Preparing for the SAS Programming Certification

Week 3: Controlling DATA Step

1. **Processing Data Review**
2. **Summarizing Data Review**
3. **Manipulating Data with Function Review**
4. **Creating and Using Custom Format Review**

```
/***********************************************/
/* This code defines macro variables and the   */
/* library for this course. You must run       */
/* this code each time you start SAS OnDemand  */
/* for Academics to access your practice data. */
/***********************************************/
data growth;
        set cr.profit_summary;
        Year=1;
        TotalProfit=TotalProfit*1.05;
        output;
        Year=2;
        TotalProfit=TotalProfit*1.05;
        output;
run;
```

Table: WORK.GROWTH | View: Column names | Filter: (none)

Columns

- Select all
- Product_Line
- Product_Category
- TotalProfit
- Year

Total rows: 24 Total columns: 4

	Product_Line	Product_Category	TotalProfit	Year
1	Children	Children Sports	$19,486.25	1
2	Children	Children Sports	$20,460.56	2
3	Clothes & Shoes	Clothes	$74,053.28	1
4	Clothes & Shoes	Clothes	$77,755.94	2
5	Clothes & Shoes	Shoes	$69,815.01	1
6	Clothes & Shoes	Shoes	$73,305.77	2
7	Outdoors	Outdoors	$122,118.98	1
8	Outdoors	Outdoors	$128,224.93	2

```
data sports nonsports;
        set cr.profit_summary;
        if Product_Line="Sports" then output sports;
        else output nonsports;
run;
```

Table: WORK.SPORTS | View: Column names | Filter: (none)

Columns

- Select all
- Product_Line
- Product_Category
- TotalProfit

Total rows: 8 Total columns: 3 Rows 1-8

	Product_Line	Product_Category	TotalProfit
1	Sports	Assorted Sports Articles	$85,105.81
2	Sports	Golf	$33,490.79
3	Sports	Indoor Sports	$19,474.95
4	Sports	Racket Sports	$15,015.81
5	Sports	Running - Jogging	$17,749.22
6	Sports	Swim Sports	$5,524.54
7	Sports	Team Sports	$7,952.82
8	Sports	Winter Sports	$42,917.21

Table: WORK.NONSPORTS | View: Column names | Filter: (none)

Columns

- Select all
- Product_Line
- Product_Category
- TotalProfit

Total rows: 4 Total columns: 3

	Product_Line	Product_Category	TotalProfit
1	Children	Children Sports	$18,558.33
2	Clothes & Shoes	Clothes	$70,526.93
3	Clothes & Shoes	Shoes	$66,490.49
4	Outdoors	Outdoors	$116,303.79

/*Programming Question 4.01

If necessary, start SAS Studio and submit libname.sas.

Open p201q1.sas from the programs folder. Fix the program to ensure that the following actions occur:

Values of Country should all be uppercase in the output tables.

Include only rows where TermDate is missing.

All rows where Country is US are written to the emp_US table.

All rows where Country is AU are written to the emp_AU table.

How many rows are in the emp_US table?

How many rows are in the emp_AU table?

*/

```
data emp_US emp_AU;
        set cr.employee(keep=EmpID Name JobTitle Salary Department Country TermDate);
        where TermDate=.;
        Country=upcase(country);
        if Country="US" then output emp_US;
        else output emp_AU;
run;
```

Table: WORK.EMP_US | View: Column names | Filter: (none)

Columns: Select all, EmpID, Name, JobTitle, Department, Salary, Country, TermDate

Total rows: 235 Total columns: 7 Rows 1-100

	EmpID	Name	JobTitle	Department	Salary	Country	TermDate
1	120259	Miller, Anthony	Chief Executive Officer	Executives	$433,800.00	US	.
2	120260	Fletcher, Christine	Chief Marketing Officer	Executives	$207,885.00	US	.
3	120261	Highpoint, Harry	Chief Sales Officer	Executives	$243,190.00	US	.
4	120262	Crown, Max	Chief Financial Officer	Executives	$268,455.00	US	.
5	120264	Croome, Latonya	Financial Analyst II	Group Financials	$46,887.50	US	.
6	120266	Krafve, Bao	Secretary IV	Secretary of the Board	$39,687.50	US	.
7	120267	Rink, Belanda	Secretary III	Secretary of the Board	$35,731.25	US	.
8	120268	Villeneuve, Jacques	Senior Strategist	Strategy	$95,131.25	US	.

Table: WORK.EMP_AU | View: Column names | Filter: (none)

Columns: Select all, EmpID, Name, JobTitle, Department, Salary, Country, TermDate

Total rows: 73 Total columns: 7 Rows 1

	EmpID	Name	JobTitle	Department	Salary	Country	TermDate
1	120101	Lu, Patrick	Director	Sales Management	$203,800.00	AU	.
2	120102	Zhou, Tom	Sales Manager	Sales Management	$135,318.75	AU	.
3	120103	Dawes, Wilson	Sales Manager	Sales Management	$109,968.75	AU	.
4	120104	Billington, Kareen	Administration Manage	Administration	$57,787.50	AU	.
5	120105	Povey, Liz	Secretary I	Administration	$33,887.50	AU	.
6	120106	Hornsey, John	Office Assistant II	Administration	$33,700.00	AU	.
7	120107	Sheedy, Sherie	Office Assistant III	Administration	$38,093.75	AU	.
8	120108	Gromek, Gladys	Warehouse Assistant II	Administration	$34,575.00	AU	.

/*Programming Question 4.02

If necessary, start SAS Studio.

Write a new program that reads the sashelp.heart table and separates rows into two tables, dead and alive, based on the value of Status.

Drop Status from both tables and drop DeathCause and AgeAtDeath from the alive table.

How many rows are in the dead table?

What is the value of Cholesterol for row 100 in the alive table?

How many columns are in the alive table?

*/

```
data dead alive;
        set sashelp.heart;
        if Status="Dead" then do;
                drop Status;
                output dead;
        end;
        else if Status="Alive" then do;
                drop Status DeathCause AgeAtDeath;
                output alive;
        end;
run;
```

Table: WORK.DEAD | View: Column names | Filter: (none)

Columns

- Select all
- AgeCHDdiag
- Sex
- AgeAtStart
- Height
- Weight
- Diastolic
- Systolic
- MRW
- Smoking
- Cholesterol

Total rows: 1991 Total columns: 14 Rows 1-100

	AgeCHDdiag	Sex	AgeAtStart	Height	Weight	Diastolic	Systolic	MRW	Smoking	Chole:
1	.	Female	29	62.5	140	78	124	121	0	
2	.	Female	41	59.75	194	92	144	183	0	
3	.	Male	53	65.5	130	80	114	99	0	
4	.	Male	52	62.5	129	78	124	106	5	
5	56	Male	56	67.25	122	72	120	87	15	
6	74	Male	46	66.5	157	84	142	116	30	
7	71	Female	49	60.5	153	110	196	140	5	
8	.	Female	59	67.75	153	82	172	113	0	
9	67	Female	49	61	142	92	138	127	30	
10	.	Male	43	65.5	172	78	118	131	10	
11	73	Male	55	67.5	193	60	148	138	15	

Table: WORK.ALIVE | View: Column names | Filter: (none)

Columns

- Select all
- AgeCHDdiag
- Sex
- AgeAtStart
- Height
- Weight
- Diastolic
- Systolic
- MRW
- Smoking
- Cholesterol

Total rows: 3218 Total columns: 14 Rows 1-10

	AgeCHDdiag	Sex	AgeAtStart	Height	Weight	Diastolic	Systolic	MRW	Smoking
77	.	Female	44	61.5	118	68	126	105	5
78	.	Male	36	72.75	169	88	130	105	0
79	68	Female	54	64.5	147	100	182	119	0
80	.	Female	30	67	132	62	112	97	10
81	.	Female	47	64.25	150	58	100	121	5
82	.	Female	35	65.75	123	84	132	96	20
83	.	Female	57	66.5	137	78	138	104	0
84	.	Female	46	63	155	98	168	129	0
85	.	Female	55	60	115	100	154	106	0
86	74	Female	56	60.5	121	78	132	111	0
87	.	Female	46	62.75	101	68	122	87	15

/*Programming Question 4.03

If necessary, start SAS Studio and submit libname.sas. Write a new program to do the following:

Read the cr.employee_current table and create a summary table named salary that calculates TotalSalary (sum of Salary) for each department.

In the output table, include only one row for each department.

Create a new table named salaryforecast that reads the salary summary table.

For each row in the salary table, write three rows to the salaryforecast table.

For the first row written to the salaryforecast table, create a column named Year that is equal to 1.

Increase TotalSalary by 3% (multiply by 1.03).

Write additional rows to the output table with Year equal to 2 and 3 and TotalSalary increased by an additional 3%.

Display TotalSalary as currency values rounded to the nearest whole number.

What is the total salary for the Strategy Department before the forecast is performed?

Note: Round your answer to the nearest whole number and enter only numbers.

How many rows are in the salaryforecast table?

*/

```
proc means data=cr.employee_current noprint;
   var Salary;
   class Department;
   output out=salary sum=TotalSalary;
   ways 1;
run;
```

```
data salaryforecast;
   set salary;
   format TotalSalary dollar12.;
   Year=1;
   TotalSalary=TotalSalary*1.03;
   output;
   Year=2;
   TotalSalary=TotalSalary*1.03;
   output;
   Year=3;
   TotalSalary=TotalSalary*1.03;
   output;
run;
```

Table: WORK.SALARY | View: Column names | Filter: (none)

Columns

- Select all
- Department
- _TYPE_
- _FREQ_
- TotalSalary

Property	Value
Label	

Total rows: 17 Total columns: 4

	Department	_TYPE_	_FREQ_	TotalSalary
1	Accounts	1	14	$723,287.50
2	Accounts Management	1	7	$385,362.50
3	Administration	1	31	$1,159,350.00
4	Concession Manageme	1	8	$343,200.00
5	Engineering	1	9	$345,356.25
6	Executives	1	4	$1,153,330.00
7	Group Financials	1	1	$46,887.50
8	Group HR Managemen	1	15	$764,131.25
9	IS	1	20	$1,265,062.50
10	Logistics Management	1	12	$836,025.00
11	Marketing	1	19	$1,038,906.25
12	Purchasing	1	15	$739,068.75
13	Sales	1	122	$4,246,800.00
14	Sales Management	1	10	$1,289,425.00
15	Secretary of the Board	1	2	$75,418.75
16	Stock & Shipping	1	18	$788,537.50
17	Strategy	1	1	$95,131.25

Table: WORK.SALARYFORECAST | View: Column names | Filter: (none)

Columns

- Select all
- Department
- _TYPE_
- _FREQ_
- TotalSalary
- Year

Total rows: 51 Total columns: 5

	Department	_TYPE_	_FREQ_	TotalSalary	Year
1	Accounts	1	14	$744,986	1
2	Accounts	1	14	$767,336	2
3	Accounts	1	14	$790,356	3
4	Accounts Management	1	7	$396,923	1
5	Accounts Management	1	7	$408,831	2
6	Accounts Management	1	7	$421,096	3
7	Administration	1	31	$1,194,131	1
8	Administration	1	31	$1,229,954	2
9	Administration	1	31	$1,266,853	3
10	Concession Manageme	1	8	$353,496	1
11	Concession Manageme	1	8	$364,101	2
12	Concession Manageme	1	8	$375,024	3
13	Engineering	1	9	$355,717	1
14	Engineering	1	9	$366,388	2
15	Engineering	1	9	$377,380	3

```
/* Summarizing Data */
data new;
        putlog "NOTE: Value of HeightCM at the top of the DATA Step";
        putlog HeightCM=;
        set sashelp.class(obs=3);
        HeightCM=Height*2.54;
        putlog "NOTE: Value of HeightCM at the bottom of the DATA Step";
        putlog HeightCM=;
run;
```

Table: WORK.NEW | View: Column names | Filter: (none)

Columns

- Select all
- HeightCM
- Name
- Sex
- Age
- Height
- Weight

Total rows: 3 Total columns: 6 Rows 1-3

	HeightCM	Name	Sex	Age	Height	Weight
1	175.26	Alfred	M	14	69	112.5
2	143.51	Alice	F	13	56.5	84
3	165.862	Barbara	F	13	65.3	98

Errors, Warnings, Notes

- Errors
- Warnings
- Notes (10)

```
NOTE: Value of HeightCM at the top of the DATA Step
HeightCM=.
NOTE: Value of HeightCM at the bottom of the DATA Step
HeightCM=175.26
NOTE: Value of HeightCM at the top of the DATA Step
HeightCM=.
NOTE: Value of HeightCM at the bottom of the DATA Step
HeightCM=143.51
NOTE: Value of HeightCM at the top of the DATA Step
HeightCM=.
NOTE: Value of HeightCM at the bottom of the DATA Step
HeightCM=165.862
NOTE: Value of HeightCM at the top of the DATA Step
HeightCM=.
NOTE: There were 3 observations read from the data set SASHELP.CLASS.
NOTE: The data set WORK.NEW has 3 observations and 6 variables.
```

```
/* Summarizing Data */

data new;

        putlog "NOTE: Value of HeightCM at the top of the DATA Step";

        putlog HeightCM=;

        retain HeightCM;

        set sashelp.class(obs=3);

        HeightCM=Height*2.54;

        putlog "NOTE: Value of HeightCM at the bottom of the DATA Step";

        putlog HeightCM=;

run;
```

CODE LOG RESULTS OUTPUT DATA

Errors, Warnings, Notes

Errors

Warnings

Notes (10)

```
72
73          /* Summarizing Data */
74          data new;
75          putlog "NOTE: Value of HeightCM at the top of the DATA Step";
76          putlog HeightCM=;
77          retain HeightCM;
78          set sashelp.class(obs=3);
79          HeightCM=Height*2.54;
80          putlog "NOTE: Value of HeightCM at the bottom of the DATA Step";
81          putlog HeightCM=;
82          run;

NOTE: Value of HeightCM at the top of the DATA Step
HeightCM=.
NOTE: Value of HeightCM at the bottom of the DATA Step
HeightCM=175.26
NOTE: Value of HeightCM at the top of the DATA Step
HeightCM=175.26
NOTE: Value of HeightCM at the bottom of the DATA Step
HeightCM=143.51
NOTE: Value of HeightCM at the top of the DATA Step
HeightCM=143.51
NOTE: Value of HeightCM at the bottom of the DATA Step
HeightCM=165.862
NOTE: Value of HeightCM at the top of the DATA Step
HeightCM=165.862
NOTE: There were 3 observations read from the data set SASHELP.CLASS.
NOTE: The data set WORK.NEW has 3 observations and 6 variables.
```

```
/* Summarizing Data */

data new;

        putlog "NOTE: Value of HeightCM at the top of the DATA Step";

        putlog HeightCM=;

        retain HeightCM 0;

        set sashelp.class(obs=3);

        HeightCM=Height*2.54;

        putlog "NOTE: Value of HeightCM at the bottom of the DATA Step";
```

```
        putlog HeightCM=;
run;
```

Errors, Warnings, Notes

- Errors
- Warnings
- Notes (10)

```
72
73          /* Summarizing Data */
74          data new;
75          putlog "NOTE: Value of HeightCM at the top of the DATA Step";
76          putlog HeightCM=;
77          retain HeightCM 0;
78          set sashelp.class(obs=3);
79          HeightCM=Height*2.54;
80          putlog "NOTE: Value of HeightCM at the bottom of the DATA Step";
81          putlog HeightCM=;
82          run;

NOTE: Value of HeightCM at the top of the DATA Step
HeightCM=0
NOTE: Value of HeightCM at the bottom of the DATA Step
HeightCM=175.26
NOTE: Value of HeightCM at the top of the DATA Step
HeightCM=175.26
NOTE: Value of HeightCM at the bottom of the DATA Step
HeightCM=143.51
NOTE: Value of HeightCM at the top of the DATA Step
HeightCM=143.51
NOTE: Value of HeightCM at the bottom of the DATA Step
HeightCM=165.862
NOTE: Value of HeightCM at the top of the DATA Step
HeightCM=165.862
NOTE: There were 3 observations read from the data set SASHELP.CLASS.
NOTE: The data set WORK.NEW has 3 observations and 6 variables.
```

```
proc sort data=cr.profit out=decDaily;
        where month(Order_Date)=12;
        by Order_Date;
run;


data DecSales;
        set decDaily;
        retain MTDSales 0;
```

```
        MTDSales=sum(MTDSales,Profit);
        format MTDSales dollar12.;
        keep Order_ID Order_Date Profit MTDSales;
run;
```

Table: WORK.DECSALES | View: Column names | Filter: (none)

Columns

- Select all
- Order_ID
- Order_Date
- Profit
- MTDSales

Total rows: 1200 Total columns: 4

	Order_ID	Order_Date	Profit	MTDSales
1	1244072351	01DEC2018	$142.55	$143
2	1244072328	01DEC2018	$399.00	$542
3	1244072170	01DEC2018	$51.80	$593
4	1244072131	01DEC2018	$139.05	$732
5	1244071500	01DEC2018	$130.40	$863
6	1244071345	01DEC2018	$14.75	$878
7	1244071081	01DEC2018	$67.04	$945
8	1244070912	01DEC2018	$366.80	$1,311

```
proc sort data=cr.profit out=decDaily;
        where month(Order_Date)=12;
        by Order_Date;
run;

data DecSales;
        set decDaily;
        by Order_Date;
        MTDSales+Profit;
        if first.Order_Date=1 then DailySales=0;
        DailySales+Profit;
        if last.Order_Date=1;
        format MTDSales DailySales dollar12.;
        keep Order_ID Order_Date MTDSales DailySales;
run;
```

Table: WORK.DECSALES View: Column names Filter: (none)

Columns

- Select all
- Order_ID
- Order_Date
- MTDSales
- DailySales

Total rows: 31 Total columns: 4

	Order_ID	Order_Date	MTDSales	DailySales
1	1244066541	01DEC2018	$14,613	$14,613
2	1244073032	02DEC2018	$21,221	$6,608
3	1244081671	03DEC2018	$31,434	$10,213
4	1244091309	04DEC2018	$42,775	$11,340
5	1244100643	05DEC2018	$52,606	$9,832
6	1244109965	06DEC2018	$62,907	$10,300
7	1244119362	07DEC2018	$76,358	$13,452
8	1244130979	08DEC2018	$91,778	$15,420
9	1244135864	09DEC2018	$102,956	$11,178

/*Programming Question 4.04

If necessary, start SAS Studio. Write a new program to do the following:

Read the sashelp.stocks table and create a new table named stocks_total.

Include only rows where the Date is in 2005.

Create an accumulating column for Volume named YTDVolume that is a running total of Volume for each stock.

How many rows are in the stocks_total table?

What is the value for YTDVolume for Intel on 01AUG2005?

*/

```
proc sort data=sashelp.stocks out=stocks_sort;
        by Stock Date;
        where Year(Date)=2005;
run;

data stocks_total;
        set stocks_sort;
        by Stock;
        if first.Stock=1 then YTDVolume=0;
        YTDVolume+Volume;
```

```
        format YTDVolume comma16.;
run;
```

Table: WORK.STOCKS_TOTAL | View: Column names | Filter: (none)

Columns

- Select all
- Stock
- Date
- Open
- High
- Low
- Close
- Volume
- AdjClose
- YTDVolume

Property	Value
Label	
Name	
Length	
Type	

Total rows: 36 Total columns: 9

	Stock	Date	Open	High	Low	Close	Volume	AdjClose	YTDVolume
1	IBM	03JAN05	$98.97	$99.10	$91.44	$93.42	5,960,945	$91.62	5,960,945
2	IBM	01FEB05	$93.67	$94.97	$91.55	$92.58	4,455,657	$90.97	10,416,602
3	IBM	01MAR05	$92.64	$93.73	$89.09	$91.38	5,025,627	$89.79	15,442,229
4	IBM	01APR05	$91.49	$91.76	$71.85	$76.38	10,709,200	$75.05	26,151,429
5	IBM	02MAY05	$76.88	$78.11	$72.50	$75.55	6,896,904	$74.43	33,048,333
6	IBM	01JUN05	$75.57	$77.73	$73.45	$74.20	6,439,536	$73.10	39,487,869
7	IBM	01JUL05	$74.30	$85.11	$74.16	$83.46	8,056,590	$82.23	47,544,459
8	IBM	01AUG05	$83.00	$84.20	$79.87	$80.62	4,801,386	$79.62	52,345,845
9	IBM	01SEP05	$80.16	$82.11	$76.93	$80.22	5,772,280	$79.22	58,118,125
10	IBM	03OCT05	$80.22	$84.60	$78.70	$81.88	7,019,666	$80.86	65,137,791
11	IBM	01NOV05	$81.85	$89.94	$80.64	$88.90	5,556,471	$88.01	70,694,262
12	IBM	01DEC05	$89.15	$89.92	$81.56	$82.20	5,976,252	$81.37	76,670,514
13	Intel	03JAN05	$23.64	$23.79	$21.89	$22.45	82,481,585	$21.81	82,481,585
14	Intel	01FEB05	$22.49	$24.63	$22.17	$23.99	78,924,846	$23.39	161,406,431
15	Intel	01MAR05	$24.36	$25.47	$22.96	$23.23	63,823,945	$22.65	225,230,376
16	Intel	01APR05	$23.34	$23.90	$21.94	$23.52	67,651,157	$22.93	292,881,533
17	Intel	02MAY05	$23.50	$27.40	$23.35	$26.96	64,584,867	$26.38	357,466,400
18	Intel	01JUN05	$26.80	$27.75	$25.73	$26.02	55,191,463	$25.46	412,657,863
19	Intel	01JUL05	$26.23	$28.84	$26.08	$27.14	55,455,755	$26.55	468,113,618
20	Intel	01AUG05	$27.24	$27.66	$25.31	$25.72	45,049,221	$25.24	513,162,839

/*Programming Question 4.05

If necessary, start SAS Studio. Write a new program to do the following:

Read the sashelp.shoes table and create a new table named highlow.

The highlow table should include one row for the lowest value of Sales for each Product value,

and another row for the highest value of Sales for each Product value.

Create a column named HighLow that includes the value High or Low depending on the value of Sales.

How many rows are in the highlow table?

Which subsidiary has the highest Sales value for Sandals?

Which region has the lowest Sales value for Boot?

*/

```
proc sort data=sashelp.shoes out=highlow;
   by Product Sales;
run;
```

```
data highlow;
   length HighLow $ 4;
   set highlow;
   by Product;
   if first.product then do;
      HighLow="Low";
      output;
   end;
   if last.product then do;
      HighLow="High";
      output;
   end;
   keep Region Product HighLow Sales Subsidiary;
run;
```

Table: WORK.HIGHLOW | View: Column names | Filter: (none)

Columns

- Select all
- HighLow
- Region
- Product
- Subsidiary
- Sales

Property | Value

Total rows: 16 Total columns: 5 Rows 1-16

	HighLow	Region	Product	Subsidiary	Sales
1	Low	Western Europe	Boot	Madrid	$1,179
2	High	Canada	Boot	Vancouver	$286,497
3	Low	Africa	Men's Casual	Khartoum	$9,244
4	High	Middle East	Men's Casual	Tel Aviv	$1,298,717
5	Low	Asia	Men's Dress	Bangkok	$3,033
6	High	Canada	Men's Dress	Vancouver	$757,798
7	Low	Pacific	Sandal	Auckland	$325
8	High	Central America/Caribbean	Sandal	Kingston	$200,156
9	Low	Asia	Slipper	Bangkok	$3,019
10	High	Canada	Slipper	Vancouver	$700,513
11	Low	Middle East	Sport Shoe	Al-Khobar	$449
12	High	Western Europe	Sport Shoe	Copenhagen	$101,922
13	Low	Asia	Women's Casual	Bangkok	$5,389
14	High	Western Europe	Women's Casual	Copenhagen	$502,636
15	Low	Pacific	Women's Dress	Jakarta	$4,659
16	High	Canada	Women's Dress	Vancouver	$756,347

/*Programming Question 4.06

If necessary, start SAS Studio and submit libname.sas. Open p202q3.sas from the programs folder. Fix the program to ensure that the following actions occur:

There should be one row for each department in the dept_salary table.

The TotalDeptSalary column should be the total of Salary within each department.

The LowSalaryJob column should be the job title for the minimum salary paid within each department.

The HighSalaryJob column should be the job title for the maximum salary paid within each department.

What is the value of TotalDeptSalary for the Accounts Department? Note: enter the numeric value with no formatting.

What is the value of LowSalaryJob for the Engineering Department?

What is the value of HighSalaryJob for the Logistics Management Department?

```
*/

proc sort data=cr.employee_current out=emp_sort;
   by Department Salary;
run;

data dept_salary;
   set emp_sort;
   retain LowSalaryJob;
   by Department;
   if first.Department then do;
      TotalDeptSalary=0;
      LowSalaryJob=JobTitle;
   end;
   TotalDeptSalary+Salary;
   if last.department then do;
      HighSalaryJob=JobTitle;
      output;
   end;
   keep Department TotalDeptSalary HighSalaryJob LowSalaryJob;
   format TotalDeptSalary dollar12.;
run;
```

Table: WORK.DEPT_SALARY | View: Column names | Filter: (none)

Columns

- Select all
- Department
- LowSalaryJob
- TotalDeptSalary
- HighSalaryJob

Property	Value
Label	

Total rows: 17 Total columns: 4

	Department	LowSalaryJob	TotalDeptSalary	HighSalaryJob
1	Accounts	Office Assistant II	$723,288	Finance Manager
2	Accounts Management	Accountant I	$385,363	Auditing Manager
3	Administration	Security Guard I	$1,159,350	Administration Manager
4	Concession Management	Concession Assistant I	$343,200	Concession Director
5	Engineering	Technician I	$345,356	Technical Manager
6	Executives	Chief Marketing Officer	$1,153,330	Chief Executive Officer
7	Group Financials	Financial Analyst II	$46,888	Financial Analyst II
8	Group HR Management	HR Generalist I	$764,131	Recruitment Manager
9	IS	Office Assistant III	$1,265,063	Senior Project Manager
10	Logistics Management	Secretary II	$836,025	Director
11	Marketing	Marketing Assistant I	$1,038,906	Senior Marketing Manager
12	Purchasing	Purchasing Agent I	$739,069	Purchasing Manager
13	Sales	Sales Rep. I	$4,246,800	Sales Rep. II
14	Sales Management	Secretary II	$1,289,425	Vice President
15	Secretary of the Board	Secretary III	$75,419	Secretary IV
16	Stock & Shipping	Warehouse Assistant I	$788,538	Shipping Manager
17	Strategy	Senior Strategist	$95,131	Senior Strategist

```
/* Manipulating Data with Function */
data qtr_detail;
        set cr.qtr_sales;
        TotalPurchase=sum(of qtr:);
        *TotalPurchase=sum(qtr1, qtr2, qtr3, qtr4);
        AvgPurchase=round(mean(of qtr:), .01);
        format TotalPurchase AvgPurchase dollar12.2;
run;
```

Table: WORK.QTR_DETAIL | View: Column names | Filter: (none)

Columns

- Select all
- Customer_ID
- Name
- BirthDate
- Qtr1
- Qtr2
- Qtr3
- Qtr4
- TotalPurchase
- AvgPurchase

Total rows: 75 Total columns: 9 Rows 1-75

	Customer_ID	Name	BirthDate	Qtr1	Qtr2	Qtr3	Qtr4	TotalPurchase	AvgPurchase
1	4	James Kvarniq	27JUN1976	$578.36	$361.83	$407.02	$694.08	$2,041.29	$510.32
2	5	Sandrina Stephano	09JUL1981	$61.31	$59.43	$250.21	$28.12	$399.07	$99.77
3	9	Cornelia Krahl	27FEB1976	$103.25	$74.18	$0.00	$244.21	$421.64	$105.41
4	10	Karen Ballinger	18OCT1986	$0.00	$196.46	$260.94	$210.20	$667.60	$166.90
5	11	Elke Wallstab	16AUG1976	$118.66	$16.80	$57.04	$565.07	$757.57	$189.39
6	12	David Black	12APR1971	$96.91	$26.15	$69.94	$216.11	$409.11	$102.28
7	13	Markus Sepke	21JUL1990	$73.49	$170.18	$96.54	$194.13	$534.34	$133.59
8	16	Ulrich Heyde	16JAN1941	$0.00	$27.99	$293.94	$861.20	$1,183.13	$295.78
9	17	Jimmie Evans	17AUG1956	$319.81	$269.63	$61.35	$693.89	$1,344.68	$336.17
10	18	Tonie Asmussen	02FEB1956	$644.63	$45.80	$97.84	$361.12	$1,149.39	$287.35

```
/* Manipulating Data with Function */
data qtr_detail;
        set cr.qtr_sales;
        TotalPurchase=sum(of qtr:);
        *TotalPurchase=sum(qtr1, qtr2, qtr3, qtr4);
```

```
        AvgPurchase=round(mean(of qtr:), .01);
        Customer_Age=int(yrdif(BirthDate, today(), "age"));
        Promo_Date=mdy(month(BirthDate), 1, year(today()));
        format TotalPurchase AvgPurchase dollar12.2 Promo_Date mmddyy10.;
run;
```

Table: WORK.QTR_DETAIL | View: Column names | Filter: (none)

Columns: Select all, Customer_ID, Name, BirthDate, Qtr1, Qtr2, Qtr3, Qtr4, TotalPurchase, AvgPurchase, Customer_Age, Promo_Date

Total rows: 75 Total columns: 11 — Rows 1-75

	Customer_ID	Name	BirthDate	Qtr1	Qtr2	Qtr3	Qtr4	TotalPurchase	AvgPurchase	Customer_Age	Promo_Date
1	4	James Kvarniq	27JUN1976	$578.36	$361.83	$407.02	$694.08	$2,041.29	$510.32	44	06/01/2021
2	5	Sandrina Stephano	09JUL1981	$61.31	$59.43	$250.21	$28.12	$399.07	$99.77	39	07/01/2021
3	9	Cornelia Krahl	27FEB1976	$103.25	$74.18	$0.00	$244.21	$421.64	$105.41	45	02/01/2021
4	10	Karen Ballinger	18OCT1986	$0.00	$196.46	$260.94	$210.20	$667.60	$166.90	34	10/01/2021
5	11	Elke Wallstab	16AUG1976	$118.66	$16.80	$57.04	$565.07	$757.57	$189.39	44	08/01/2021
6	12	David Black	12APR1971	$96.91	$26.15	$69.94	$216.11	$409.11	$102.28	50	04/01/2021
7	13	Markus Sepke	21JUL1990	$73.49	$170.18	$96.54	$194.13	$534.34	$133.59	30	07/01/2021
8	16	Ulrich Heyde	16JAN1941	$0.00	$27.99	$293.94	$861.20	$1,183.13	$295.78	80	01/01/2021
9	17	Jimmie Evans	17AUG1956	$319.81	$269.63	$61.35	$693.89	$1,344.68	$336.17	64	08/01/2021
10	18	Tonie Asmussen	02FEB1956	$644.63	$45.80	$97.84	$361.12	$1,149.39	$287.35	65	02/01/2021
11	19	Oliver S. F►►ling	23FEB1966	$382.01	$0.00	$434.86	$492.04	$1,308.91	$327.23	55	02/01/2021
12	20	Michael Dineley	17APR1961	$10.32	$515.13	$976.50	$75.89	$1,577.84	$394.46	60	04/01/2021

```
/* Manipulating Data with Function */
data qtr_detail;
        set cr.qtr_sales;
        TotalPurchase=sum(of qtr:);
        *TotalPurchase=sum(qtr1, qtr2, qtr3, qtr4);
        AvgPurchase=round(mean(of qtr:), .01);
        Customer_Age=int(yrdif(BirthDate, today(), "age"));
        Promo_Date=mdy(month(BirthDate), 1, year(today()));
        FirstName=scan(Name, 1, " ");
        ID=put(Customer_ID, z5.);
        format TotalPurchase AvgPurchase dollar12.2 Promo_Date mmddyy10.;
        drop qtr: Customer_ID;
run;
```

Table: WORK.QTR_DETAIL | View: Column names | Filter: (none)

Columns: Select all, Name, BirthDate, TotalPurchase, AvgPurchase, Customer_Age, Promo_Date, FirstName, ID

Total rows: 75 Total columns: 8 Rows 1-75

	Name	BirthDate	TotalPurchase	AvgPurchase	Customer_Age	Promo_Date	FirstName	ID
1	James Kvarniq	27JUN1976	$2,041.29	$510.32	44	06/01/2021	James	00004
2	Sandrina Stephano	09JUL1981	$399.07	$99.77	39	07/01/2021	Sandrina	00005
3	Cornelia Krahl	27FEB1976	$421.64	$105.41	45	02/01/2021	Cornelia	00009
4	Karen Ballinger	18OCT1986	$667.60	$166.90	34	10/01/2021	Karen	00010
5	Elke Wallstab	16AUG1976	$757.57	$189.39	44	08/01/2021	Elke	00011
6	David Black	12APR1971	$409.11	$102.28	50	04/01/2021	David	00012
7	Markus Sepke	21JUL1990	$534.34	$133.59	30	07/01/2021	Markus	00013
8	Ulrich Heyde	16JAN1941	$1,183.13	$295.78	80	01/01/2021	Ulrich	00016
9	Jimmie Evans	17AUG1956	$1,344.68	$336.17	64	08/01/2021	Jimmie	00017
10	Tonie Asmussen	02FEB1956	$1,149.39	$287.35	65	02/01/2021	Tonie	00018

```
/* Manipulating Data with Function */
data qtr_detail;
        set cr.qtr_sales;
        TotalPurchase=sum(of qtr:);
        *TotalPurchase=sum(qtr1, qtr2, qtr3, qtr4);
        AvgPurchase=round(mean(of qtr:), .01);
        Customer_Age=int(yrdif(BirthDate, today(), "age"));
        Promo_Date=mdy(month(BirthDate), 1, year(today()));
        FirstName=scan(Name, 1, " ");
        ID=put(Customer_ID, z5.);
        format TotalPurchase AvgPurchase dollar12.2 Promo_Date mmddyy10.;
        drop qtr: Customer_ID;
run;

data qtr_detail;
        retain ID Name FirstName BirthDate Customer_Age Promo_Date TotalPurchase AvgPurchase;
        set qtr_detail;
run;
```

Table: WORK.QTR_DETAIL | View: Column names | Filter: (none)

Columns: Select all, ID, Name, FirstName, BirthDate, Customer_Age, Promo_Date, TotalPurchase, AvgPurchase

Total rows: 75 Total columns: 8 Rows 1-75

	ID	Name	FirstName	BirthDate	Customer_Age	Promo_Date	TotalPurchase	AvgPurchase
1	00004	James Kvarniq	James	27JUN1976	44	06/01/2021	$2,041.29	$510.32
2	00005	Sandrina Stephano	Sandrina	09JUL1981	39	07/01/2021	$399.07	$99.77
3	00009	Cornelia Krahl	Cornelia	27FEB1976	45	02/01/2021	$421.64	$105.41
4	00010	Karen Ballinger	Karen	18OCT1986	34	10/01/2021	$667.60	$166.90
5	00011	Elke Wallstab	Elke	16AUG1976	44	08/01/2021	$757.57	$189.39
6	00012	David Black	David	12APR1971	50	04/01/2021	$409.11	$102.28
7	00013	Markus Sepke	Markus	21JUL1990	30	07/01/2021	$534.34	$133.59
8	00016	Ulrich Heyde	Ulrich	16JAN1941	80	01/01/2021	$1,183.13	$295.78
9	00017	Jimmie Evans	Jimmie	17AUG1956	64	08/01/2021	$1,344.68	$336.17

```
data _6months;
        set cr.profit;
        where Order_Date>=intnx("month", today(), -36, "same");
        *where Order_Date>=intnx("month", today(), -6);
        keep Order_ID Order_Date Delivery_Date BusDays;
        BusDays=intck("weekday", Order_Date, Delivery_Date);
run;
```

Table: WORK._6MONTHS | View: Column names | Filter: (none)

Columns

- Select all
- Order_ID
- Order_Date
- Delivery_Date
- BusDays

Total rows: 7651 Total columns: 4

	Order_ID	Order_Date	Delivery_Date	BusDays
1	1244333347	31DEC2018	31DEC2018	0
2	1244336610	31DEC2018	04JAN2019	4
3	1244336421	31DEC2018	04JAN2019	4
4	1244336321	31DEC2018	04JAN2019	4
5	1244336156	31DEC2018	31DEC2018	0

/* Programming Question 4.07

If necessary, start SAS Studio. Write a new program to do the following:

Read the sashelp.fish table and create a new temporary table named fish.

Compute a new column named Length that is the mean of Length1 through Length3. Round the stored values to the nearest hundredth.

Create a summary report that calculates the average of Length for each value of Species. Round the average to two decimal places.

What is the value of Length for the second row in the fish table?

How many Perch fish were measured?

What is the average length for all Pike fish?

*/

```
data fish;
        set sashelp.fish;
        Length=round(mean(of Length:), 0.01);
run;


proc means data=fish mean maxdec=2;
```

```
        var Length;
        class Species;
run;
```

The MEANS Procedure

Analysis Variable : Length		
Species	N Obs	Mean
Bream	35	33.92
Parkki	11	20.62
Perch	56	27.73
Pike	17	45.56
Roach	20	22.63
Smelt	14	12.07
Whitefish	6	31.48

Table: WORK.FISH | View: Column names | Filter: (none)

Columns: Select all, Species, Weight, Length1, Length2, Length3, Height, Width, Length

Total rows: 159 Total columns: 8 Rows 1-100

	Species	Weight	Length1	Length2	Length3	Height	Width	Length
1	Bream	242	23.2	25.4	30	11.52	4.02	26.2
2	Bream	290	24	26.3	31.2	12.48	4.3056	27.17
3	Bream	340	23.9	26.5	31.1	12.3778	4.6961	27.17
4	Bream	363	26.3	29	33.5	12.73	4.4555	29.6
5	Bream	430	26.5	29	34	12.444	5.134	29.83
6	Bream	450	26.8	29.7	34.7	13.6024	4.9274	30.4
7	Bream	500	26.8	29.7	34.5	14.1795	5.2785	30.33
8	Bream	390	27.6	30	35	12.67	4.69	30.87
9	Bream	450	27.6	30	35.1	14.0049	4.8438	30.9

/* Programming Question 4.08

If necessary, start SAS Studio. Write a new program to do the following:

Read the sashelp.baseball table and create a new table named outfield.

Filter rows to include only players where the second letter of Position is F.

Create a new column named Player that rearranges the value in the Name column so that first name is first and then last name separated by a space.

Compute a new column named BatAvg as nHits divided by nAtBat. Round the calculated value to three decimal places.

Sort the outfield table by descending BatAvg.

How many rows are in the outfield table?

What is the value of Player in the first row?

What is the value of BatAvg for Kirby Puckett?

*/

data outfield;

```
        set sashelp.baseball;
        where substr(Position, 2, 1)="F";
        Player=catx(" ", scan(Name, 2, ","), scan(Name, 1, ","));
        BatAvg=round(nHits/nAtBat, .001);
run;

proc sort data=outfield out=outfield_sort;
        by descending BatAvg;
run;

*Solution;
```

```
data outfield;
    set sashelp.baseball;
    where substr(Position, 2, 1)="F";
    Player=catx(" ", scan(Name, 2), scan(Name, 1));
    BatAvg=round(nHits/nAtBat, .001);
    keep Player BatAvg Position;
run;

proc sort data=outfield;
    by descending BatAvg;
run;
```

Table: WORK.OUTFIELD_SORT | View: Column names | Filter: (none)

Columns: Select all, Name, Team, nAtBat, nHits, nHome

Total rows: 107 Total columns: 26 Rows 1-100

CrRbi	CrBB	League	Division	Position	nOuts	nAssts	nError	Salary	Div	logSalary	Player	BatAvg
314	469	National	East	LF	270	13	6	.	NE	.	Tim Raines	0.334
230	193	National	West	RF	337	19	4	740	NW	6.6066501862	Tony Gwynn	0.329
201	91	American	West	CF	429	8	6	365	AW	5.8998973536	Kirby Puckett	0.328
1289	564	American	East	LF	330	16	8	2412.5	AE	7.7884188332	Jim Rice	0.324
23	22	National	West	OF	88	0	3	86.5	NW	4.4601444139	Kal Daniels	0.32
419	240	American	West	LF	237	8	1	600	AW	6.3969296552	Gary Ward	0.316

/* Programming Question 4.09

If necessary, start SAS Studio and submit libname.sas. Open p203q3.sas from the programs folder.

Fix the program to ensure that the following actions occur:

HireDate is converted to a numeric SAS date value.

Salary is the numeric equivalent of the character column AnnualSalary.

What is the unformatted SAS date value of HireDate for the first row of the emp_new table?

Which informat can be used in the INPUT function to create the Salary column? Note: Do not type the informat width or period.

```
*/

data emp_new;

        set cr.employee_new;

        EmpID=substr(EmpID, 4);

        HireDate2=input(HireDate, anydtdte10.);

        Salary=input(AnnualSalary, dollar12.);

run;


*Solution;

data emp_new;

         set cr.employee_new(rename=(HireDate=HireDateC));

         EmpID=substr(EmpID,4);

         HireDate=input(HireDateC, anydtdte10.);

         Salary=input(AnnualSalary, dollar10.);

         drop HireDateC;

run;
```

Table: WORK.EMP_NEW | View: Column names | Filter: (none)

Columns

- Select all
- Country
- EmpID
- Name
- City
- Job_Title
- Department
- Manager_ID
- AnnualSalary
- HireDate
- Salary

Total rows: 12 Total columns: 10 Rows 1-12

Country	EmpID	Name	City	Job_Title	Department	Manager_ID	AnnualSalary	HireDate	Salary
AU	121150	Hornsby, Philip	Sydney	Sales Manager	Sales Management	120101	$151,100	18642	151100
US	121151	Melcham, Rhonda	Philadelphia	Pricing Specialist	Logistics Management	120663	$94,900	18643	94900
US	121152	Davis, Keith	Philadelphia	Warehouse Assistant III	Stock & Shipping	120679	$32,600	18643	32600
AU	121153	Pierce, Melonie	Melbourne	Sales Rep. II	Sales	120102	$97,000	18646	97000
AU	121154	Brainard, Barbara	Sydney	Sales Rep. II	Sales	120102	$88,000	18646	88000
US	121155	Andrews, Melvin	Philadelphia	Applications Developer IV	IS	120798	$77,400	18647	77400
US	121156	MacKenzie, Scott	Philadelphia	Temp. Sales Rep.	Sales	121145	$45,000	18647	45000
AU	121157	Priestley, Kevin	Melbourne	Sales Rep. III	Sales	120103	$103,200	18649	103200
AU	121158	Jones, Meredith	Melbourne	Sales Rep. III	Sales	120102	$109,700	18650	109700
US	121159	Mangiano, Lena	Philadelphia	Temp. Sales Rep.	Sales	121145	$38,200	18653	38200
AU	121160	Woodward, Patricl	Sydney	Temp. Sales Rep.	Sales	120103	$42,000	18653	42000
US	121161	Gray, Philicia	San Diego	Trainee	Sales	121145	$32,000	18655	32000

```
proc format;

        value shiprange 0="Same day"

                        1-3="1-3 days"

                        4-7="4-7 days"

                        8-high="8+ days"
```

```
                                        .="Unknown";
run;


data profit2;
        set cr.profit;
        format ShipDays shiprange.;
run;
```

Table: WORK.PROFIT2 | View: Column names | Filter: (none)

Columns: Select all, Order_ID, Order_Date, Delivery_Date, Product_ID, Product_Line

Total rows: 10781 Total columns: 13 — Rows 1-100

oduct_Line	Product_Category	Quantity	Customer_Country	Customer_Continent	Order_Source	Profit	ShipDays	Age_Range
orts	Assorted Sports Articles	2	BE	Europe	Retail	$414.60	Same day	46-60
utdoors	Outdoors	3	RU	Asia	Internet	$946.80	4-7 days	31-45
utdoors	Outdoors	2	NL	Europe	Retail	$322.80	4-7 days	15-30
othes & Shoes	Shoes	2	NL	Europe	Phone	$216.80	4-7 days	15-30
orts	Running - Jogging	.	IT	Europe	Retail	.	Same day	61-75
othes & Shoes	Shoes	2	IT	Europe	Retail	$79.60	Same day	31-45

```
data profit2;
        set cr.profit;
        ShipRange=put(ShipDays, shiprange.);
run;
```

Table: WORK.PROFIT2 | View: Column names | Filter: (none)

Columns: Select all, Order_ID, Order_Date, Delivery_Date, Product_ID, Product_Line

Total rows: 10781 Total columns: 14 — Rows 1-100

Product_Category	Quantity	Customer_Country	Customer_Continent	Order_Source	Profit	ShipDays	Age_Range	ShipRange
Assorted Sports Articles	2	BE	Europe	Retail	$414.60	0	46-60	Same day
Outdoors	3	RU	Asia	Internet	$946.80	4	31-45	4-7 days
Outdoors	2	NL	Europe	Retail	$322.80	4	15-30	4-7 days
Shoes	2	NL	Europe	Phone	$216.80	4	15-30	4-7 days
Running - Jogging	.	IT	Europe	Retail	.	0	61-75	Same day
Shoes	2	IT	Europe	Retail	$79.60	0	31-45	Same day

```
proc freq data=cr.profit;
        table ShipDays;
        format ShipDays shiprange.;
run;
```

The FREQ Procedure

ShipDays	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Same day	7828	72.61	7828	72.61
1-3 days	1492	13.84	9320	86.45
4-7 days	1318	12.23	10638	98.67
8+ days	143	1.33	10781	100.00

```
data country;
```

```
        set cr.country_clean;
        FmtName="$ctryfmt";
        Start=Country_Key;
        Label=Country_Name;
        keep FmtName Start Label;
run;

proc format cntlin=country;
run;

data profit2;
        set cr.profit;
        ShipRange=put(ShipDays, shiprange.);
        format Customer_Country ctryfmt.;
run;
```

Table: WORK.PROFIT2 | View: Column names | Filter: (none)

Columns: Select all, Order_ID, Order_Date, Delivery_Date, Product_ID, Product_Line, Product_Category

Total rows: 10781 Total columns: 14 — Rows 1-100

Product_Category	Quantity	Customer_Country	Customer_Continent	Order_Source	Profit	ShipDays	Age_Range	ShipRange
Assorted Sports Articles	2	Belgium	Europe	Retail	$414.60	0	46-60	Same day
Outdoors	3	Russia	Asia	Internet	$946.80	4	31-45	4-7 days
Outdoors	2	Netherlands	Europe	Retail	$322.80	4	15-30	4-7 days
Shoes	2	Netherlands	Europe	Phone	$216.80	4	15-30	4-7 days
Running - Jogging	.	Italy	Europe	Retail	.	0	61-75	Same day
Shoes	2	Italy	Europe	Retail	$79.60	0	31-45	Same day

/* Programming Question 4.10

If necessary, start SAS Studio. Write a new program to do the following:

Create a numeric format named BMIRANGE that assigns the following ranges and labels.

Range	Label
less than 18.5	Underweight
18.5 to 24.9	Normal
25 to 29.9	Overweight
30 or more	Obese

Create a frequency report based on the BMI column in the sashelp.bmimen table.

Include rows where Age is greater than or equal to 21. Apply the BMIRANGE format to the BMI column.

What is the frequency count of men in the Overweight BMI range?

What is the frequency count of men in the Underweight BMI range?

```
*/
proc format;
        value bmirange low-<18.5="Underweight"
                                18.5-24.9="Normal"
                                25-29.9="Overweight"
                                30-high="Obese"
                                .="Unknown";
run;

proc freq data=sashelp.bmimen;
        where Age>=21;
        table bmi;
        format bmi bmirange.;
run;
```

The FREQ Procedure

BMI	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Underweight	14	1.20	14	1.20
Normal	318	27.25	332	28.45
Overweight	521	44.64	853	73.09
Obese	314	26.91	1167	100.00

/*Programming Question 4.11

If necessary, start SAS Studio and submit libname.sas. Write a new program to do the following:

Use the cr.continent_codes table to create a numeric format named CONTFMT.

The Code column contains data values, and the Continent column contains labels.

Read the cr.demographics table and compute the sum of the Pop column for each value of Cont.

Apply the CONTFMT format to the Cont column. Note: If you don't have the cr.demograhics table, copy and paste this code into a new program window and run it to create the table.

How many countries are in the European continent?

What is the total population in Africa? Note: Type only numbers in your answer.

```
*/
data continent;
        set cr.continent_codes;
        FmtName="contfmt";
        Start=Code;
        Label=Continent;
        keep FmtName Start Label;
run;

proc format cntlin=continent;
run;

proc means data=cr.demographics sum;
        var Pop;
        class Cont;
        format Pop comma15. Cont contfmt.;
run;



*Solution;
data continentfmt;
   set cr.continent_codes;
   retain fmtname "contfmt";
   Start=Code;
   Label=Continent;
run;

proc format cntlin=continentfmt;
```

```
run;

proc means data=cr.demographics sum maxdec=0;
   var pop;
   class cont;
   format cont contfmt.;
run;
```

The MEANS Procedure

Analysis Variable : Pop Population (2005)		
Numeric Rep. for Continent	N Obs	Sum
North America	16	509218938
South America	19	377115072
Europe	46	779756261
Africa	53	904804386
Asia	45	4029012970
Oceania	18	72540243.00

```
/* Programming Question 4.12
If necessary, start SAS Studio and submit libname.sas. Open p204q3.sas from the programs folder.
Fix the program to ensure that the following actions occur:
Create custom formats for the values of Status (M, S, and O) and for ranges of Salary.
Apply the custom formats in the PROC FREQ step.
How many employees are in the Other category?
How many employees in San Diego make between $50K and $100K?
*/
proc format;
        value $statfmt S="Single"
                M="Married"
                O="Other";
        value salrange low-<50000="Under $50K"
                 50000-100000="50K-100K"
                 100000<-high="Over 100K";
```

```
run;

proc freq data=cr.employee;
        tables Status;
        tables City*Salary / nopercent nocol;
        format Status $statfmt. Salary salrange.;
run;
```

The FREQ Procedure

Status	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Married	215	50.71	215	50.71
Other	40	9.43	255	60.14
Single	169	39.86	424	100.00

Frequency
Row Pct

Table of City by Salary

City(City)	Salary			
	Under $50K	50K-100K	Over 100K	Total
Melbourne	40 97.56	0 0.00	1 2.44	41
Miami-Dade	70 64.22	33 30.28	6 5.50	109
Philadelphia	67 70.53	24 25.26	4 4.21	95
San Diego	87 77.68	20 17.86	5 4.46	112
Sydney	62 92.54	3 4.48	2 2.99	67
Total	326	80	18	424