
8	ZA	South Africa	801	94	N	

4. Creating Multiple SAS Data Sets

The data set **orion.employee_organization** contains information about employee job titles, departments, and managers.

Partial Listing of **orion.employee_organization** (424 Total Observations)

Employee_ ID	Job_Title	Department	Manager_ ID
120101	Director	Sales Management	120261
120102	Sales Manager	Sales Management	120101
120103	Sales Manager	Sales Management	120101
120104	Administration Manager	Administration	120101
120105	Secretary I	Administration	120101
120106	Office Assistant II	Administration	120104
120107	Office Assistant III	Administration	120104

- a. Create a separate data set for each department.

Name the data sets **work.admin**, **work.stock**, and **work.purchasing**.

Use conditional logic and explicit OUTPUT statements to write to these data sets depending on whether the value of **Department** is Administration, Stock & Shipping, or Purchasing, respectively. Ignore all other **Department** values.

Hint: Be careful with capitalization and the spelling of the **Department** values.

- b. Print **work.admin** and verify your results.

Add an appropriate title.

Partial Listing of **work.admin** (34 Total Observations)

Administration Employees					
Obs	Employee_ ID	Job_Title	Department	Manager_ ID	
1	120104	Administration Manager	Administration	120101	
2	120105	Secretary I	Administration	120101	
3	120106	Office Assistant II	Administration	120104	
4	120107	Office Assistant III	Administration	120104	
5	120108	Warehouse Assistant II	Administration	120104	
6	120109	Warehouse Assistant I	Administration	120104	
7	120110	Warehouse Assistant III	Administration	120104	
8	120111	Security Guard II	Administration	120104	

- c. Print **work.stock** and verify your results.

Add an appropriate title.

Partial Listing of **work.stock** (26 Total Observations)

Stock and Shipping Employees				
Obs	Employee_ ID	Job_Title	Department	Manager_ ID
1	120670	Shipping Manager	Stock & Shipping	120659
2	120671	Shipping Agent III	Stock & Shipping	120670
3	120672	Shipping Manager	Stock & Shipping	120659
4	120673	Shipping Agent II	Stock & Shipping	120672
5	120677	Shipping Manager	Stock & Shipping	120659
6	120678	Shipping Agent III	Stock & Shipping	120677
7	120679	Shipping Manager	Stock & Shipping	120659
8	120680	Shipping Agent I	Stock & Shipping	120679
9	120681	Shipping Agent II	Stock & Shipping	120679
10	120682	Shipping Agent I	Stock & Shipping	120679

d. Print **work.purchasing** and verify your results.

Add an appropriate title.

Partial Listing of **work.purchasing** (18 Total Observations)

Purchasing Employees				
Obs	Employee_ ID	Job_Title	Department	Manager_ ID
1	120728	Purchasing Agent II	Purchasing	120735
2	120729	Purchasing Agent I	Purchasing	120735
3	120730	Purchasing Agent I	Purchasing	120735
4	120731	Purchasing Agent II	Purchasing	120735
5	120732	Purchasing Agent III	Purchasing	120736
6	120733	Purchasing Agent I	Purchasing	120736
7	120734	Purchasing Agent III	Purchasing	120736
8	120735	Purchasing Manager	Purchasing	120261
9	120736	Purchasing Manager	Purchasing	120261
10	120737	Purchasing Manager	Purchasing	120261

5. Creating Multiple SAS Data Sets with Derived Values

The data set **orion.orders** contains information on in-store, catalog, and Internet orders as well as delivery dates.

Partial Listing of **orion.orders** (490 Total Observations)

Order_ID	Order_ Type	Employee_ID	Customer_ID	Order_ Date	Delivery_ Date
1230058123	1	121039	63	11JAN2003	11JAN2003
1230080101	2	99999999	5	15JAN2003	19JAN2003
1230106883	2	99999999	45	20JAN2003	22JAN2003
1230147441	1	120174	41	28JAN2003	28JAN2003
1230315085	1	120134	183	27FEB2003	27FEB2003

- a. Orion Star wants to study catalog and Internet orders that were delivered quickly, as well as those that went slowly.
- Create three data sets named **work.fast**, **work.slow**, and **work.veryslow**.
 - Write a WHERE statement to read only the observations with **Order_Type** equal to 2 (catalog) or 3 (Internet).
 - Create a variable named **ShipDays** that is the number of days between when the order was placed and when the order was delivered.
 - Handle the output as follows:
 - Output to **work.fast** when the value of **ShipDays** is less than 3.
 - Output to **work.slow** when the value of **ShipDays** is 5 to 7.
 - Output to **work.veryslow** when the value of **ShipDays** is greater than 7.
 - Do not output an observation when the value of **ShipDays** is 3 or 4.
 - Drop the variable **Employee_ID**.
 - There should be 80 observations in **work.fast**, 69 observations in **work.slow**, and 5 observations in **work.veryslow**.



Of the 490 observations in **orion.orders**, only 230 are read due to the WHERE statement.

- b. Print your results from **work.veryslow** with an appropriate title.

Listing of **work.veryslow** (5 Total Observations)

Orders taking more than 7 days to deliver						
Obs	Order_ID	Order_ Type	Customer_ID	Order_ Date	Delivery_ Date	Ship Days
1	1231305521	2	16	27AUG2003	04SEP2003	8
2	1236483576	2	70108	22JUL2005	02AUG2005	11
3	1236965430	3	70165	08SEP2005	18SEP2005	10
4	1237165927	3	79	27SEP2005	08OCT2005	11
5	1241298131	2	2806	29JAN2007	08FEB2007	10

6. Using a SELECT Group

Write a solution to the previous exercise. Using SELECT logic instead of IF-THEN/ELSE logic. Refer to SAS documentation to explore the use of a compound expression in a SELECT statement. Print the data set **work.veryslow**.

7. Specifying Variables and Observations

The data set **orion.employee_organization** contains information on employee job titles, departments, and managers.

Partial Listing of **orion.employee_organization** (424 Total Observations)

Employee_ ID	Job_Title	Department	Manager_ ID
120101	Director	Sales Management	120261
120102	Sales Manager	Sales Management	120101
120103	Sales Manager	Sales Management	120101
120104	Administration Manager	Administration	120101
120105	Secretary I	Administration	120101
120106	Office Assistant II	Administration	120104
120107	Office Assistant III	Administration	120104

- Create two data sets: one for the Sales department and another for the Executive department.
 - Name the data sets **work.sales** and **work.exec**.
 - Output to these data sets depending on whether the value of **Department** is Sales or Executives, respectively. Ignore all other **Department** values.
 - The **work.sales** data set should contain three variables (**Employee_ID**, **Job_Title**, and **Manager_ID**) and have 201 observations.
 - The **work.exec** data set should contain two variables (**Employee_ID** and **Job_Title**) and have four observations.
- Print only the first six observations from **work.sales**.
 - Use a data set option to display only the first six observations.
 - Add an appropriate title.

Listing of First Six Observations from **work.sales**

Sales Employees			
Obs	Employee_ ID	Job_Title	Manager_ ID
1	120121	Sales Rep. II	120102
2	120122	Sales Rep. II	120102
3	120123	Sales Rep. I	120102
4	120124	Sales Rep. I	120102
5	120125	Sales Rep. IV	120102
6	120126	Sales Rep. II	120102

- Print selected observations from **work.exec**.
 - Use data set options to process only observations 2 and 3.
 - Add an appropriate title.

Listing of Observations 2 and 3 from **work.exec**

Executives			
Employee_			

Obs	ID	Job_Title
2	120260	Chief Marketing Officer
3	120261	Chief Sales Officer

8. Specifying Variables and Observations

The data set **orion.orders** contains information on in-store, catalog, and Internet orders as well as delivery dates.


Partial Listing of **orion.orders** (490 Total Observations)

Order_ID	Order_ Type	Employee_ID	Customer_ID	Order_ Date	Delivery_ Date
1230058123	1	121039	63	11JAN2006	11JAN2006
1230080101	2	99999999	5	15JAN2006	19JAN2006
1230088186	1	120455	11	17JAN2006	17JAN2006
1230106883	2	99999999	45	20JAN2006	22JAN2006
1230147441	1	120174	41	28JAN2006	28JAN2006

- Create two data sets, **work.instore** and **work.delivery**, to analyze in-store sales.
 - Use a WHERE statement to read only observations with **Order_Type** equal to 1.
 - Create a variable **ShipDays** that is the number of days between when the order was placed and when the order was delivered.
 - Output to **work.instore** when **ShipDays** is equal to 0.
 - Output to **work.delivery** when **ShipDays** is greater than 0.
 - The **work.instore** data set should contain three variables (**Order_ID**, **Customer_ID**, and **Order_Date**).

The **work.delivery** data set should contain four variables (**Order_ID**, **Customer_ID**, **Order_Date**, and **ShipDays**).

- Test this program by reading the first 30 observations that satisfy the WHERE statement. Check the SAS log to verify that no warnings or errors were reported.

 Of the 490 observations in **orion.orders**, only 230 are read due to the WHERE statement.

- Modify the program to read the full **orion.orders** data set.

- Print your results from **work.delivery** with an appropriate title.

Partial Listing of **work.delivery** (10 Total Observations)

Deliveries from In-store Purchases					
Obs	Order_ID	Customer_ID	Order_ Date	Ship Days	
1	1231468750	52	25SEP2003	5	
2	1231657078	63	29OCT2003	4	

3	1232648239	49	07APR2004	8
---	------------	----	-----------	---

d. Use PROC FREQ to display the number of orders per year in **work.instore**. Add an appropriate title.

Hint: Format the variable **Order_Date** with a YEAR. format. Restrict the analysis to the variable **Order_Date** with a TABLES statement.

PROC FREQ Output

In-stock Store Purchases, By Year					
The FREQ Procedure					
Date Order was placed by Customer					
Order_Date	Frequency	Percent	Cumulative Frequency	Cumulative Percent	
2003	43	17.20	43	17.20	
2004	50	20.00	93	37.20	
2005	27	10.80	120	48.00	
2006	67	26.80	187	74.80	
2007	63	25.20	250	100.00	