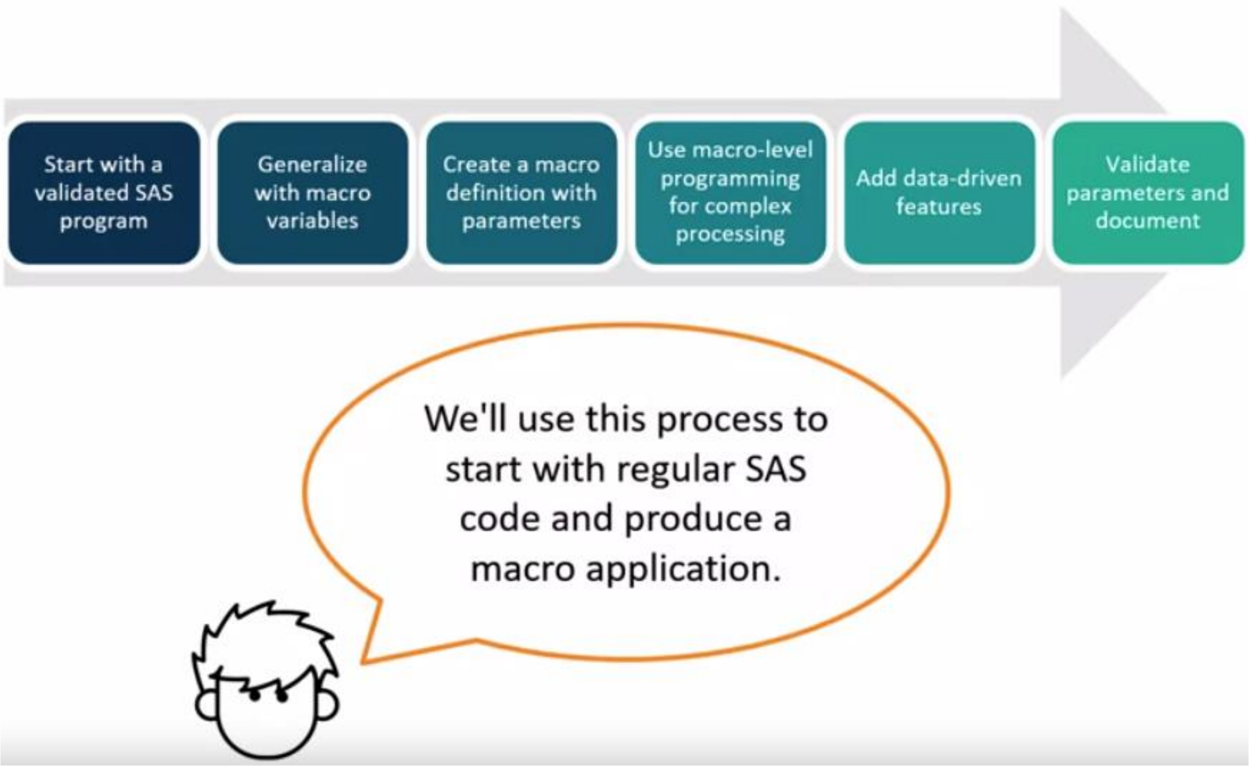
SAS Advanced Programmer

SAP2 SAS Macro

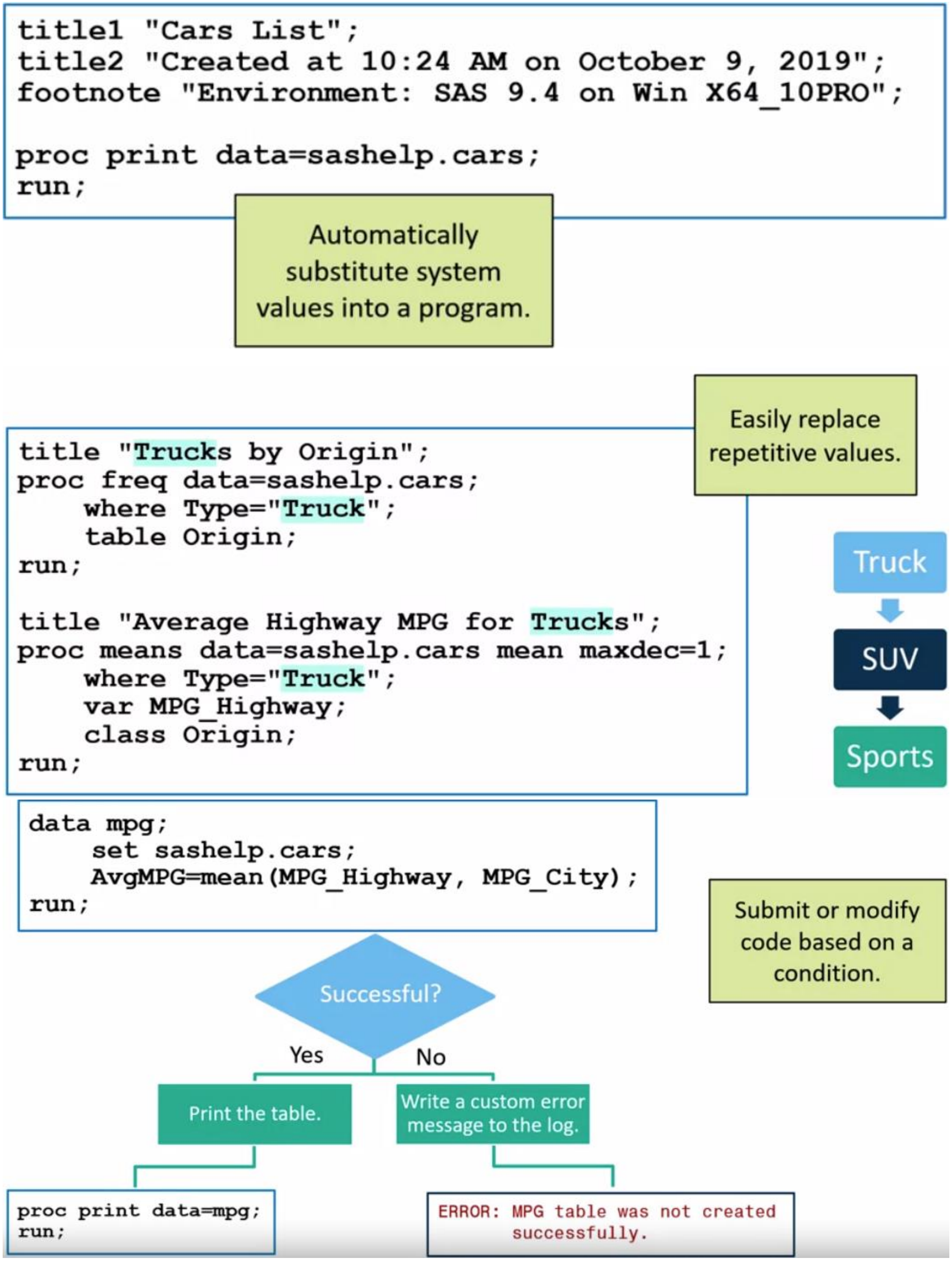
SAP201-202 SAS Program Flow, Creating and Using Macro Variables

SAP203 Macro Functions, Using SQL to Create Macro Variables, Using the DATA Step to Create Macro Variables, Indirect References to Macro Variables

```
%let path=~/EMC1V2;

libname mc1 "&path/data";
```



Start with a validated SAS program
Generalize with macro variables
Create a macro definition with parameters
Use macro-level programming for complex processing
Add data-driven features
Validate parameters and document
We'll use this process to start with regular SAS code and produce a macro application.



```
title1 "Cars List";
title2 "Created at 10:24 AM on October 9, 2019";
footnote "Environment: SAS 9.4 on Win X64_10PRO";

proc print data=sashelp.cars;
run;
```

Automatically substitute system values into a program.

Easily replace repetitive values.

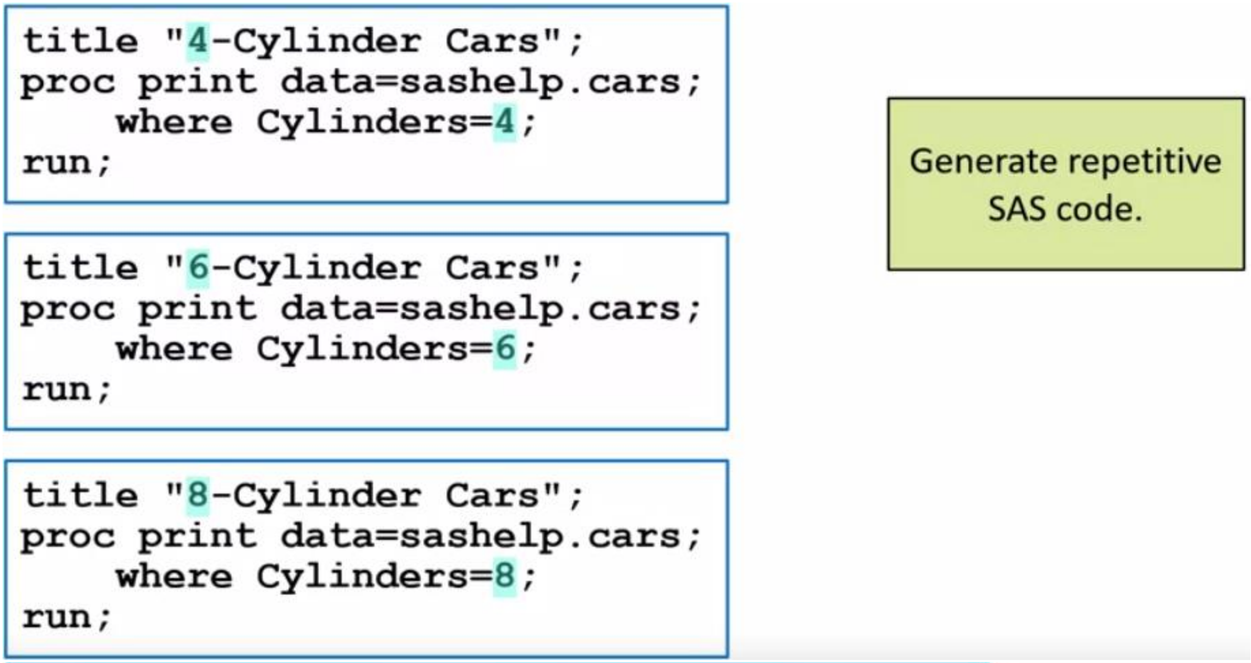
```
title "Trucks by Origin";
proc freq data=sashelp.cars;
    where Type="Truck";
    table Origin;
run;

title "Average Highway MPG for Trucks";
proc means data=sashelp.cars mean maxdec=1;
    where Type="Truck";
    var MPG_Highway;
    class Origin;
run;
```

```
data mpg;
    set sashelp.cars;
    AvgMPG=mean(MPG_Highway, MPG_City);
run;
```

Submit or modify code based on a condition.

```
proc print data=mpg;
run;
```

```
ERROR: MPG table was not created
       successfully.
```

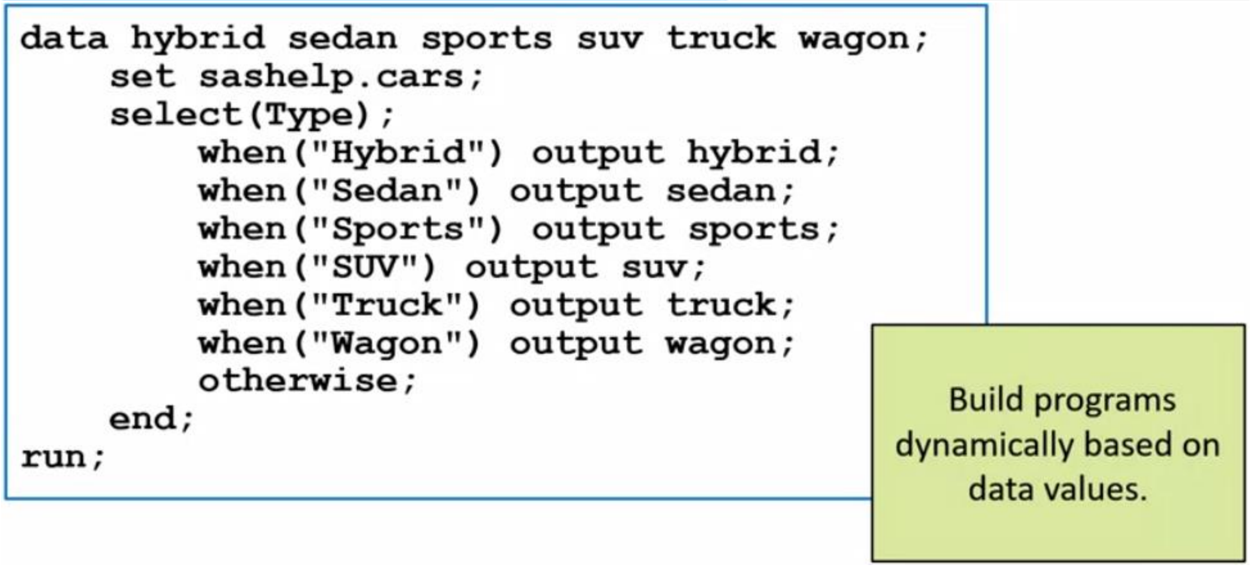


```
title "4-Cylinder Cars";
proc print data=sashelp.cars;
    where Cylinders=4;
run;
```

```
title "6-Cylinder Cars";
proc print data=sashelp.cars;
    where Cylinders=6;
run;
```

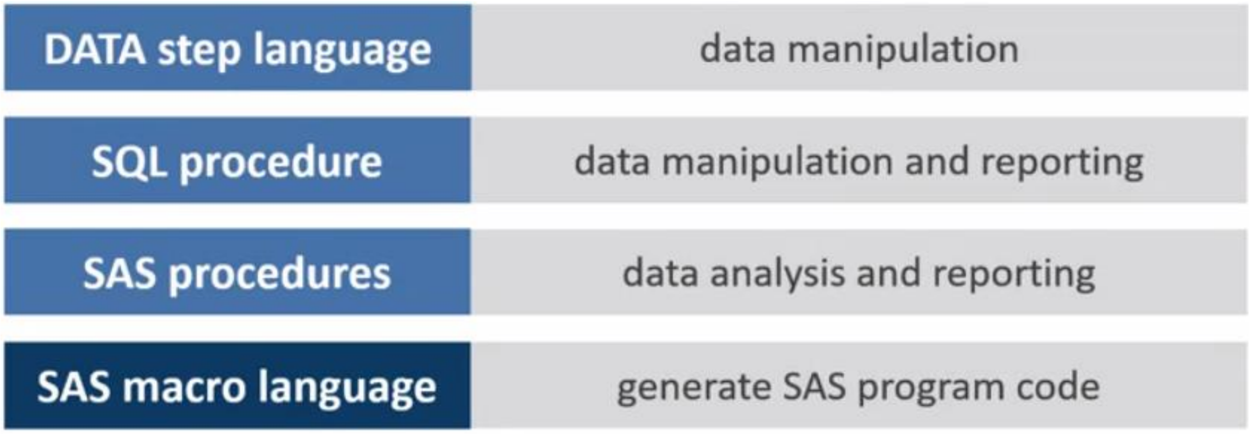
```
title "8-Cylinder Cars";
proc print data=sashelp.cars;
    where Cylinders=8;
run;
```

Generate repetitive SAS code.

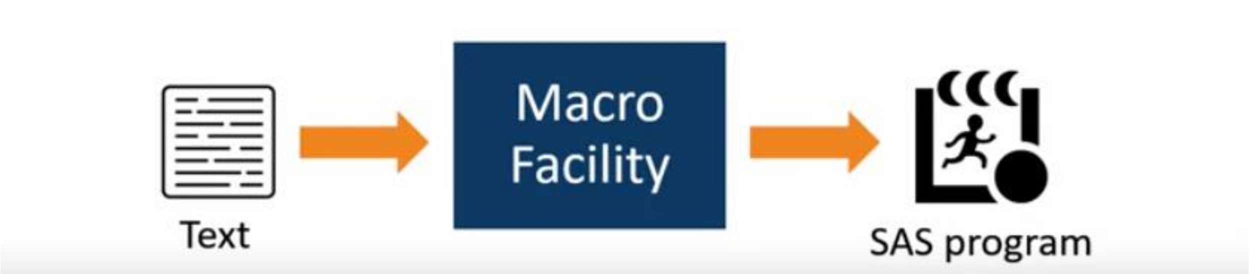


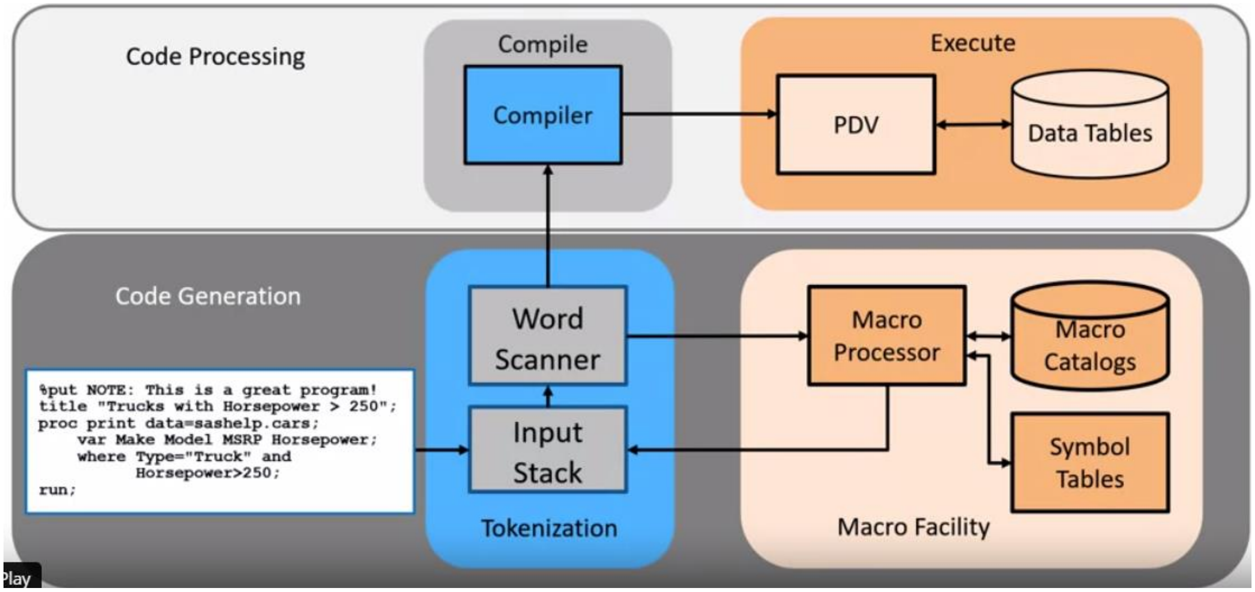
```
data hybrid sedan sports suv truck wagon;
    set sashelp.cars;
    select(Type);
        when("Hybrid") output hybrid;
        when("Sedan") output sedan;
        when("Sports") output sports;
        when("SUV") output suv;
        when("Truck") output truck;
        when("Wagon") output wagon;
        otherwise;
    end;
run;
```

Build programs dynamically based on data values.

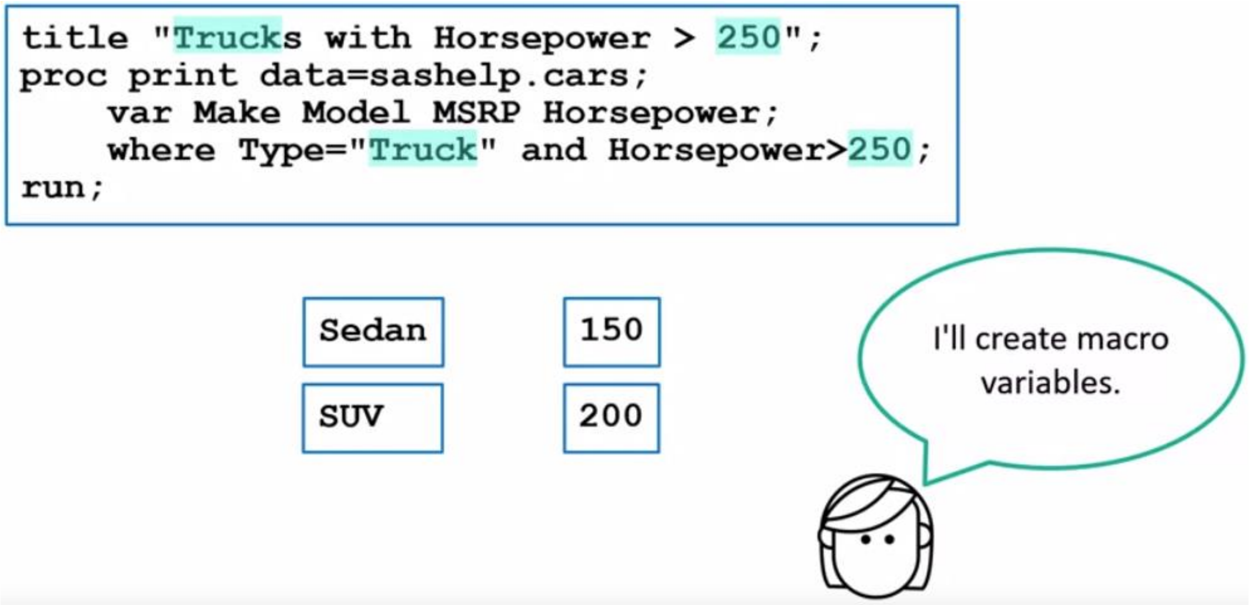


DATA step language	data manipulation
SQL procedure	data manipulation and reporting
SAS procedures	data analysis and reporting
SAS macro language	generate SAS program code





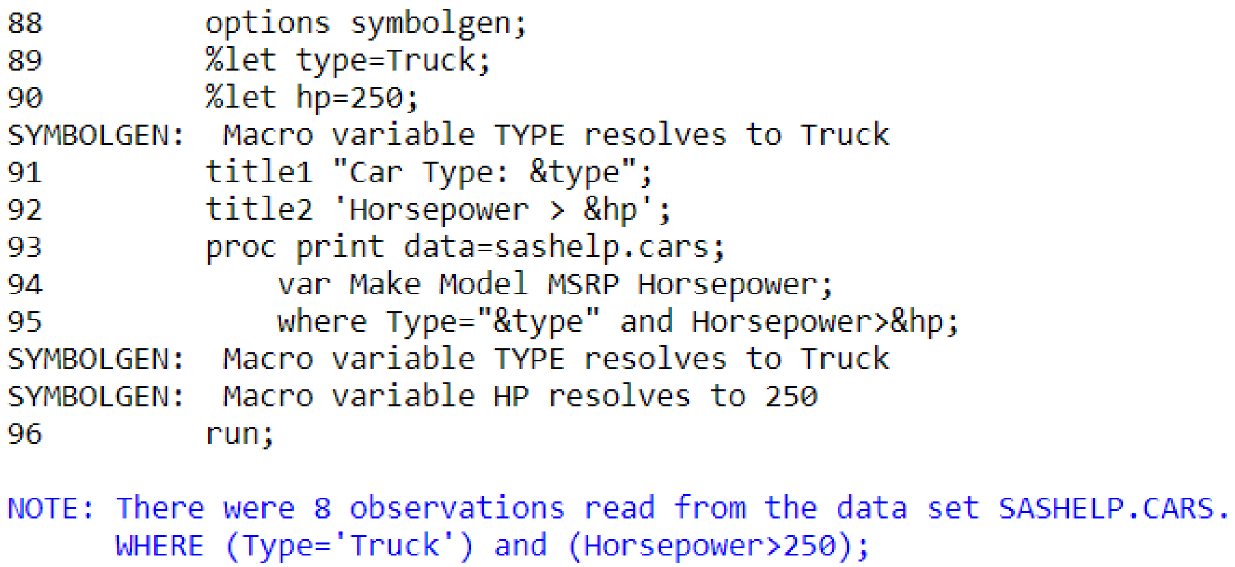
```
%put NOTE: This is a great program!
title "Trucks with Horsepower > 250";
proc print data=sashelp.cars;
    var Make Model MSRP Horsepower;
    where Type="Truck" and
          Horsepower>250;
run;
```



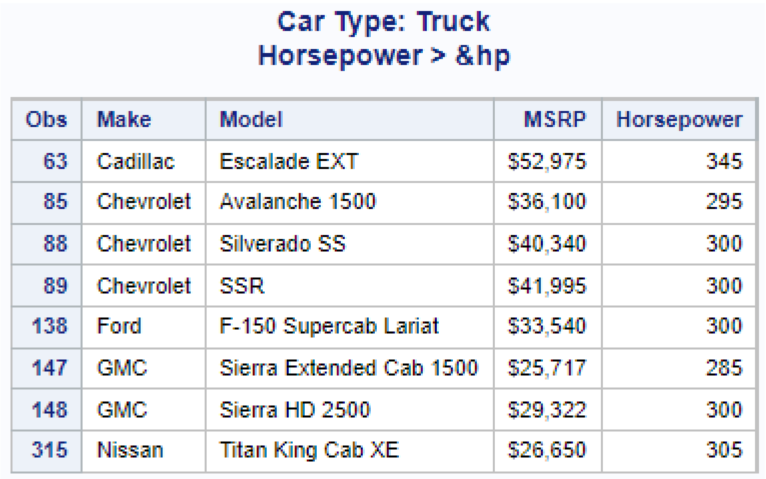
```
title "Trucks with Horsepower > 250";
proc print data=sashelp.cars;
    var Make Model MSRP Horsepower;
    where Type="Truck" and Horsepower>250;
run;
```

```
**********************************************************;
*  Activity 2.04                                   *;
*  1) Notice that the program includes two TITLE          *;
*     statements, each referencing a macro variable.      *;
*  2) At the top of the program, turn on the SYMBOLGEN    *;
*     option. At the bottom of the program, turn off      *;
*     SYMBOLGEN.                                      *;
*  3) Run the program and review the log and results.     *;
*     What is printed as the second title?                *;
*  4) In the TITLE2 statement, change the single          *;
*     quotation marks to double quotation marks and run   *;
*     the program again. How do the results and the log   *;
*     differ?                                         *;
**********************************************************;

options symbolgen;
%let type=Truck;
%let hp=250;
```

```
title1 "Car Type: &type";

title2 'Horsepower > &hp';

proc print data=sashelp.cars;

   var Make Model MSRP Horsepower;

   where Type="&type" and Horsepower>&hp;

run;

title;

options nosymbolgen;
```



```
88          options symbolgen;
89          %let type=Truck;
90          %let hp=250;
SYMBOLGEN:  Macro variable TYPE resolves to Truck
91          title1 "Car Type: &type";
92          title2 'Horsepower > &hp';
93          proc print data=sashelp.cars;
94              var Make Model MSRP Horsepower;
95              where Type="&type" and Horsepower>&hp;
SYMBOLGEN:  Macro variable TYPE resolves to Truck
SYMBOLGEN:  Macro variable HP resolves to 250
96          run;

NOTE: There were 8 observations read from the data set SASHELP.CARS.
      WHERE (Type='Truck') and (Horsepower>250);
```

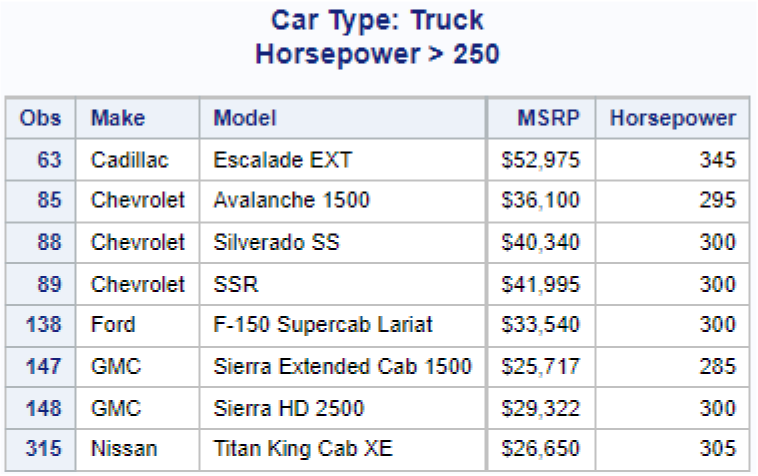


Car Type: Truck
Horsepower > &hp

Obs	Make	Model	MSRP	Horsepower
63	Cadillac	Escalade EXT	$52,975	345
85	Chevrolet	Avalanche 1500	$36,100	295
88	Chevrolet	Silverado SS	$40,340	300
89	Chevrolet	SSR	$41,995	300
138	Ford	F-150 Supercab Lariat	$33,540	300
147	GMC	Sierra Extended Cab 1500	$25,717	285
148	GMC	Sierra HD 2500	$29,322	300
315	Nissan	Titan King Cab XE	$26,650	305

```
options symbolgen;

%let type=Truck;

%let hp=250;
```

```
title1 "Car Type: &type";

title2 "Horsepower > &hp";

proc print data=sashelp.cars;

   var Make Model MSRP Horsepower;

   where Type="&type" and Horsepower>&hp;

run;

title;

options nosymbolgen;
```

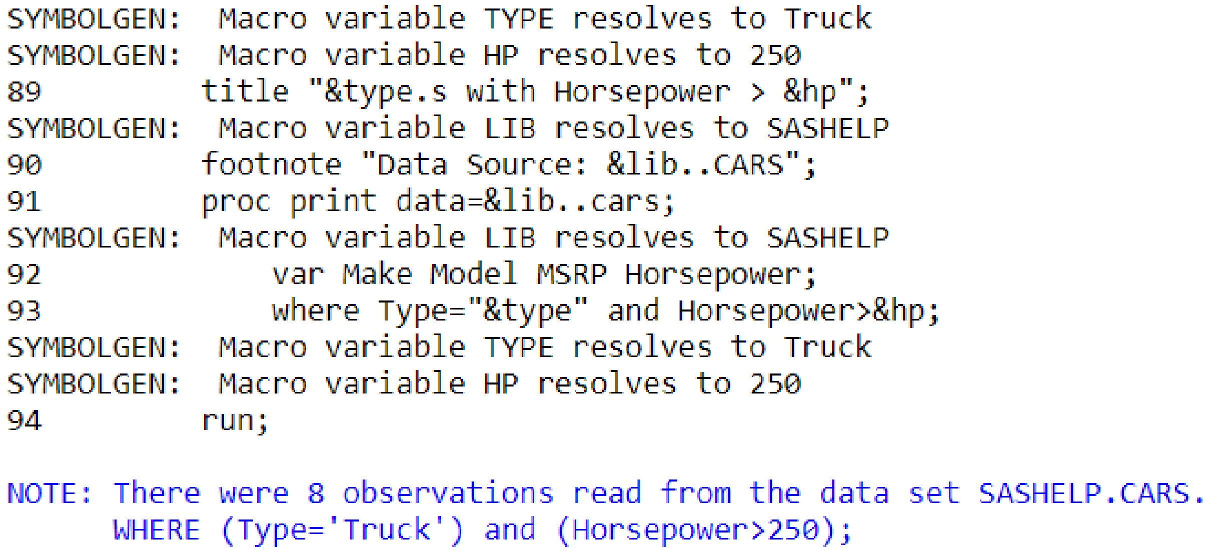


Car Type: Truck
Horsepower > 250

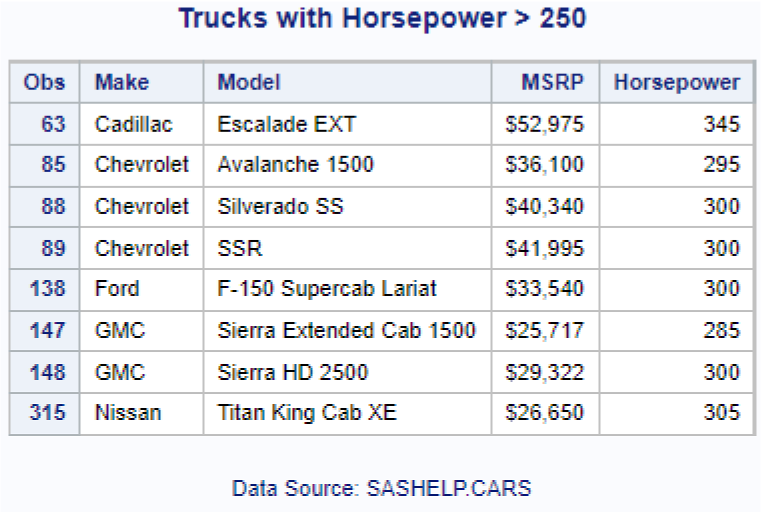
Obs	Make	Model	MSRP	Horsepower
63	Cadillac	Escalade EXT	$52,975	345
85	Chevrolet	Avalanche 1500	$36,100	295
88	Chevrolet	Silverado SS	$40,340	300
89	Chevrolet	SSR	$41,995	300
138	Ford	F-150 Supercab Lariat	$33,540	300
147	GMC	Sierra Extended Cab 1500	$25,717	285
148	GMC	Sierra HD 2500	$29,322	300
315	Nissan	Titan King Cab XE	$26,650	305

```
*******************************************************;

*  Activity 2.05                              *;

*  1) Modify the TITLE statement reference to &type by    *;

*     adding a period before the 's'.                  *;

*  2) Replace the hardcoded text sashelp in the FOOTNOTE  *;

*     and PROC statements with a reference to the Lib     *;

*     macro variable (&lib).                          *;

*  3) Run the program and examine the log and the error   *;

*     statements. Why did the program fail to run?        *;

*******************************************************;


options symbolgen;
```

```
%let type=Truck;

%let hp=250;

%let lib=SASHELP;


title "&type.s with Horsepower > &hp";

footnote "Data Source: &lib..CARS";

proc print data=&lib..cars;

   var Make Model MSRP Horsepower;

   where Type="&type" and Horsepower>&hp;

run;

title;footnote;

options nosymbolgen;
```



```
SYMBOLGEN:  Macro variable TYPE resolves to Truck
SYMBOLGEN:  Macro variable HP resolves to 250
89         title "&type.s with Horsepower > &hp";
SYMBOLGEN:  Macro variable LIB resolves to SASHELP
90         footnote "Data Source: &lib..CARS";
91         proc print data=&lib..cars;
SYMBOLGEN:  Macro variable LIB resolves to SASHELP
92            var Make Model MSRP Horsepower;
93            where Type="&type" and Horsepower>&hp;
SYMBOLGEN:  Macro variable TYPE resolves to Truck
SYMBOLGEN:  Macro variable HP resolves to 250
94         run;

NOTE: There were 8 observations read from the data set SASHELP.CARS.
      WHERE (Type='Truck') and (Horsepower>250);
```



Trucks with Horsepower > 250

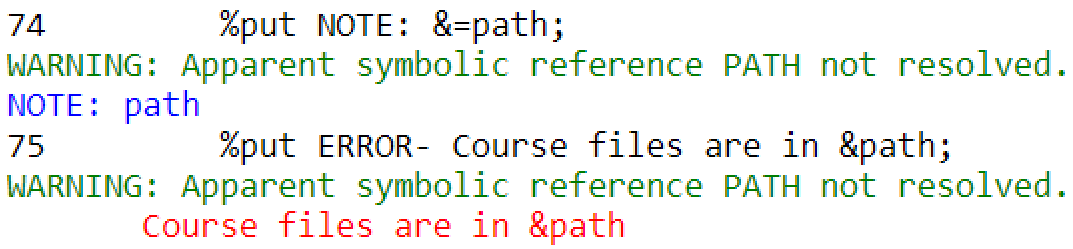
Obs	Make	Model	MSRP	Horsepower
63	Cadillac	Escalade EXT	$52,975	345
85	Chevrolet	Avalanche 1500	$36,100	295
88	Chevrolet	Silverado SS	$40,340	300
89	Chevrolet	SSR	$41,995	300
138	Ford	F-150 Supercab Lariat	$33,540	300
147	GMC	Sierra Extended Cab 1500	$25,717	285
148	GMC	Sierra HD 2500	$29,322	300
315	Nissan	Titan King Cab XE	$26,650	305

Data Source: SASHELP.CARS

%put _all_;

%put NOTE: &=path;

%put ERROR- Course files are in &path;



```
74          %put NOTE: &=path;
WARNING: Apparent symbolic reference PATH not resolved.
NOTE: path
75          %put ERROR- Course files are in &path;
WARNING: Apparent symbolic reference PATH not resolved.
       Course files are in &path
```

***;

* Activity 2.07   *;

* 1) Review the program and notice that the DATA step   *;

*   creates a table named Avg_MPG. Highlight the DATA   *;

*   step and %PUT statement, and run the selected code. *;

*   Review the log to see all automatic macro variables *;

*   stored in the global symbol table.   *;

* 2) Identify the macro variables that store the date   *;

*   and the last table created.   *;

* 3) Use macro variable references in the TITLE2 and   *;

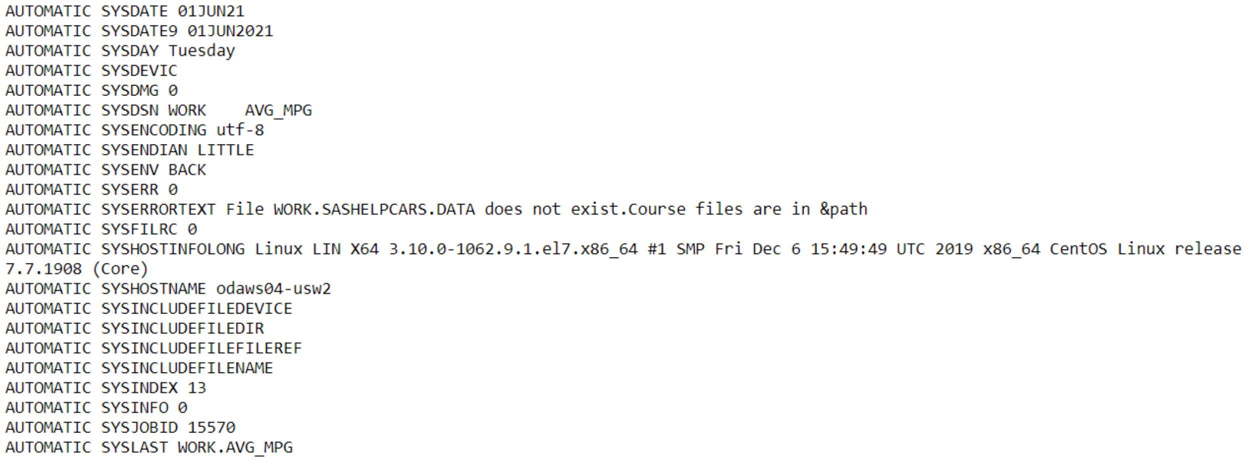
*   FOOTNOTE statements to insert the table name and   *;

*   date into the program.   *;

```
***********************************************************;
```

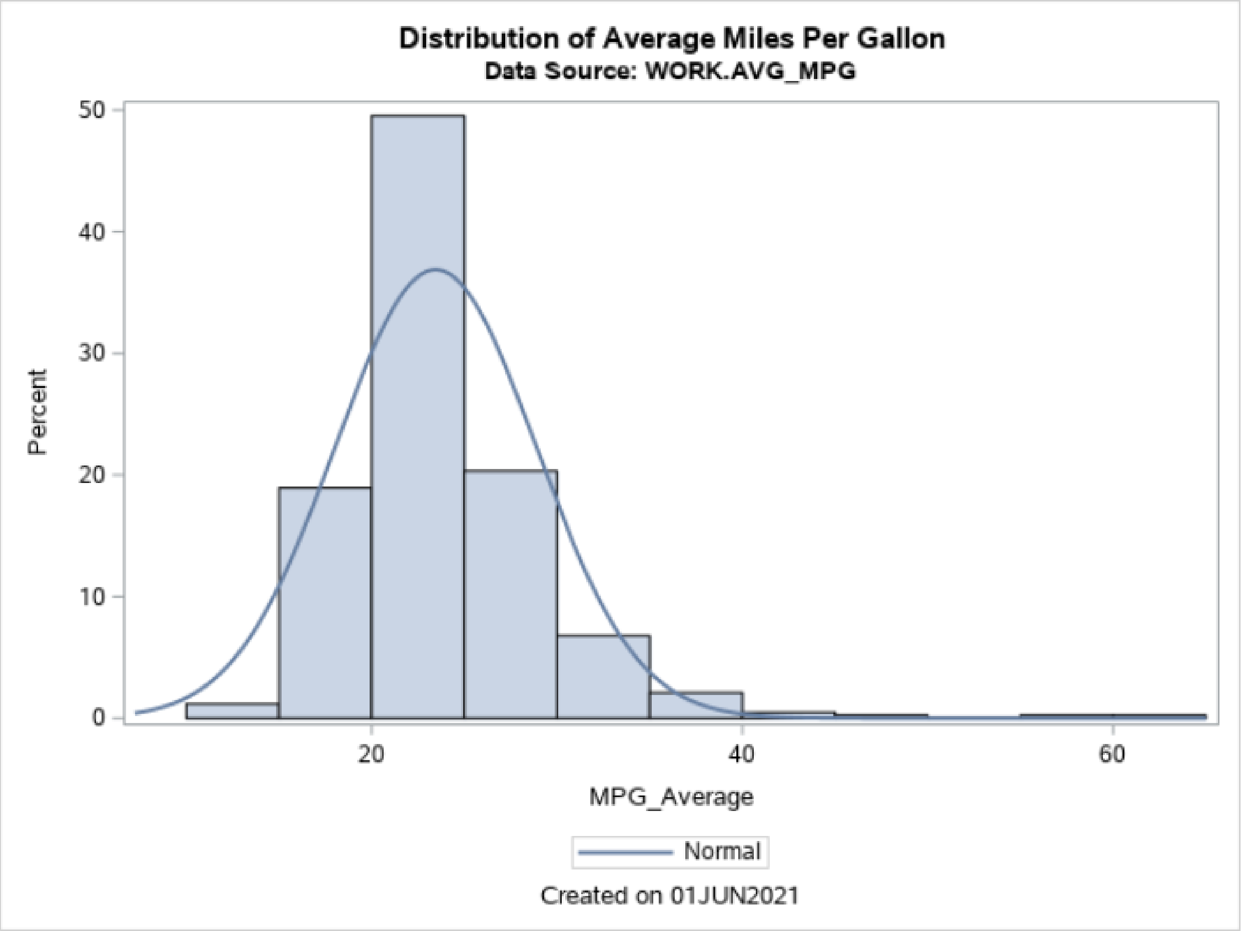
```
data Avg_MPG;

   set sashelp.cars;

   MPG_Average=mean(MPG_City, MPG_Highway);

run;
```

```
%put _automatic_;
```



```
AUTOMATIC SYSDATE 01JUN21
AUTOMATIC SYSDATE9 01JUN2021
AUTOMATIC SYSDAY Tuesday
AUTOMATIC SYSDEVIC
AUTOMATIC SYSDMG 0
AUTOMATIC SYSDSN WORK    AVG_MPG
AUTOMATIC SYSENCODING utf-8
AUTOMATIC SYSENDIAN LITTLE
AUTOMATIC SYSENV BACK
AUTOMATIC SYSERR 0
AUTOMATIC SYSERRORTEXT File WORK.SASHELPCARS.DATA does not exist.Course files are in &path
AUTOMATIC SYSFILRC 0
AUTOMATIC SYSHOSTINFOLONG Linux LIN X64 3.10.0-1062.9.1.el7.x86_64 #1 SMP Fri Dec 6 15:49:49 UTC 2019 x86_64 CentOS Linux release
7.7.1908 (Core)
AUTOMATIC SYSHOSTNAME odaws04-usw2
AUTOMATIC SYSINCLUDEFILEDEVICE
AUTOMATIC SYSINCLUDEFILEDIR
AUTOMATIC SYSINCLUDEFILEFILEREF
AUTOMATIC SYSINCLUDEFILENAME
AUTOMATIC SYSINDEX 13
AUTOMATIC SYSINFO 0
AUTOMATIC SYSJOBID 15570
AUTOMATIC SYSLAST WORK.AVG_MPG
```

```
title1 "Distribution of Average Miles Per Gallon";

title2 "Data Source: &syslast";

footnote "Created on &sysdate9";

proc sgplot;

   histogram MPG_Average;

   density MPG_Average;

run;

title;footnote;
```



/**

* Creating and Using Macro Variables: Practice #1 *

**/

/*Level 1 Practice: Defining and Using Macro Variables for Substitution

Reminder: If you restarted your SAS session,

you must submit the libname.sas program in the EMC1V2 folder to access your practice files

Open the m102p01.sas program from the practices folder.

Submit the program and review the log and output.

Verify that the title is US Customers Ages 18 to 24 and that 127 rows were read.

Create a macro variable named Country and assign the value US. Replace all occurrences of US with references to Country.

Submit the program and verify that the results are the same as step 1.

Modify the value of Country to FR. Resubmit the program.

How many rows were read?

Modify the program as follows:

Create additional macro variables, Age1 and Age2.

Set Age1 to 25, Age2 to 34, and Country to AU.

Replace all occurrences of 18 and 24 with references to Age1 and Age2.

Submit the program.

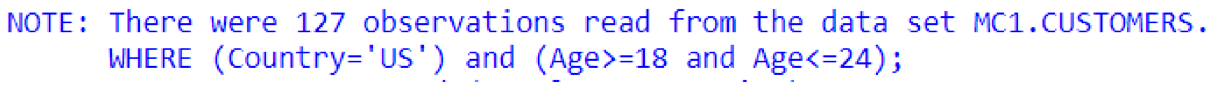
What is the report title?

How many rows were read?

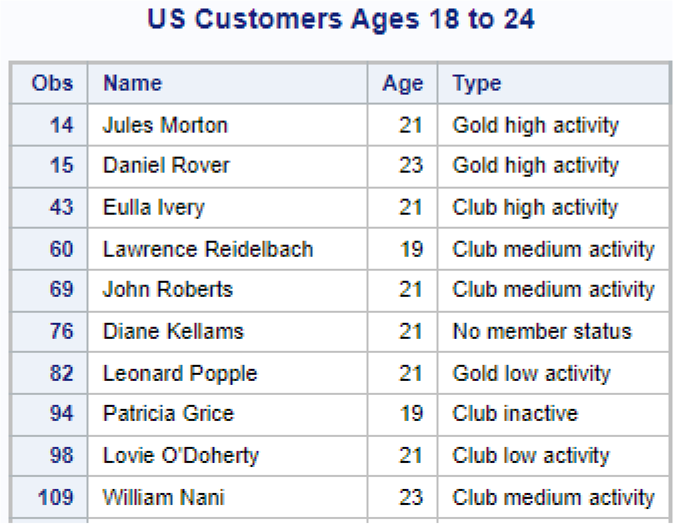
Note: Enter a numeric value for your answer.

*/

```
title 'US Customers Ages 18 to 24';
proc print data=mc1.customers;
   var Name Age Type;
   where Country = 'US'
      and Age between 18 and 24;
run;
title;
```



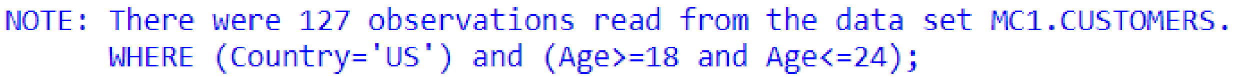
```
NOTE: There were 127 observations read from the data set MC1.CUSTOMERS.
      WHERE (Country='US') and (Age>=18 and Age<=24);
```



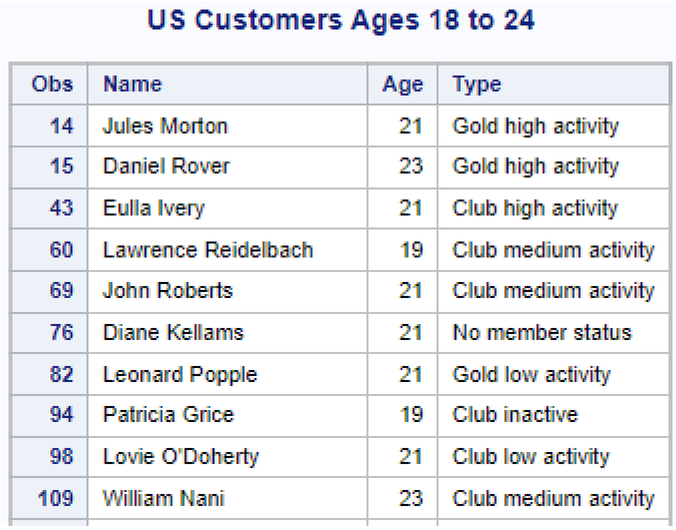
US Customers Ages 18 to 24

Obs	Name	Age	Type
14	Jules Morton	21	Gold high activity
15	Daniel Rover	23	Gold high activity
43	Eulla Ivery	21	Club high activity
60	Lawrence Reidelbach	19	Club medium activity
69	John Roberts	21	Club medium activity
76	Diane Kellams	21	No member status
82	Leonard Popple	21	Gold low activity
94	Patricia Grice	19	Club inactive
98	Lovie O'Doherty	21	Club low activity
109	William Nani	23	Club medium activity

```
%let Country=US;

title "&Country Customers Ages 18 to 24";

proc print data=mc1.customers;

   var Name Age Type;

   where Country = "&Country"

      and Age between 18 and 24;

run;

title;
```



```
NOTE: There were 127 observations read from the data set MC1.CUSTOMERS.
      WHERE (Country='US') and (Age>=18 and Age<=24);
```

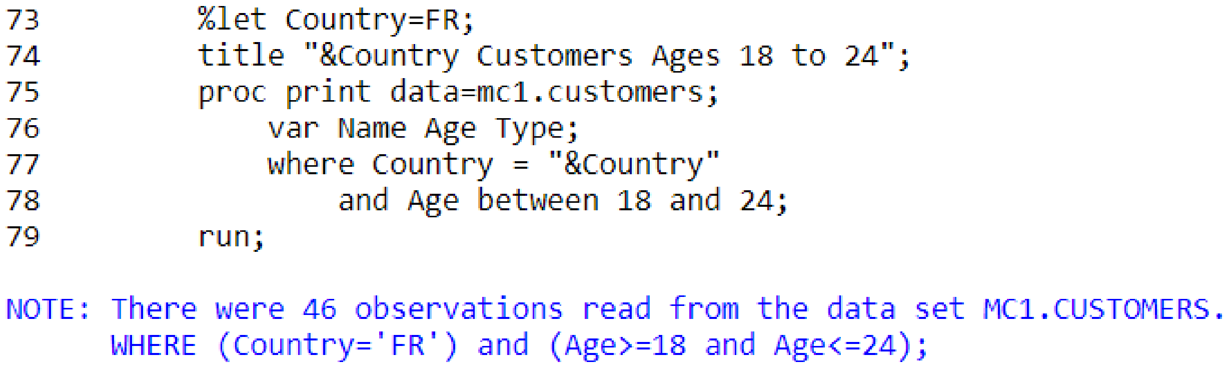


US Customers Ages 18 to 24

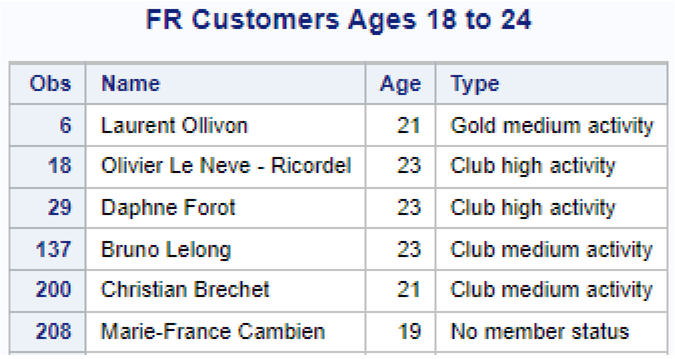
Obs	Name	Age	Type
14	Jules Morton	21	Gold high activity
15	Daniel Rover	23	Gold high activity
43	Eulla Ivery	21	Club high activity
60	Lawrence Reidelbach	19	Club medium activity
69	John Roberts	21	Club medium activity
76	Diane Kellams	21	No member status
82	Leonard Popple	21	Gold low activity
94	Patricia Grice	19	Club inactive
98	Lovie O'Doherty	21	Club low activity
109	William Nani	23	Club medium activity

```
%let Country=FR;
```

```
title "&Country Customers Ages 18 to 24";

proc print data=mc1.customers;

   var Name Age Type;

   where Country = "&Country"

      and Age between 18 and 24;

run;

title;
```



```
%let Country=FR;
title "&Country Customers Ages 18 to 24";
proc print data=mc1.customers;
    var Name Age Type;
    where Country = "&Country"
        and Age between 18 and 24;
run;

NOTE: There were 46 observations read from the data set MC1.CUSTOMERS.
      WHERE (Country='FR') and (Age>=18 and Age<=24);
```

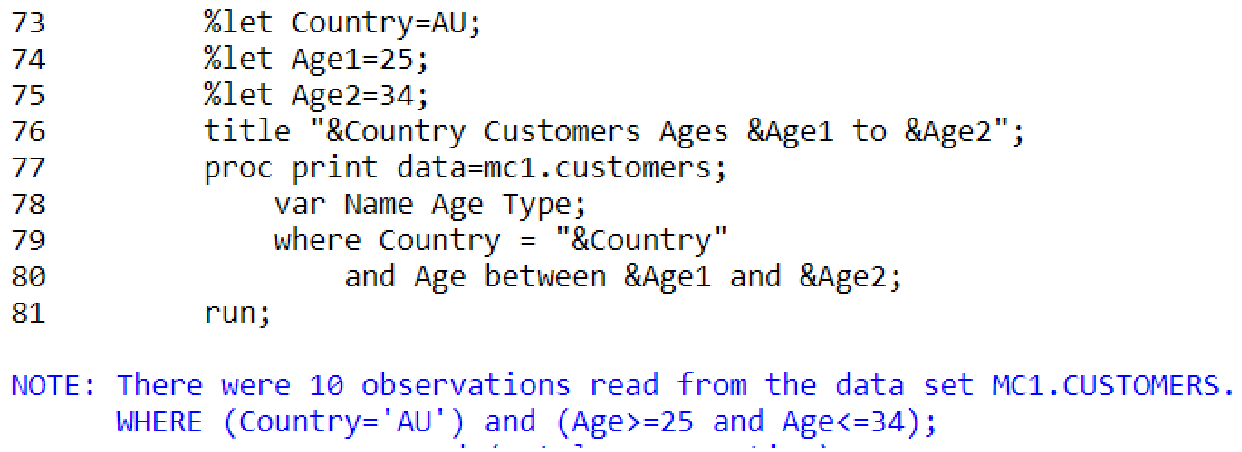


FR Customers Ages 18 to 24

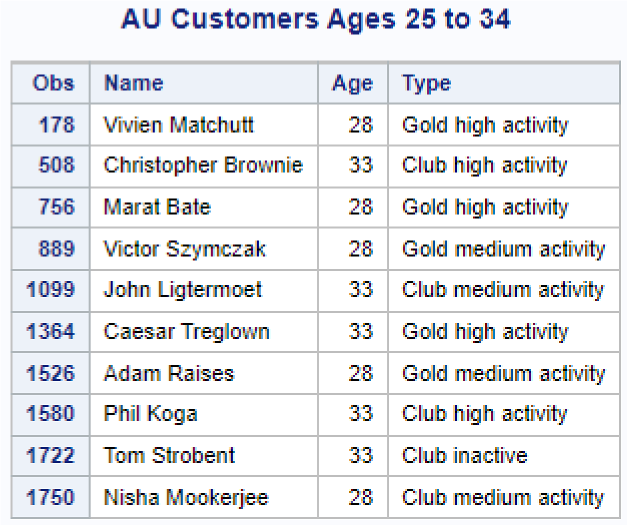
Obs	Name	Age	Type
6	Laurent Ollivon	21	Gold medium activity
18	Olivier Le Neve - Ricordel	23	Club high activity
29	Daphne Forot	23	Club high activity
137	Bruno Lelong	23	Club medium activity
200	Christian Brechet	21	Club medium activity
208	Marie-France Cambien	19	No member status

```
%let Country=AU;

%let Age1=25;

%let Age2=34;

title "&Country Customers Ages &Age1 to &Age2";

proc print data=mc1.customers;

   var Name Age Type;

   where Country = "&Country"

      and Age between &Age1 and &Age2;

run;
```

title;



```
%let Country=AU;
%let Age1=25;
%let Age2=34;
title "&Country Customers Ages &Age1 to &Age2";
proc print data=mc1.customers;
    var Name Age Type;
    where Country = "&Country"
        and Age between &Age1 and &Age2;
run;

NOTE: There were 10 observations read from the data set MC1.CUSTOMERS.
      WHERE (Country='AU') and (Age>=25 and Age<=34);
```



AU Customers Ages 25 to 34

Obs	Name	Age	Type
178	Vivien Matchutt	28	Gold high activity
508	Christopher Brownie	33	Club high activity
756	Marat Bate	28	Gold high activity
889	Victor Szymczak	28	Gold medium activity
1099	John Ligtermoet	33	Club medium activity
1364	Caesar Treglown	33	Gold high activity
1526	Adam Raises	28	Gold medium activity
1580	Phil Koga	33	Club high activity
1722	Tom Strobent	33	Club inactive
1750	Nisha Mookerjee	28	Club medium activity

/**

* Creating and Using Macro Variables: Practice #2 *

**/

/*Level 2 Practice: Using Macro Variable References with Delimiters

Reminder: If you restarted your SAS session, you must submit the libname.sas program in the EMC1V2 folder to access your data.

Open the m102p02.sas program from the practices folder. Submit the program and review the results.

Verify that the report contains 17 rows.

Create macro variables Lib, Dsn, and Var and assign the values mc1, newhires, and Employee respectively.

Modify every occurrence of mc1, newhires, and Employee so that they are replaced by references to the corresponding macro variable. Note: Be sure to include the 's' in Employees as part of the title text.

Submit the program.

What is the report title?

Hint: you can copy and paste the title from the report.

How many rows are in the report?

*/

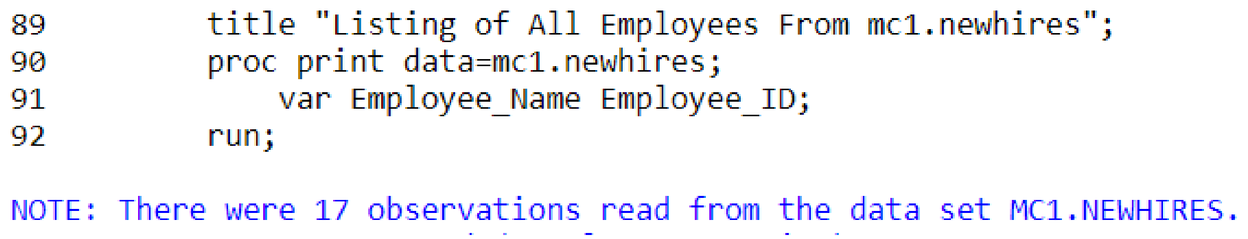
title "Listing of All Employees From mc1.newhires";

proc print data=mc1.newhires;

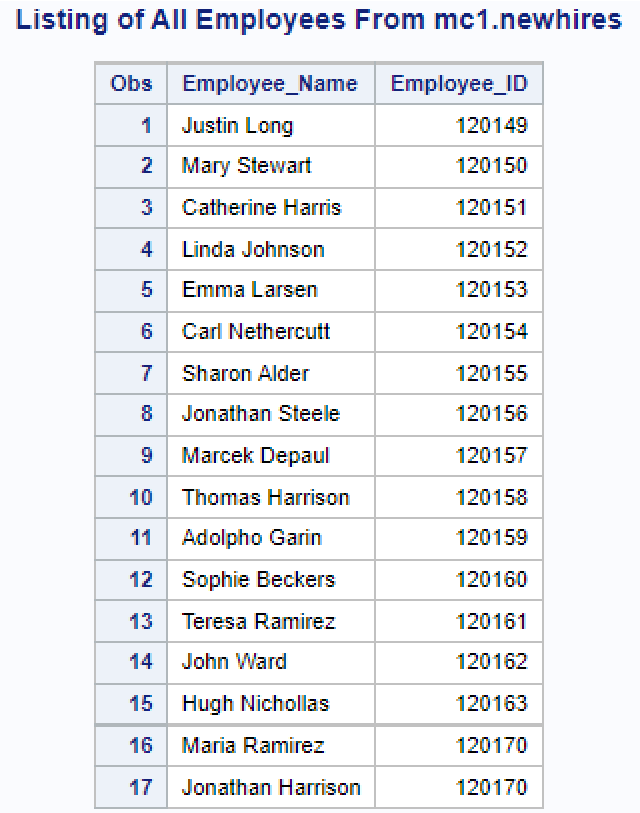
   var Employee_Name Employee_ID;

run;

title;



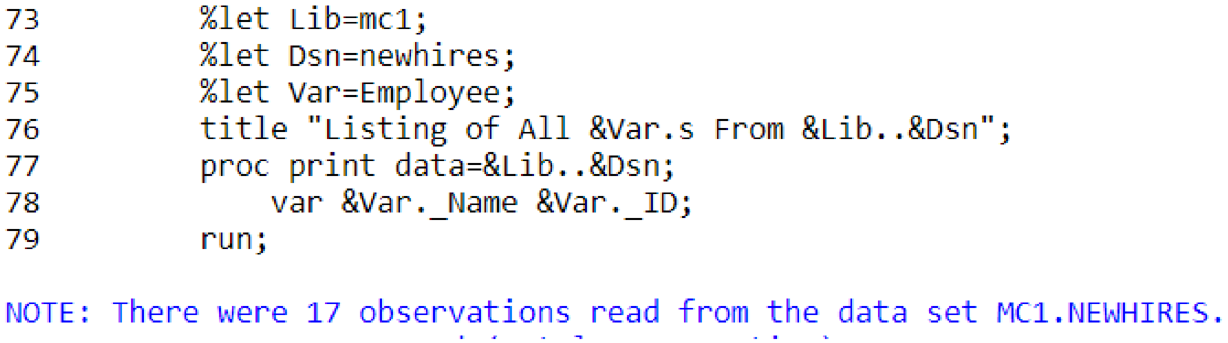
```
title "Listing of All Employees From mc1.newhires";
proc print data=mc1.newhires;
    var Employee_Name Employee_ID;
run;

NOTE: There were 17 observations read from the data set MC1.NEWHIRES.
```



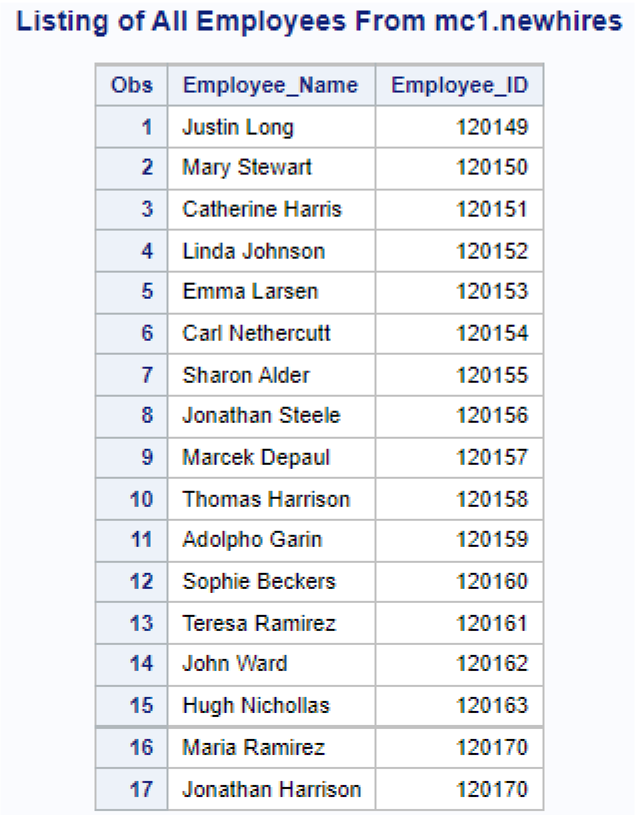
Listing of All Employees From mc1.newhires

Obs	Employee_Name	Employee_ID
1	Justin Long	120149
2	Mary Stewart	120150
3	Catherine Harris	120151
4	Linda Johnson	120152
5	Emma Larsen	120153
6	Carl Nethercutt	120154
7	Sharon Alder	120155
8	Jonathan Steele	120156
9	Marcek Depaul	120157
10	Thomas Harrison	120158
11	Adolpho Garin	120159
12	Sophie Beckers	120160
13	Teresa Ramirez	120161
14	John Ward	120162
15	Hugh Nichollas	120163
16	Maria Ramirez	120170
17	Jonathan Harrison	120170

```
%let Lib=mc1;

%let Dsn=newhires;

%let Var=Employee;

title "Listing of All &Var.s From &Lib..&Dsn";

proc print data=&Lib..&Dsn;

   var &Var._Name &Var._ID;

run;

title;
```



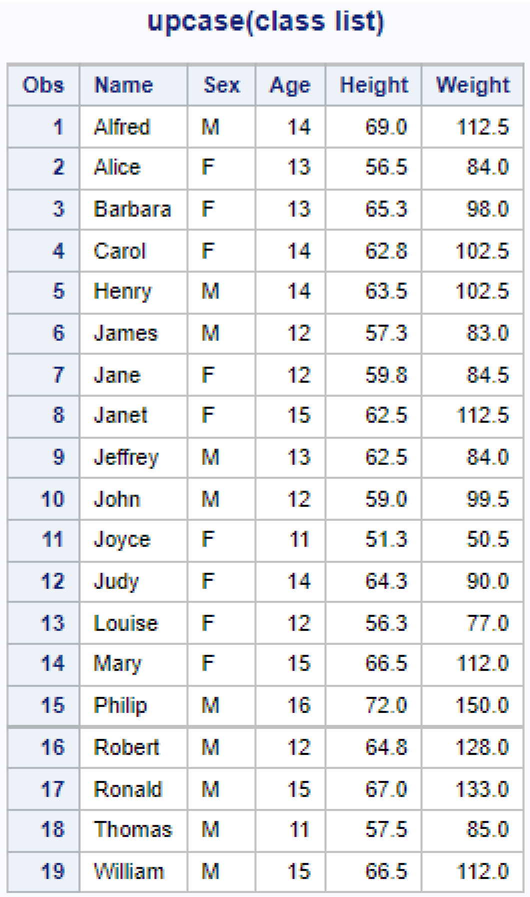
```
 73          %let Lib=mc1;
 74          %let Dsn=newhires;
 75          %let Var=Employee;
 76          title "Listing of All &Var.s From &Lib..&Dsn";
 77          proc print data=&Lib..&Dsn;
 78              var &Var._Name &Var._ID;
 79          run;

 NOTE: There were 17 observations read from the data set MC1.NEWHIRES.
```



Listing of All Employees From mc1.newhires

Obs	Employee_Name	Employee_ID
1	Justin Long	120149
2	Mary Stewart	120150
3	Catherine Harris	120151
4	Linda Johnson	120152
5	Emma Larsen	120153
6	Carl Nethercutt	120154
7	Sharon Alder	120155
8	Jonathan Steele	120156
9	Marcek Depaul	120157
10	Thomas Harrison	120158
11	Adolpho Garin	120159
12	Sophie Beckers	120160
13	Teresa Ramirez	120161
14	John Ward	120162
15	Hugh Nichollas	120163
16	Maria Ramirez	120170
17	Jonathan Harrison	120170

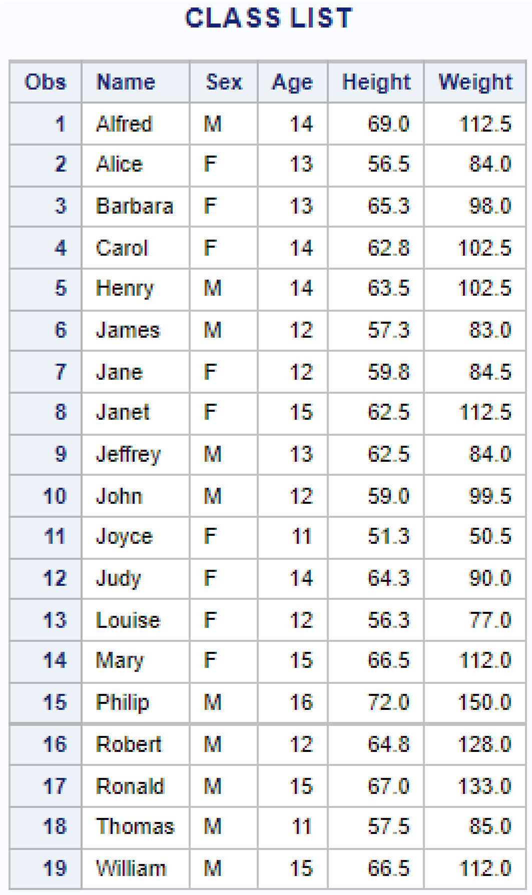
```
***********************************************************;
*  Activity 3.01                              *;
*  1) Examine the TITLE statement. What text will appear  *;
*     as the title? Run the program and view the results. *;
*  2) Add % before the UPCASE function in the TITLE       *;
*     statement. Run the program. What text appears as    *;
*     the title?                              *;
***********************************************************;

%let text=class list;
title "upcase(&text)";
proc print data=sashelp.class;
run;
title;
```



upcase(class list)

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84.0
10	John	M	12	59.0	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90.0
13	Louise	F	12	56.3	77.0
14	Mary	F	15	66.5	112.0
15	Philip	M	16	72.0	150.0
16	Robert	M	12	64.8	128.0
17	Ronald	M	15	67.0	133.0
18	Thomas	M	11	57.5	85.0
19	William	M	15	66.5	112.0

```
%let text=class list;

title "%upcase(&text)";

proc print data=sashelp.class;

run;

title;
```

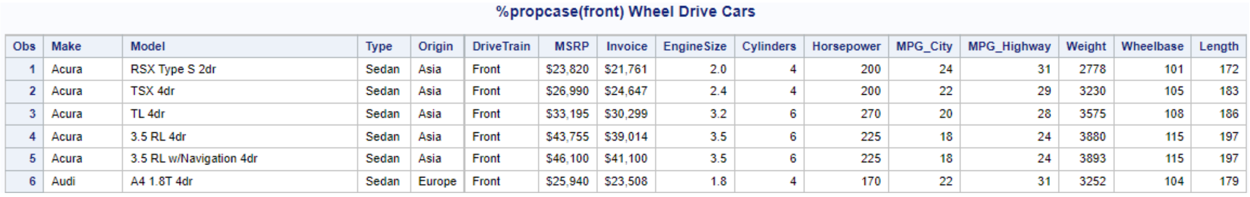


CLASS LIST

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84.0
10	John	M	12	59.0	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90.0
13	Louise	F	12	56.3	77.0
14	Mary	F	15	66.5	112.0
15	Philip	M	16	72.0	150.0
16	Robert	M	12	64.8	128.0
17	Ronald	M	15	67.0	133.0
18	Thomas	M	11	57.5	85.0
19	William	M	15	66.5	112.0

```
******************************************************;
*  Activity 3.02                                  *;
*  1) Run the program and examine the output and the log. *;
*     SAS does not have a %PROPCASE macro function, so it *;
*     does not successfully resolve and the title is     *;
*     incorrect.                                   *;
*  2) Modify the TITLE statement to use the %SYSFUNC     *;
*     macro function in combination with the PROPCASE    *;
*     function.                                    *;
*     title "%sysfunc(propcase(&dt)) Wheel Drive Cars";   *;
*  3) Run the program and confirm that the title is      *;
*     correct.                                     *;
******************************************************;
```

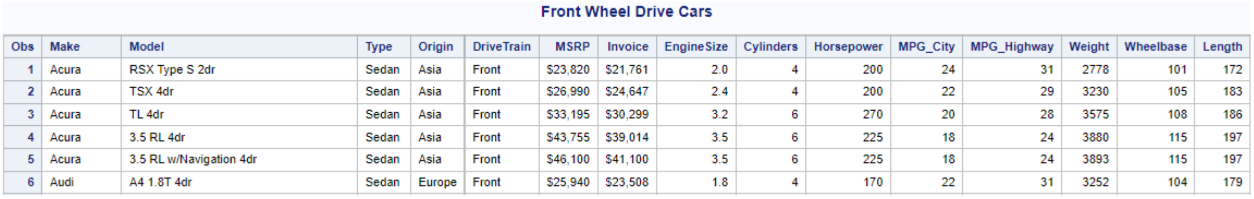
```
%let dt=front;

data cars_subset;

   set sashelp.cars;

   where upcase(DriveTrain)="%upcase(&dt)";

run;


title "%propcase(&dt) Wheel Drive Cars";

footnote "Listing from %scan(&syslast,2) Table";

proc print data=&syslast;

run;
```



%propcase(front) Wheel Drive Cars

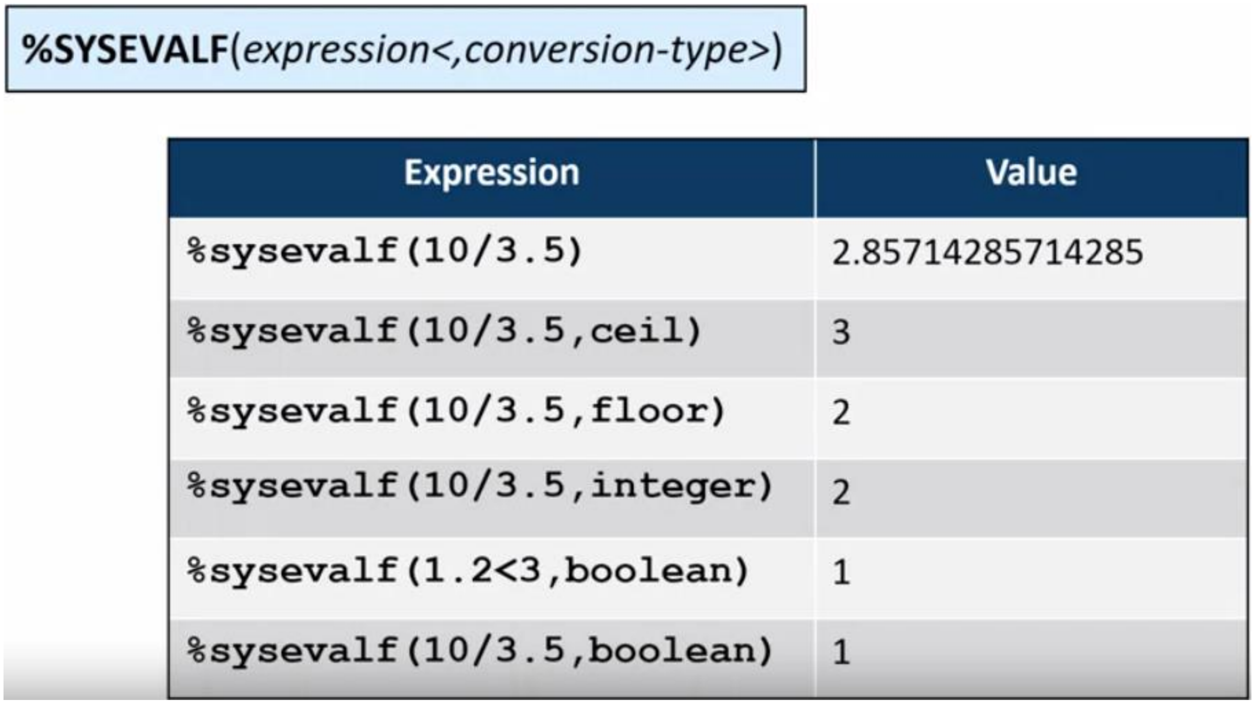
Obs	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4	200	24	31	2778	101	172
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4	200	22	29	3230	105	183
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6	270	20	28	3575	108	186
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6	225	18	24	3880	115	197
5	Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	$46,100	$41,100	3.5	6	225	18	24	3893	115	197
6	Audi	A4 1.8T 4dr	Sedan	Europe	Front	$25,940	$23,508	1.8	4	170	22	31	3252	104	179

```
%let dt=front;

data cars_subset;

   set sashelp.cars;

   where upcase(DriveTrain)="%upcase(&dt)";

run;


title "%sysfunc(propcase(&dt)) Wheel Drive Cars";

footnote "Listing from %scan(&syslast,2) Table";

proc print data=&syslast;

run;
```



Front Wheel Drive Cars

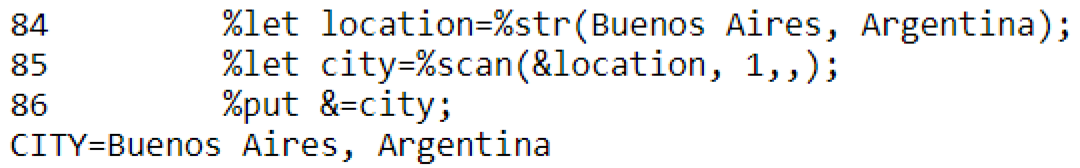
Obs	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4	200	24	31	2778	101	172
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4	200	22	29	3230	105	183
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6	270	20	28	3575	108	186
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6	225	18	24	3880	115	197
5	Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	$46,100	$41,100	3.5	6	225	18	24	3893	115	197
6	Audi	A4 1.8T 4dr	Sedan	Europe	Front	$25,940	$23,508	1.8	4	170	22	31	3252	104	179



%SYSEVALF(*expression<,conversion-type>*)

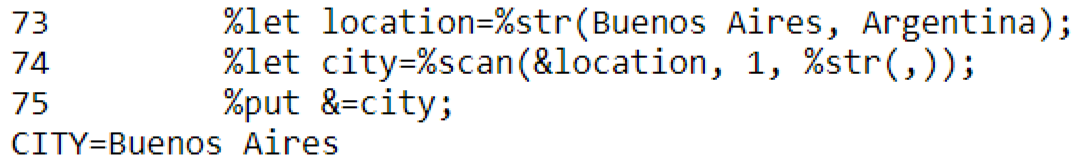
Expression	Value
`%sysevalf(10/3.5)`	2.85714285714285
`%sysevalf(10/3.5,ceil)`	3
`%sysevalf(10/3.5,floor)`	2
`%sysevalf(10/3.5,integer)`	2
`%sysevalf(1.2<3,boolean)`	1
`%sysevalf(10/3.5,boolean)`	1

```
********************************************************;
*  Activity 3.03                                 *;
*  1) The intent of the %SCAN function is to return the   *;
*     city from the Location macro variable using only    *;
*     the comma as a delimiter. Run the program and       *;
*     examine the log. Why does the program fail?         *;
*  2) Use %STR to mask the comma in the appropriate       *;
*     places so that the value of the City macro variable *;
*     is Buenos Aires.                                *;
********************************************************;


%let location=%str(Buenos Aires, Argentina);

%let city=%scan(&location, 1,,);

%put &=city;
```



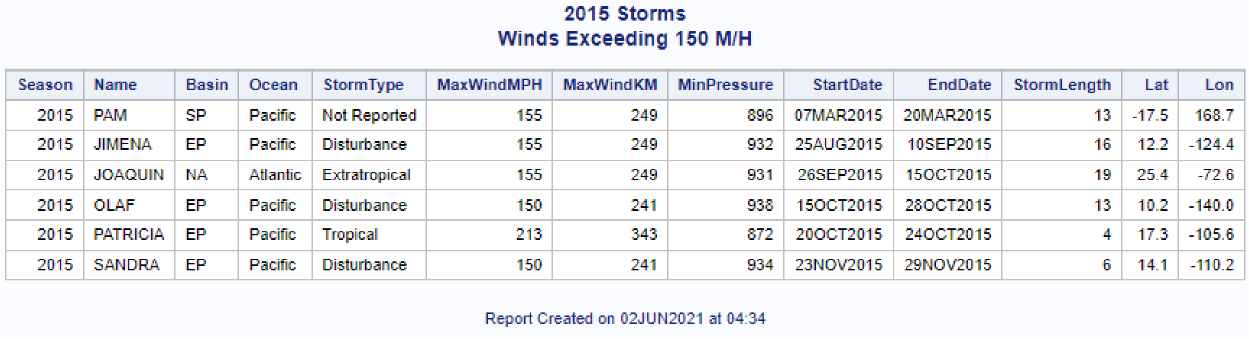
```
84         %let location=%str(Buenos Aires, Argentina);
85         %let city=%scan(&location, 1,,);
86         %put &=city;
CITY=Buenos Aires, Argentina
```

```
%let location=%str(Buenos Aires, Argentina);

%let city=%scan(&location, 1, %str(,));

%put &=city;
```



```
73         %let location=%str(Buenos Aires, Argentina);
74         %let city=%scan(&location, 1, %str(,));
75         %put &=city;
CITY=Buenos Aires
```

```
/***********************

* Macro Functions: Demo *

***********************/


%let year=2015;

%let windm=150;


title1 "&year Storms";

title2 "Winds Exceeding &windm M/H";

footnote "Report Created on &sysdate9 at &systime";

proc print data=mc1.storm_final noobs;

  where Season=&year and MaxWindMPH>=&windm;

run;

title;footnote;
```

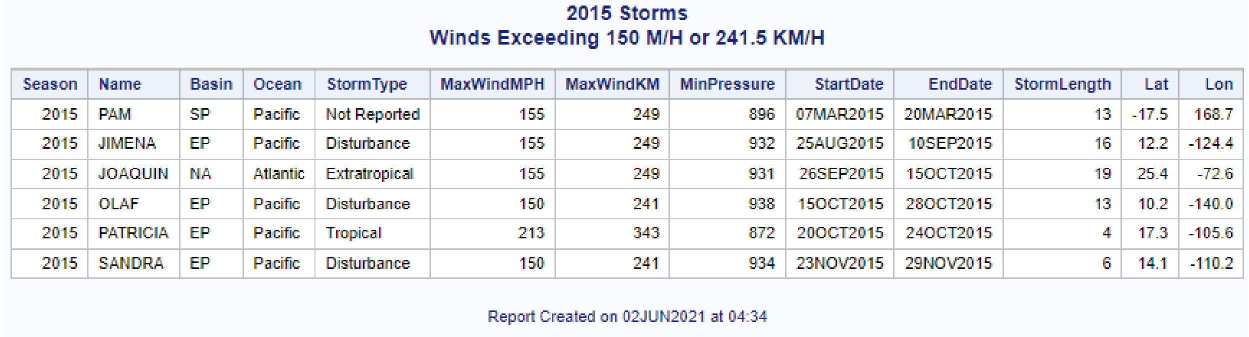


2015 Storms
Winds Exceeding 150 M/H

Season	Name	Basin	Ocean	StormType	MaxWindMPH	MaxWindKM	MinPressure	StartDate	EndDate	StormLength	Lat	Lon
2015	PAM	SP	Pacific	Not Reported	155	249	896	07MAR2015	20MAR2015	13	-17.5	168.7
2015	JIMENA	EP	Pacific	Disturbance	155	249	932	25AUG2015	10SEP2015	16	12.2	-124.4
2015	JOAQUIN	NA	Atlantic	Extratropical	155	249	931	26SEP2015	15OCT2015	19	25.4	-72.6
2015	OLAF	EP	Pacific	Disturbance	150	241	938	15OCT2015	28OCT2015	13	10.2	-140.0
2015	PATRICIA	EP	Pacific	Tropical	213	343	872	20OCT2015	24OCT2015	4	17.3	-105.6
2015	SANDRA	EP	Pacific	Disturbance	150	241	934	23NOV2015	29NOV2015	6	14.1	-110.2

Report Created on 02JUN2021 at 04:34

```
%let year=2015;

%let windm=150;


title1 "&year Storms";

title2 "Winds Exceeding &windm M/H or %sysevalf(&windm*1.61) KM/H";

footnote "Report Created on &sysdate9 at &systime";

proc print data=mc1.storm_final noobs;

  where Season=&year and MaxWindMPH>=&windm;

run;

title;footnote;
```



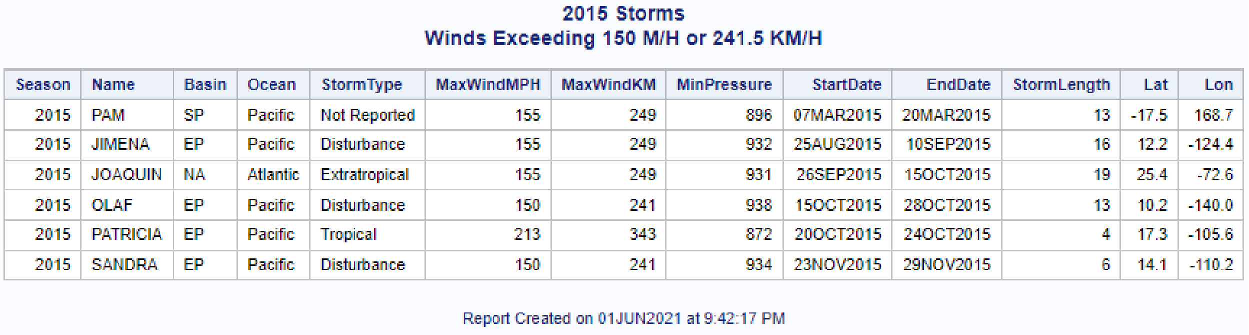
2015 Storms
Winds Exceeding 150 M/H or 241.5 KM/H

Season	Name	Basin	Ocean	StormType	MaxWindMPH	MaxWindKM	MinPressure	StartDate	EndDate	StormLength	Lat	Lon
2015	PAM	SP	Pacific	Not Reported	155	249	896	07MAR2015	20MAR2015	13	-17.5	168.7
2015	JIMENA	EP	Pacific	Disturbance	155	249	932	25AUG2015	10SEP2015	16	12.2	-124.4
2015	JOAQUIN	NA	Atlantic	Extratropical	155	249	931	26SEP2015	15OCT2015	19	25.4	-72.6
2015	OLAF	EP	Pacific	Disturbance	150	241	938	15OCT2015	28OCT2015	13	10.2	-140.0
2015	PATRICIA	EP	Pacific	Tropical	213	343	872	20OCT2015	24OCT2015	4	17.3	-105.6
2015	SANDRA	EP	Pacific	Disturbance	150	241	934	23NOV2015	29NOV2015	6	14.1	-110.2

Report Created on 02JUN2021 at 04:34

```
%let year=2015;

%let windm=150;


title1 "&year Storms";

title2 "Winds Exceeding &windm M/H or %sysevalf(&windm*1.61) KM/H";

footnote "Report Created on %sysfunc(today(), date9.) at %sysfunc(time(), timeampm.)";

proc print data=mc1.storm_final noobs;
```

```
  where Season=&year and MaxWindMPH>=&windm;
run;
title;footnote;
```

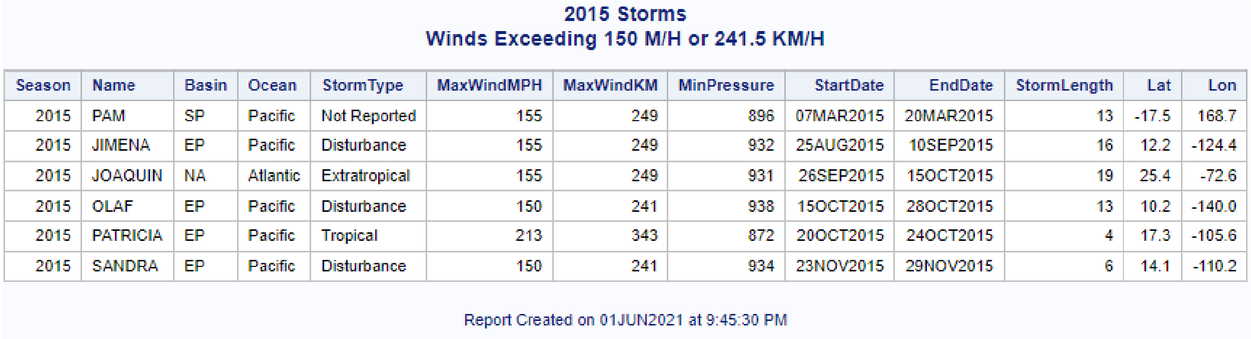


2015 Storms
Winds Exceeding 150 M/H or 241.5 KM/H

Season	Name	Basin	Ocean	StormType	MaxWindMPH	MaxWindKM	MinPressure	StartDate	EndDate	StormLength	Lat	Lon
2015	PAM	SP	Pacific	Not Reported	155	249	896	07MAR2015	20MAR2015	13	-17.5	168.7
2015	JIMENA	EP	Pacific	Disturbance	155	249	932	25AUG2015	10SEP2015	16	12.2	-124.4
2015	JOAQUIN	NA	Atlantic	Extratropical	155	249	931	26SEP2015	15OCT2015	19	25.4	-72.6
2015	OLAF	EP	Pacific	Disturbance	150	241	938	15OCT2015	28OCT2015	13	10.2	-140.0
2015	PATRICIA	EP	Pacific	Tropical	213	343	872	20OCT2015	24OCT2015	4	17.3	-105.6
2015	SANDRA	EP	Pacific	Disturbance	150	241	934	23NOV2015	29NOV2015	6	14.1	-110.2

Report Created on 01JUN2021 at 9:42:17 PM

```
%let year=2015;
%let windm=150;
%let dtfoot=%str(footnote "Report Created on %sysfunc(today(), date9.) at %sysfunc(time(), timeampm.)";);


title1 "&year Storms";
title2 "Winds Exceeding &windm M/H or %sysevalf(&windm*1.61) KM/H";
&dtfoot
proc print data=mc1.storm_final noobs;
  where Season=&year and MaxWindMPH>=&windm;
run;
title;footnote;
```



2015 Storms
Winds Exceeding 150 M/H or 241.5 KM/H

Season	Name	Basin	Ocean	StormType	MaxWindMPH	MaxWindKM	MinPressure	StartDate	EndDate	StormLength	Lat	Lon
2015	PAM	SP	Pacific	Not Reported	155	249	896	07MAR2015	20MAR2015	13	-17.5	168.7
2015	JIMENA	EP	Pacific	Disturbance	155	249	932	25AUG2015	10SEP2015	16	12.2	-124.4
2015	JOAQUIN	NA	Atlantic	Extratropical	155	249	931	26SEP2015	15OCT2015	19	25.4	-72.6
2015	OLAF	EP	Pacific	Disturbance	150	241	938	15OCT2015	28OCT2015	13	10.2	-140.0
2015	PATRICIA	EP	Pacific	Tropical	213	343	872	20OCT2015	24OCT2015	4	17.3	-105.6
2015	SANDRA	EP	Pacific	Disturbance	150	241	934	23NOV2015	29NOV2015	6	14.1	-110.2

Report Created on 01JUN2021 at 9:45:30 PM

```
/*******************************
* Macro Functions: Practice #1 *
```

*******************************/

/*Level 1 Practice: Using the %UPCASE and %SCAN Functions

Reminder: If you restarted your SAS session, you must submit the libname.sas program in the EMC1V2 folder to access your data.

Open m103p01.sas from the practices folder.

Add a %LET statement to convert the value of FullName to uppercase and assign the result to FullName.

Write a single %PUT statement to display FullName in a sentence using both its current form and proper case,

as shown here: ANTHONY MILLER in proper case is Anthony Miller.

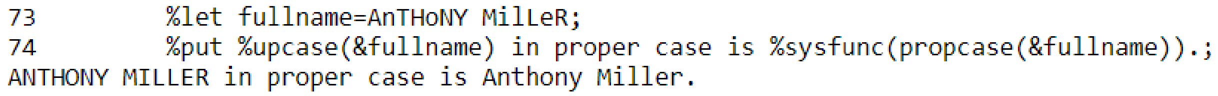
Note: Use %SYSFUNC to execute the PROPCASE function.

Submit the program and view the log to verify the results.

Did you get the results you expected?

*/

```
%let fullname=AnTHoNY MilLeR;

%put %upcase(&fullname) in proper case is %sysfunc(propcase(&fullname)).;
```



```
73         %let fullname=AnTHoNY MilLeR;
74         %put %upcase(&fullname) in proper case is %sysfunc(propcase(&fullname)).;
ANTHONY MILLER in proper case is Anthony Miller.
```

```
*Solution;

%let fullname=AnTHoNY MilLeR;

%put &fullname;


%let fullname=%upcase(&fullname);

%put &fullname in proper case is %sysfunc(propcase(&fullname)).;
```

/*Add a %LET statement to extract the first name from FullName, convert it to proper case,

and assign the result to a macro variable named First.

Add another %LET statement to extract the last name from FullName, convert it to proper case,

and assign the result to a macro variable named Last.

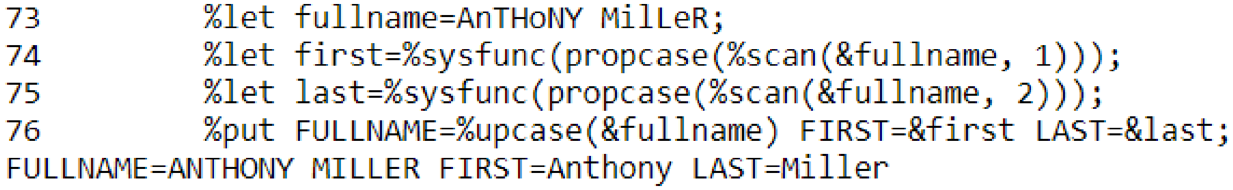
Display the values of FullName, First, and Last in the log as shown here:

FULLNAME=ANTHONY MILLER FIRST=Anthony LAST=Miller

Did the log display the results you expected?

*/

```
%let fullname=AnTHoNY MilLeR;

%let first=%sysfunc(propcase(%scan(&fullname, 1)));

%let last=%sysfunc(propcase(%scan(&fullname, 2)));

%put FULLNAME=%upcase(&fullname) FIRST=&first LAST=&last;
```



```
          %let fullname=AnTHoNY MilLeR;
          %let first=%sysfunc(propcase(%scan(&fullname, 1)));
          %let last=%sysfunc(propcase(%scan(&fullname, 2)));
          %put FULLNAME=%upcase(&fullname) FIRST=&first LAST=&last;
FULLNAME=ANTHONY MILLER FIRST=Anthony LAST=Miller
```

```
*Solution;

%let first=%sysfunc(propcase(%scan(&fullname,1)));

%let last=%sysfunc(propcase(%scan(&fullname,-1)));

%put &=fullname &=first &=last;
```

/*Add a %SYMDEL statement to delete FullName, First, and Last from the global symbol table.

Use a %PUT statement to write the values of all user-defined macro variables to the log.

Submit the two statements.

Are FullName, First, and Last in the user-defined macros list?

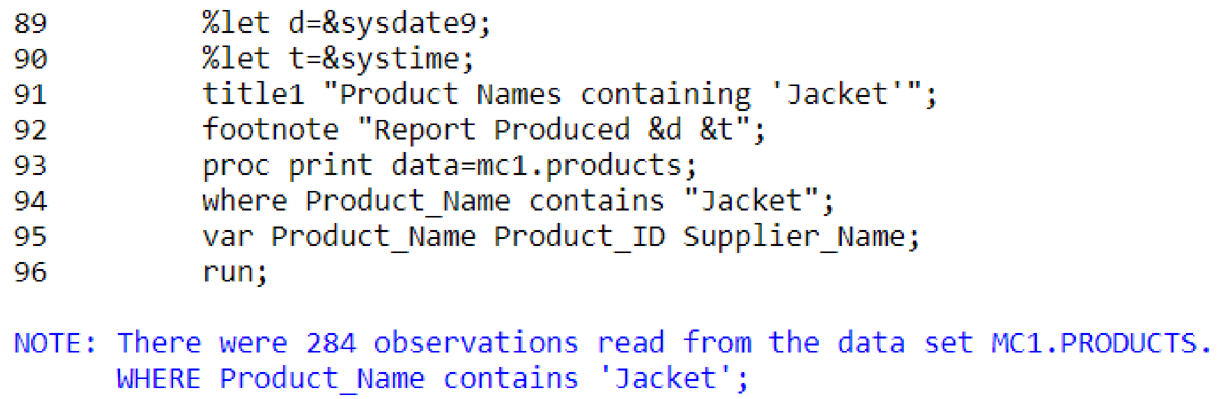
*/

```
%SYMDEL FullName First Last;

%put _all_;
```

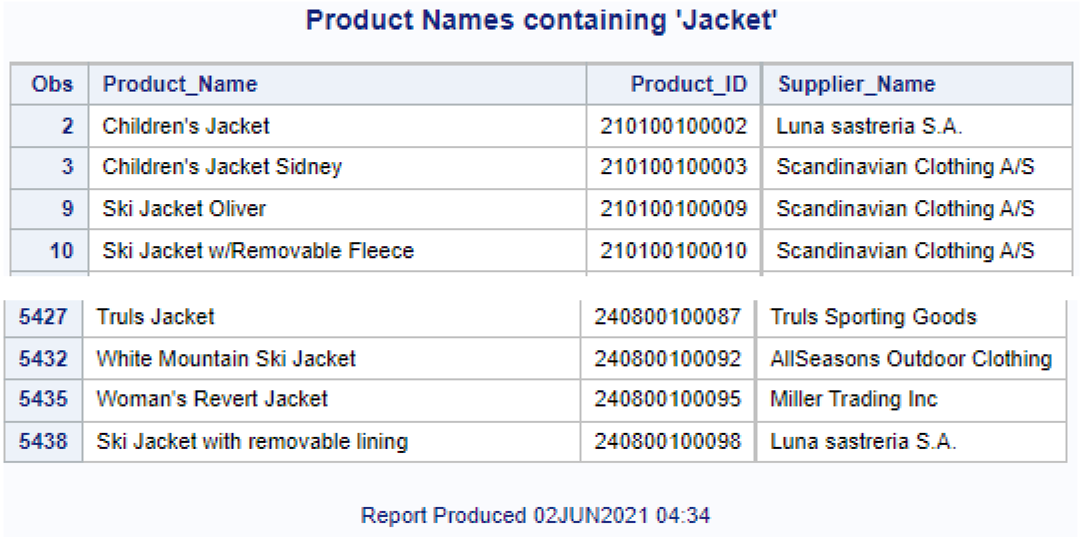
/******************************

* Macro Functions: Practice #2 *

```
*******************************/
/*Level 2 Practice: Using Macro Quoting Functions

Reminder: If you restarted your SAS session, you must submit the libname.sas program in the EMC1V2 folder to access your data.

Open m103p02.sas from the practices folder and review the code.

Submit the program and review the results. Note that the footnote includes the date and time that the SAS session started.

Modify the program to create a macro variable named Product and assign the value R&D.

Reference Product in the TITLE and WHERE statements, replacing Jacket.

Modify the FOOTNOTE statement to display the current date and time, using the DATE9 and TIMEAMPM9 formats respectively.

Submit the program and verify that the title is Product Names Containing 'R&D' and

that the current date and time are displayed in the footnote.

How many rows are in the report?

Note: Type a numeric value for your answer.

*/


%let d=&sysdate9;

%let t=&systime;

title1 "Product Names containing 'Jacket'";

footnote "Report Produced &d &t";

proc print data=mc1.products;

	where Product_Name contains "Jacket";

	var Product_Name Product_ID Supplier_Name;

run;

title;footnote;
```



```
%let d=&sysdate9;
%let t=&systime;
title1 "Product Names containing 'Jacket'";
footnote "Report Produced &d &t";
proc print data=mc1.products;
where Product_Name contains "Jacket";
var Product_Name Product_ID Supplier_Name;
run;

NOTE: There were 284 observations read from the data set MC1.PRODUCTS.
      WHERE Product_Name contains 'Jacket';
```

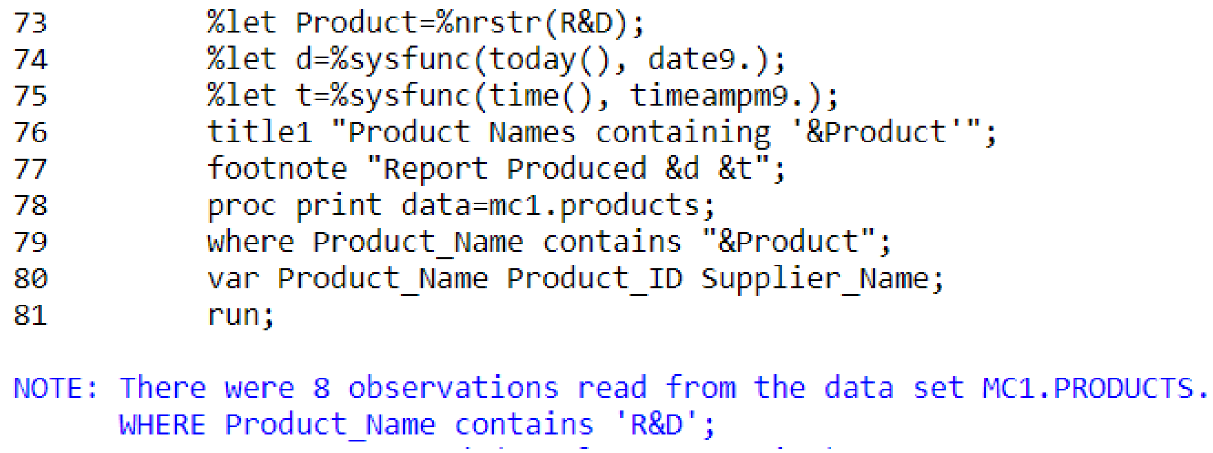


Product Names containing 'Jacket'

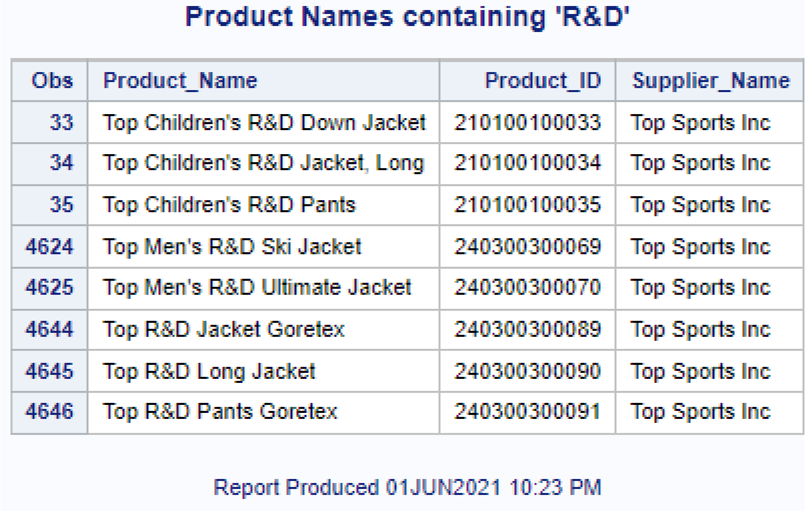
Obs	Product_Name	Product_ID	Supplier_Name
2	Children's Jacket	210100100002	Luna sastreria S.A.
3	Children's Jacket Sidney	210100100003	Scandinavian Clothing A/S
9	Ski Jacket Oliver	210100100009	Scandinavian Clothing A/S
10	Ski Jacket w/Removable Fleece	210100100010	Scandinavian Clothing A/S
5427	Truls Jacket	240800100087	Truls Sporting Goods
5432	White Mountain Ski Jacket	240800100092	AllSeasons Outdoor Clothing
5435	Woman's Revert Jacket	240800100095	Miller Trading Inc
5438	Ski Jacket with removable lining	240800100098	Luna sastreria S.A.

Report Produced 02JUN2021 04:34

```
%let Product=%nrstr(R&D);

%let d=%sysfunc(today(), date9.);

%let t=%sysfunc(time(), timeampm9.);

title1 "Product Names containing '&Product'";

footnote "Report Produced &d &t";

proc print data=mc1.products;

        where Product_Name contains "&Product";

        var Product_Name Product_ID Supplier_Name;

run;

title;footnote;
```



```
%let Product=%nrstr(R&D);
%let d=%sysfunc(today(), date9.);
%let t=%sysfunc(time(), timeampm9.);
title1 "Product Names containing '&Product'";
footnote "Report Produced &d &t";
proc print data=mc1.products;
where Product_Name contains "&Product";
var Product_Name Product_ID Supplier_Name;
run;

NOTE: There were 8 observations read from the data set MC1.PRODUCTS.
      WHERE Product_Name contains 'R&D';
```



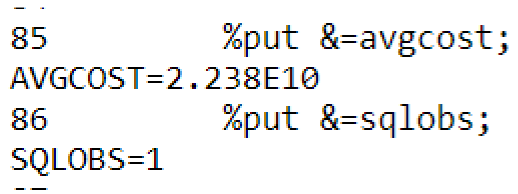
Product Names containing 'R&D'

Obs	Product_Name	Product_ID	Supplier_Name
33	Top Children's R&D Down Jacket	210100100033	Top Sports Inc
34	Top Children's R&D Jacket, Long	210100100034	Top Sports Inc
35	Top Children's R&D Pants	210100100035	Top Sports Inc
4624	Top Men's R&D Ski Jacket	240300300069	Top Sports Inc
4625	Top Men's R&D Ultimate Jacket	240300300070	Top Sports Inc
4644	Top R&D Jacket Goretex	240300300089	Top Sports Inc
4645	Top R&D Long Jacket	240300300090	Top Sports Inc
4646	Top R&D Pants Goretex	240300300091	Top Sports Inc

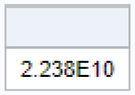
Report Produced 01JUN2021 10:23 PM

```
/*********************************************

* Using SQL to Create Macro Variables: Demo *

*********************************************/


 /* Section 1 */


proc sql;

select mean(cost)

   into :avgcost

   from mc1.storm_damage;

quit;
```

```
%put &=avgcost;

%put &=sqlobs;
```

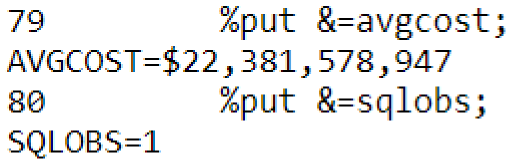


```
85          %put &=avgcost;
AVGCOST=2.238E10
86          %put &=sqlobs;
SQLOBS=1
```



2.238E10

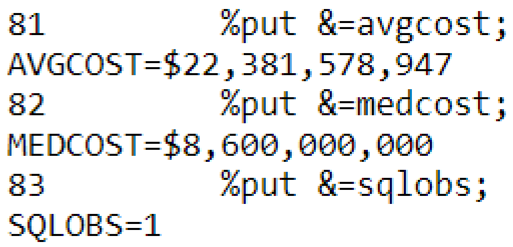
```
proc sql noprint;

select mean(cost) format=dollar20.

   into :avgcost trimmed

   from mc1.storm_damage;

quit;


%put &=avgcost;

%put &=sqlobs;
```



```
79          %put &=avgcost;
AVGCOST=$22,381,578,947
80          %put &=sqlobs;
SQLOBS=1
```

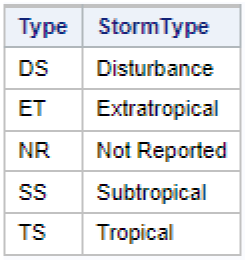
```
proc sql noprint;

select mean(cost) format=dollar20.,

             median(cost) format=dollar20.

   into :avgcost trimmed,

       :medcost trimmed

   from mc1.storm_damage;

quit;


%put &=avgcost;

%put &=medcost;
```

```
%put &=sqlobs;
```



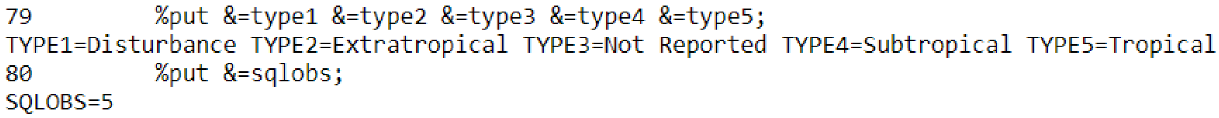
```
81          %put &=avgcost;
AVGCOST=$22,381,578,947
82          %put &=medcost;
MEDCOST=$8,600,000,000
83          %put &=sqlobs;
SQLOBS=1
```

```
/* Section 2 */


proc sql;

select *

          from mc1.storm_type_codes;

quit;
```

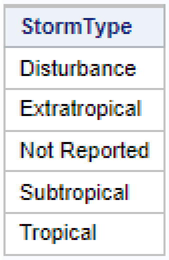


Type	StormType
DS	Disturbance
ET	Extratropical
NR	Not Reported
SS	Subtropical
TS	Tropical

```
proc sql;

select StormType

          into :Type1-

          from mc1.storm_type_codes;

quit;


%put &=type1 &=type2 &=type3 &=type4 &=type5;

%put &=sqlobs;
```



```
79          %put &=type1 &=type2 &=type3 &=type4 &=type5;
TYPE1=Disturbance TYPE2=Extratropical TYPE3=Not Reported TYPE4=Subtropical TYPE5=Tropical
80          %put &=sqlobs;
SQLOBS=5
```



StormType
Disturbance
Extratropical
Not Reported
Subtropical
Tropical

proc sql;

select StormType

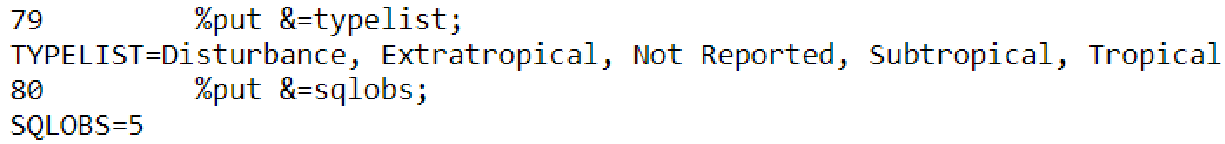
	into :typelist separated by ", "

	from mc1.storm_type_codes;

quit;

%put &=typelist;

%put &=sqlobs;



```
79          %put &=typelist;
TYPELIST=Disturbance, Extratropical, Not Reported, Subtropical, Tropical
80          %put &=sqlobs;
SQLOBS=5
```

/***

* Using SQL to Create Maro Variables: Practice #4 *

***/

/*Level 1 Practice: Using PROC SQL to Generate Macro Variables for Use in a Report Title

Reminder: If you restarted your SAS session, you must submit the libname.sas program

in the EMC1V2 folder to access your practice files.

Open m103p04.sas from the practices folder.

Review the code and submit the %LET statements and the PROC SQL step.

Verify that Qty (the mean for Quantity) is 1.43 and Price (the mean for Total_Retail_Price) is 137.72.

In the TITLE2 statement, replace xxx with the mean for Quantity (1.43), and

replace yyy with the mean for Total_Retail_Price (137.72).

Submit the TITLE statements and the PROC PRINT step, and review the log and results.

How many rows were read from the input table?

Note: Type a numeric value for the answer.

In the PROC SQL step, add an INTO clause to assign the mean for Quantity to a macro variable named Qty,

and the mean for Total_Retail_Price to a macro variable named Price.

In the TITLE2 statement, replace the hardcoded mean values with references to Qty and Price.

Submit the entire program and review the log and results.

Verify that the title displayed correctly.

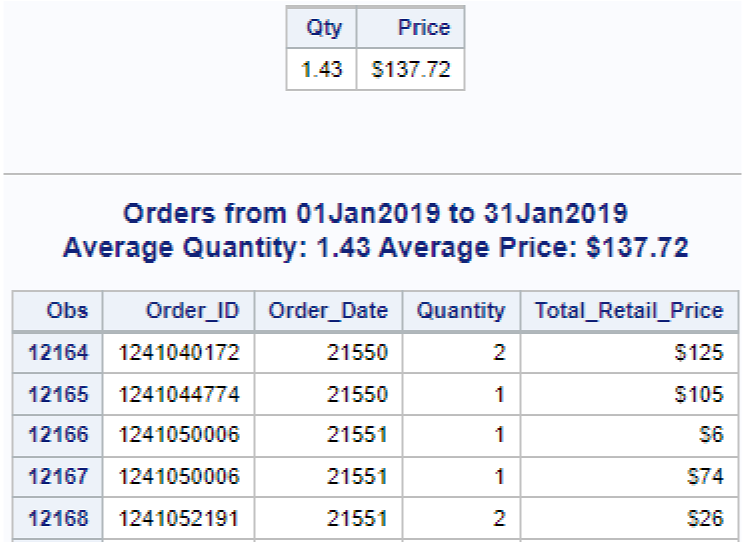
Think about why the average price is displayed with a leading dollar sign in the title.

How many rows were read from the input table?

Note: Type a numeric value for your answer.

```
*/

%let start=01Jan2019;

%let stop=31Jan2019;

proc sql;

        select mean(Quantity) format 4.2 as Qty,

            mean(Total_Retail_Price) format=dollar7.2 as Price

        into :qty, :price

        from mc1.orders

        where Order_Date between "&start"d and "&stop"d;

quit;


title1 "Orders from &start to &stop";

title2 "Average Quantity: &qty   Average Price: &price";

proc print data=mc1.orders;

        where Order_Date between "&start"d and "&stop"d;

        var Order_ID Order_Date Quantity Total_Retail_Price;

        sum Quantity Total_Retail_Price;

        format Total_Retail_Price dollar8.;
```

```
run;

title;
```



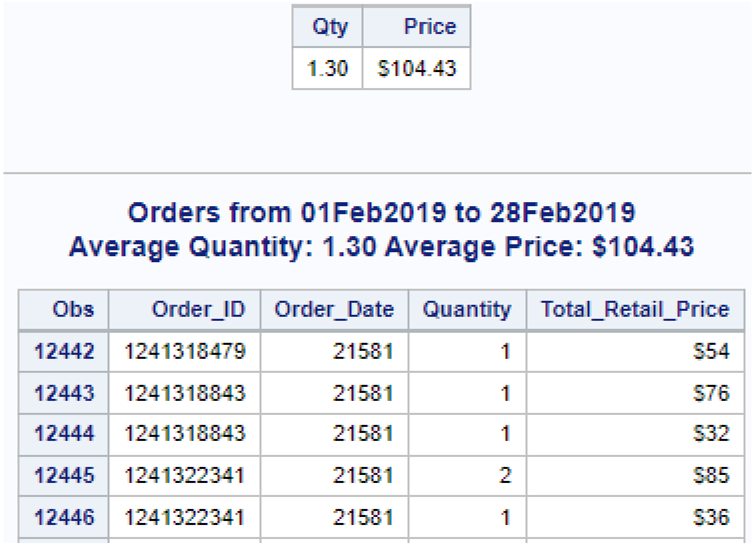
Qty	Price
1.43	$137.72

Orders from 01Jan2019 to 31Jan2019
Average Quantity: 1.43 Average Price: $137.72

Obs	Order_ID	Order_Date	Quantity	Total_Retail_Price
12164	1241040172	21550	2	$125
12165	1241044774	21550	1	$105
12166	1241050006	21551	1	$6
12167	1241050006	21551	1	$74
12168	1241052191	21551	2	$26

```
/*Question 3

Modify the %LET statements to assign 01Feb2019 to start and 28Feb2019 to stop.

Submit the program again.

What is the resolved value of qty in TITLE2?

*/

%let start=01Feb2019;

%let stop=28Feb2019;

proc sql;

        select mean(Quantity) format 4.2 as Qty,

            mean(Total_Retail_Price) format=dollar7.2 as Price

        into :qty, :price

        from mc1.orders

        where Order_Date between "&start"d and "&stop"d;

quit;


title1 "Orders from &start to &stop";

title2 "Average Quantity: &qty   Average Price: &price";
```

```
proc print data=mc1.orders;

        where Order_Date between "&start"d and "&stop"d;

        var Order_ID Order_Date Quantity Total_Retail_Price;

        sum Quantity Total_Retail_Price;

        format Total_Retail_Price dollar8.;

run;

title;
```



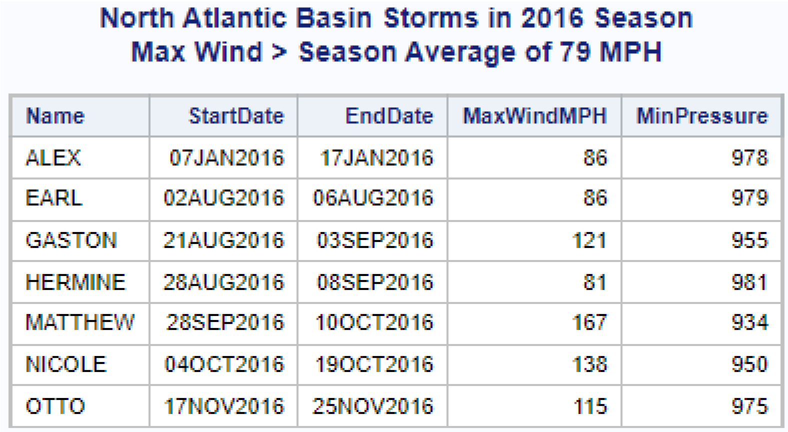
Qty	Price
1.30	$104.43

Orders from 01Feb2019 to 28Feb2019
Average Quantity: 1.30 Average Price: $104.43

Obs	Order_ID	Order_Date	Quantity	Total_Retail_Price
12442	1241318479	21581	1	$54
12443	1241318843	21581	1	$76
12444	1241318843	21581	1	$32
12445	1241322341	21581	2	$85
12446	1241322341	21581	1	$36

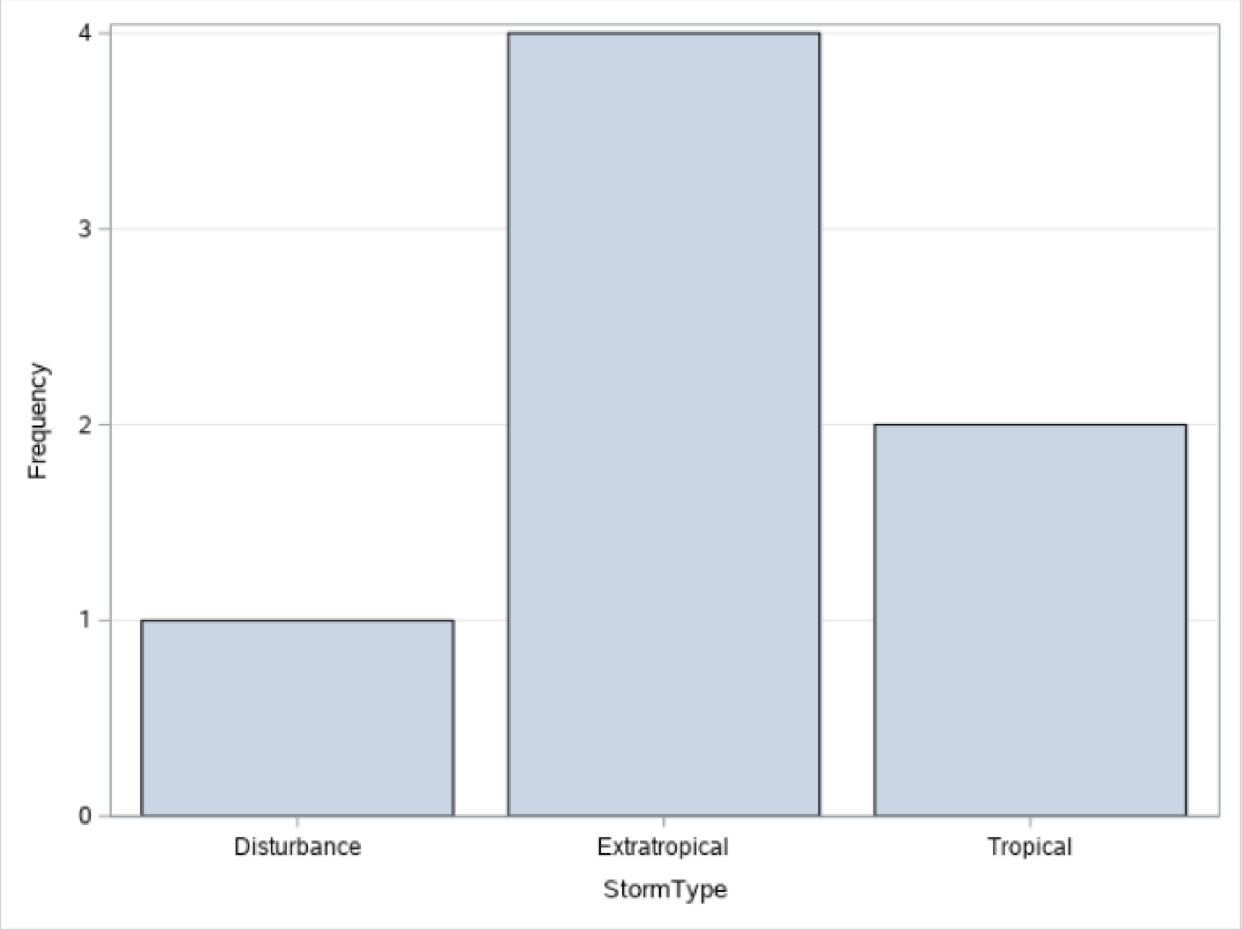
```
/***************************************************

* Using SQL to Create Maro Variables: Practice #5 *

***************************************************/

/*Level 2 Practice: Using PROC SQL to Generate Macro Variables for Use in Subsequent Steps

Reminder: If you restarted your SAS session, you must submit the libname.sas program in the EMC1V2 folder

to access your practice files.

Open m103p05.sas from the practices folder. Review and submit the code in part a.

Verify that the reported average wind speed is 79.

Review the code in part b. Replace every occurrence of XX with the average wind speed from step a.

Submit the code in part b and review the results.

Which bar has the highest frequency?

*/
```

```
 /* Part a. */
%let year=2016;
%let basincode=NA;

proc sql;
select round(mean(MaxWindMPH)) as AvgWind
   from mc1.storm_final
   where Season=&year and Basin="&basincode";
quit;

 /* Part b. */
title1 "North Atlantic Basin Storms in &year Season";
title2 "Max Wind > Season Average of 79 MPH";
proc print data=mc1.storm_final noobs;
          var Name StartDate EndDate MaxWindMPH MinPressure;
          where MaxWindMPH>79 and Season=&year and Basin="&basincode";
run;
title;

proc sgplot data=mc1.storm_final;
   where MaxWindMPH>79 and Season=&year and Basin="&basincode";
          vbar StormType;
          yaxis display=(noline) grid;
run;
```



North Atlantic Basin Storms in 2016 Season
Max Wind > Season Average of 79 MPH

Name	StartDate	EndDate	MaxWindMPH	MinPressure
ALEX	07JAN2016	17JAN2016	86	978
EARL	02AUG2016	06AUG2016	86	979
GASTON	21AUG2016	03SEP2016	121	955
HERMINE	28AUG2016	08SEP2016	81	981
MATTHEW	28SEP2016	10OCT2016	167	934
NICOLE	04OCT2016	19OCT2016	138	950
OTTO	17NOV2016	25NOV2016	115	975



/*Modify the PROC SQL step:

Suppress the PROC SQL output.

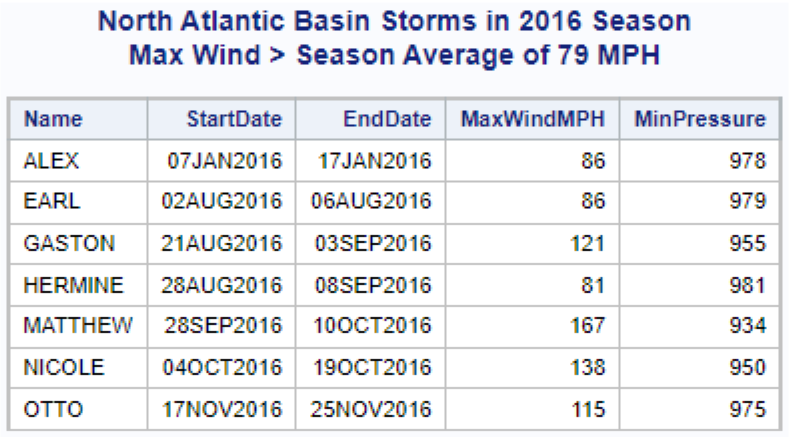
Store the calculated value in a macro variable named AvgWind with no leading spaces.

Add another SELECT statement to select the BasinName value from mc1.storm_basin_codes

where Basin is equal to the basincode macro variable. Write the value to a macro variable named BasinName.

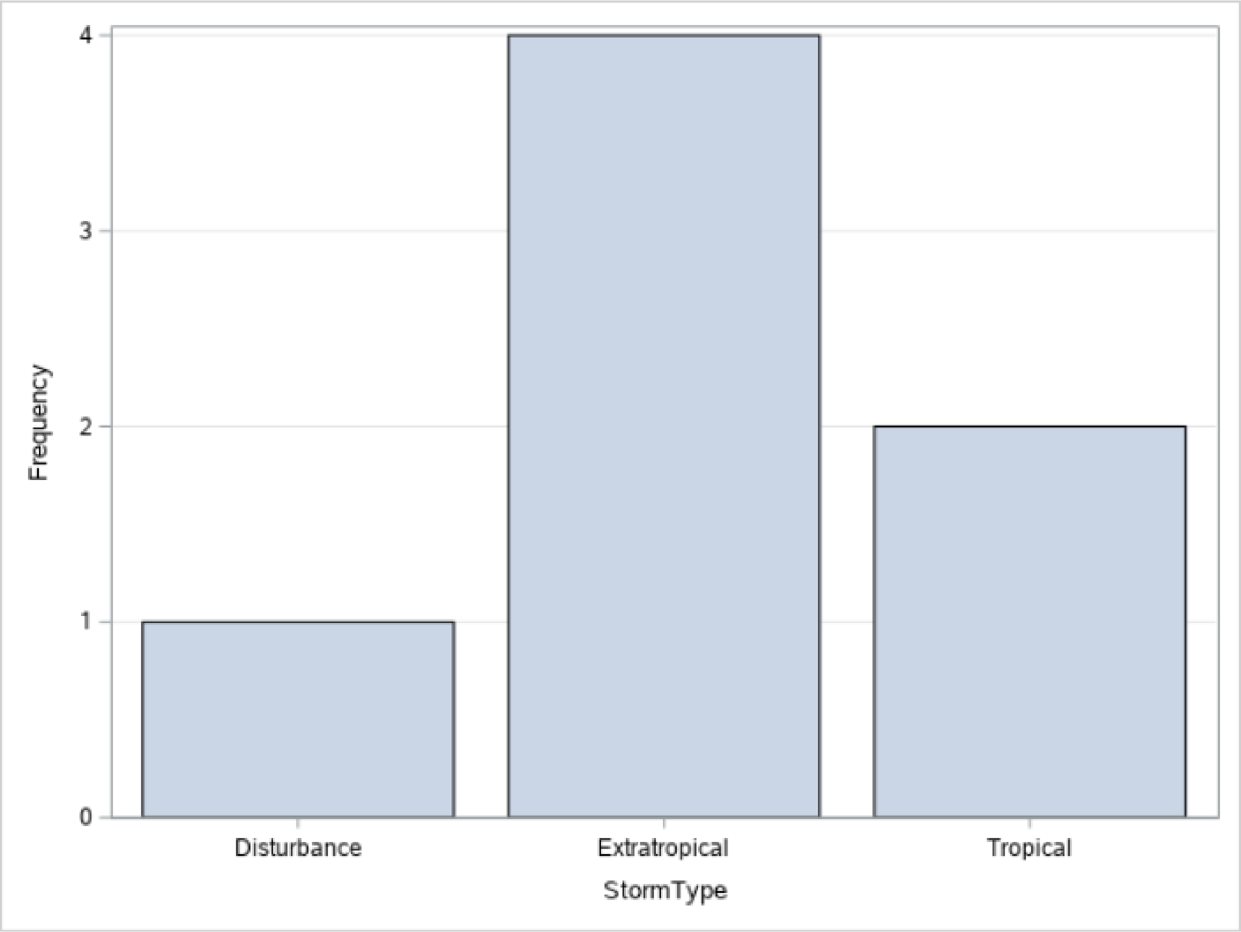
```
Modify the report code:
Replace all hardcoded values of 79 with a reference to AvgWind.
Replace all hardcoded text values of North Atlantic with a reference to BasinName.
Submit the modified code and verify that the report contains the same information as the report
generated in step b.
Modify the %LET statements to assign 2015 to Year and EP to BasinCode. Submit the entire program,
including the modified %LET statements and the PROC SQL step.
Review the log to ensure that there are no errors or warnings.
Verify that the report title is East Pacific Storms in 2015 Season Max Wind > Season Average of 92 MPH.
Which bar has the highest frequency?
*/
 /* Part a. */
%let year=2016;
%let basincode=NA;

proc sql noprint;
select round(mean(MaxWindMPH)) as AvgWind
        into :AvgWind trimmed
  from mc1.storm_final
  where Season=&year and Basin="&basincode";
select BasinName
        into :BasinName
        from mc1.storm_basin_codes
        where Basin="&basincode";
quit;

 /* Part b. */
title1 "&BasinName Basin Storms in &year Season";
title2 "Max Wind > Season Average of &AvgWind MPH";
```

```
proc print data=mc1.storm_final noobs;
        var Name StartDate EndDate MaxWindMPH MinPressure;
        where MaxWindMPH>&AvgWind and Season=&year and Basin="&basincode";
run;
title;

proc sgplot data=mc1.storm_final;
  where MaxWindMPH>&AvgWind and Season=&year and Basin="&basincode";
        vbar StormType;
        yaxis display=(noline) grid;
run;
```



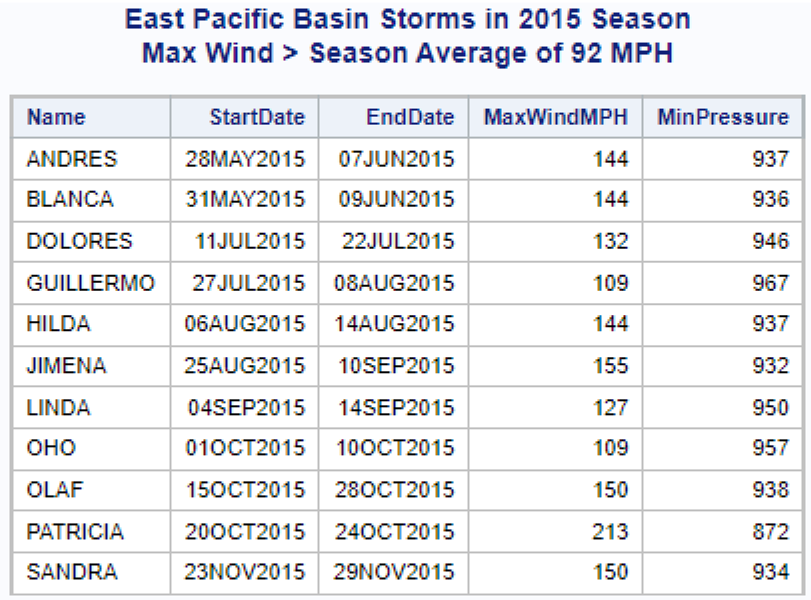
North Atlantic Basin Storms in 2016 Season
Max Wind > Season Average of 79 MPH

Name	StartDate	EndDate	MaxWindMPH	MinPressure
ALEX	07JAN2016	17JAN2016	86	978
EARL	02AUG2016	06AUG2016	86	979
GASTON	21AUG2016	03SEP2016	121	955
HERMINE	28AUG2016	08SEP2016	81	981
MATTHEW	28SEP2016	10OCT2016	167	934
NICOLE	04OCT2016	19OCT2016	138	950
OTTO	17NOV2016	25NOV2016	115	975



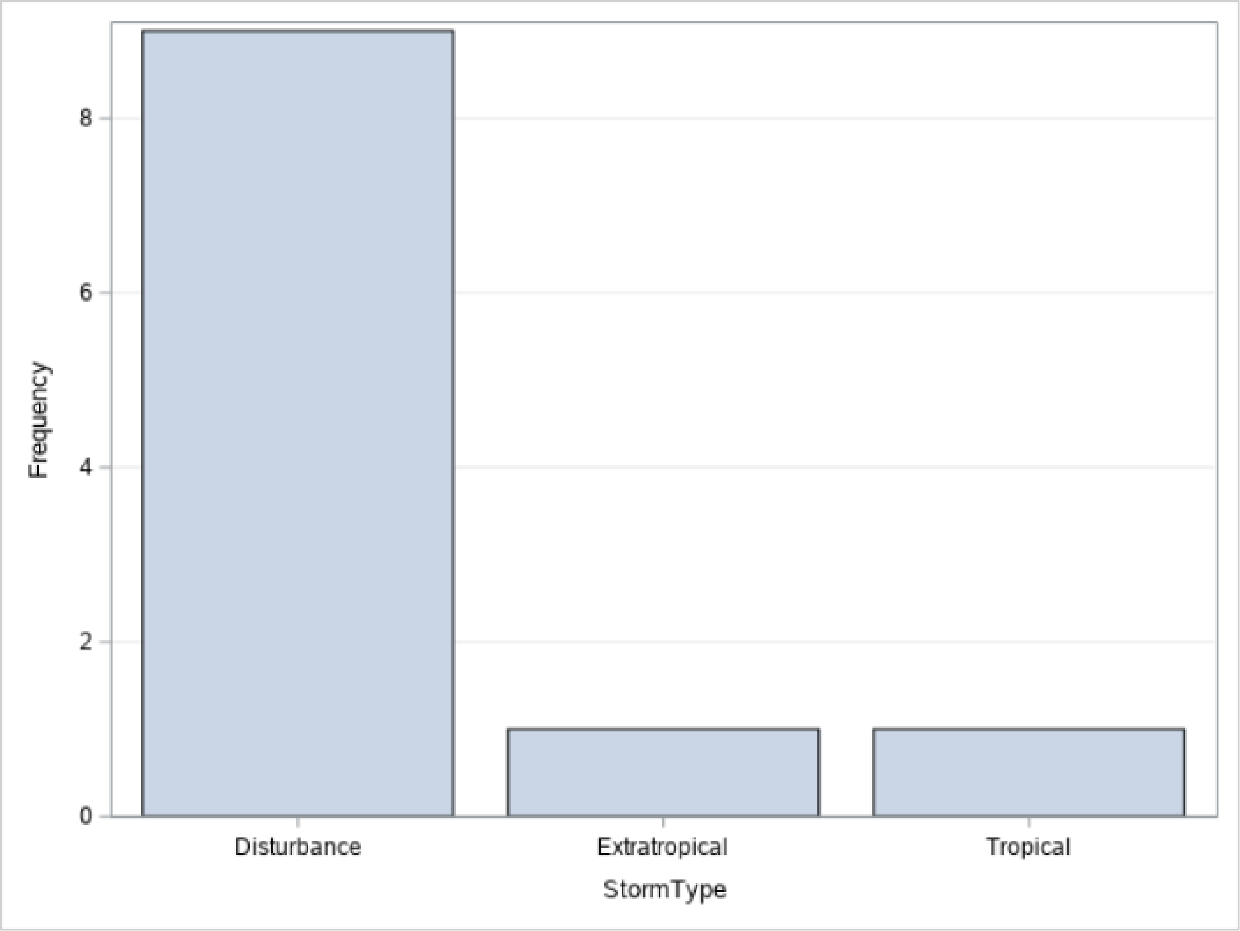
```
/* Part a. */

%let year=2015;

%let basincode=EP;


proc sql noprint;

select round(mean(MaxWindMPH)) as AvgWind

            into :AvgWind trimmed

   from mc1.storm_final

   where Season=&year and Basin="&basincode";

select BasinName

            into :BasinName

            from mc1.storm_basin_codes

            where Basin="&basincode";
```

```
quit;


 /* Part b. */
title1 "&BasinName Basin Storms in &year Season";
title2 "Max Wind > Season Average of &AvgWind MPH";
proc print data=mc1.storm_final noobs;
            var Name StartDate EndDate MaxWindMPH MinPressure;
            where MaxWindMPH>&AvgWind and Season=&year and Basin="&basincode";
run;
title;


proc sgplot data=mc1.storm_final;
   where MaxWindMPH>&AvgWind and Season=&year and Basin="&basincode";
            vbar StormType;
            yaxis display=(noline) grid;
run;
```



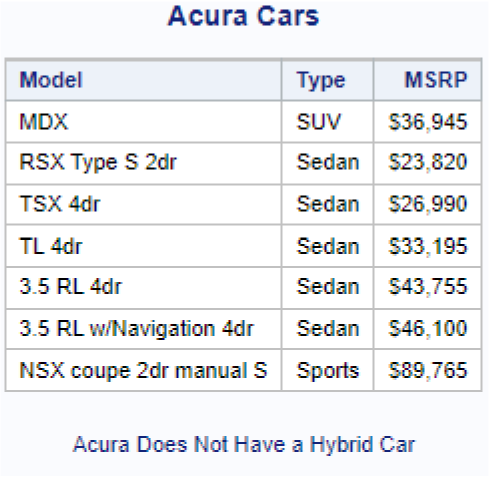
East Pacific Basin Storms in 2015 Season
Max Wind > Season Average of 92 MPH

Name	StartDate	EndDate	MaxWindMPH	MinPressure
ANDRES	28MAY2015	07JUN2015	144	937
BLANCA	31MAY2015	09JUN2015	144	936
DOLORES	11JUL2015	22JUL2015	132	946
GUILLERMO	27JUL2015	08AUG2015	109	967
HILDA	06AUG2015	14AUG2015	144	937
JIMENA	25AUG2015	10SEP2015	155	932
LINDA	04SEP2015	14SEP2015	127	950
OHO	01OCT2015	10OCT2015	109	957
OLAF	15OCT2015	28OCT2015	150	938
PATRICIA	20OCT2015	24OCT2015	213	872
SANDRA	23NOV2015	29NOV2015	150	934



```
********************************************************;
*  Activity 3.04                                *;
*  1) Run the program with Acura as the value of Make.    *;
*     First, examine the output data and confirm that the *;
*     value of HybridFlag is missing for the last row     *;
*     because there are no hybrid cars. Second, confirm   *;
*     that the footnote for the report is correct.        *;
*  2) Modify the %LET statement to assign Honda as the    *;
*     value of the Make macro variable. Run the program   *;
*     and view the output data. Confirm that the value of *;
*     HybridFlag is 1 for the last row.                   *;
*  3) Examine the report. Is the footnote correct?        *;
********************************************************;
```

```
%let make=Acura;

data &make(keep=Make Model Type MSRP HybridFlag);
   set sashelp.cars end=lastrow;
   where upcase(Make)="%upcase(&make)";
   retain HybridFlag;
   if Type="Hybrid" then HybridFlag=1;
   if lastrow then do;
      if HybridFlag=1 then do;
         %let foot=&make Offers Hybrid Cars;
      end;
      else do;
         %let foot=&make Does Not Have a Hybrid Car;
      end;
   end;
run;

title "&make Cars";
footnote "&foot";
proc print data=&make noobs;
           var Model Type MSRP;
run;
title;footnote;
```



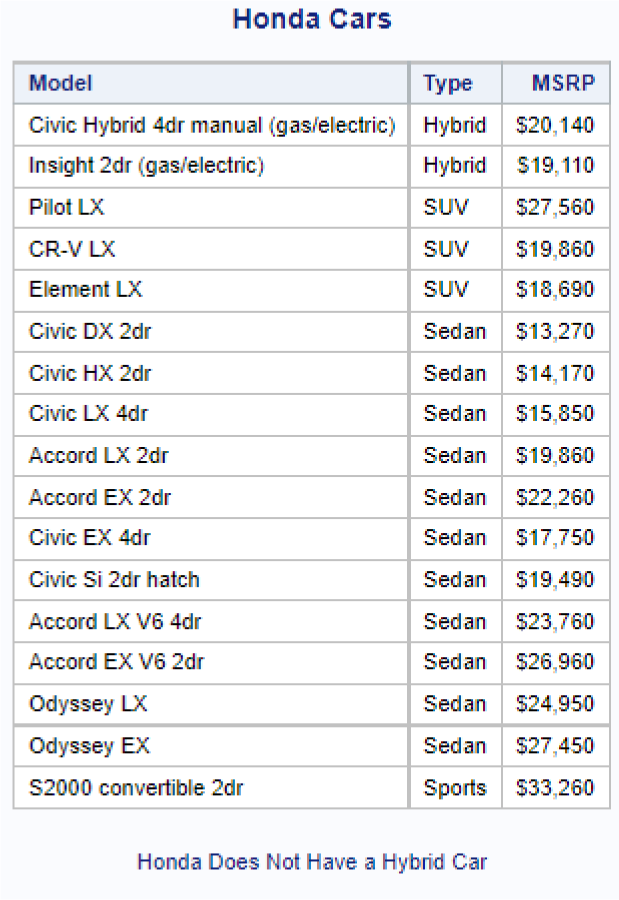
Acura Cars

Model	Type	MSRP
MDX	SUV	$36,945
RSX Type S 2dr	Sedan	$23,820
TSX 4dr	Sedan	$26,990
TL 4dr	Sedan	$33,195
3.5 RL 4dr	Sedan	$43,755
3.5 RL w/Navigation 4dr	Sedan	$46,100
NSX coupe 2dr manual S	Sports	$89,765

Acura Does Not Have a Hybrid Car

```
%let make=Honda;


data &make(keep=Make Model Type MSRP HybridFlag);
   set sashelp.cars end=lastrow;
   where upcase(Make)="%upcase(&make)";
   retain HybridFlag;
   if Type="Hybrid" then HybridFlag=1;
   if lastrow then do;
      if HybridFlag=1 then do;
         %let foot=&make Offers Hybrid Cars;
      end;
      else do;
         %let foot=&make Does Not Have a Hybrid Car;
      end;
   end;
run;


title "&make Cars";
footnote "&foot";
proc print data=&make noobs;
```

```
        var Model Type MSRP;
run;
title;footnote;
```



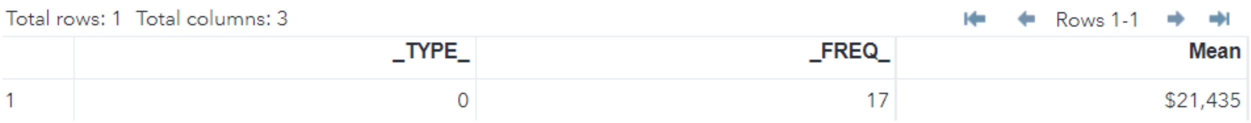
Honda Cars

Model	Type	MSRP
Civic Hybrid 4dr manual (gas/electric)	Hybrid	$20,140
Insight 2dr (gas/electric)	Hybrid	$19,110
Pilot LX	SUV	$27,560
CR-V LX	SUV	$19,860
Element LX	SUV	$18,690
Civic DX 2dr	Sedan	$13,270
Civic HX 2dr	Sedan	$14,170
Civic LX 4dr	Sedan	$15,850
Accord LX 2dr	Sedan	$19,860
Accord EX 2dr	Sedan	$22,260
Civic EX 4dr	Sedan	$17,750
Civic Si 2dr hatch	Sedan	$19,490
Accord LX V6 4dr	Sedan	$23,760
Accord EX V6 2dr	Sedan	$26,960
Odyssey LX	Sedan	$24,950
Odyssey EX	Sedan	$27,450
S2000 convertible 2dr	Sports	$33,260

Honda Does Not Have a Hybrid Car

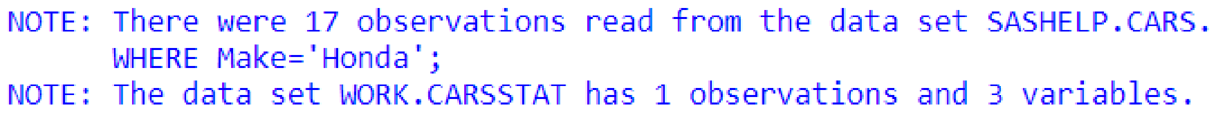
```
*****************************************************************;
*  Activity 3.06                              *;
*  1) Highlight the %LET statement and PROC MEANS step    *;
*     and run the selection. Examine the output data.    *;
*  2) Complete the CALL SYMPUTX statement to create a     *;
*     macro variable named AvgMSRP and load the value of  *;
*     the Mean column from the CarsStat table.          *;
*  3) Run the completed program. What is the value of the *;
*     second title?                           *;
*****************************************************************;
```

```
%let make=Honda;


proc means data=sashelp.cars noprint maxdec=0;
        where Make="&make";
        var MSRP;
        output out=CarsStat Mean=Mean;
run;
```



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

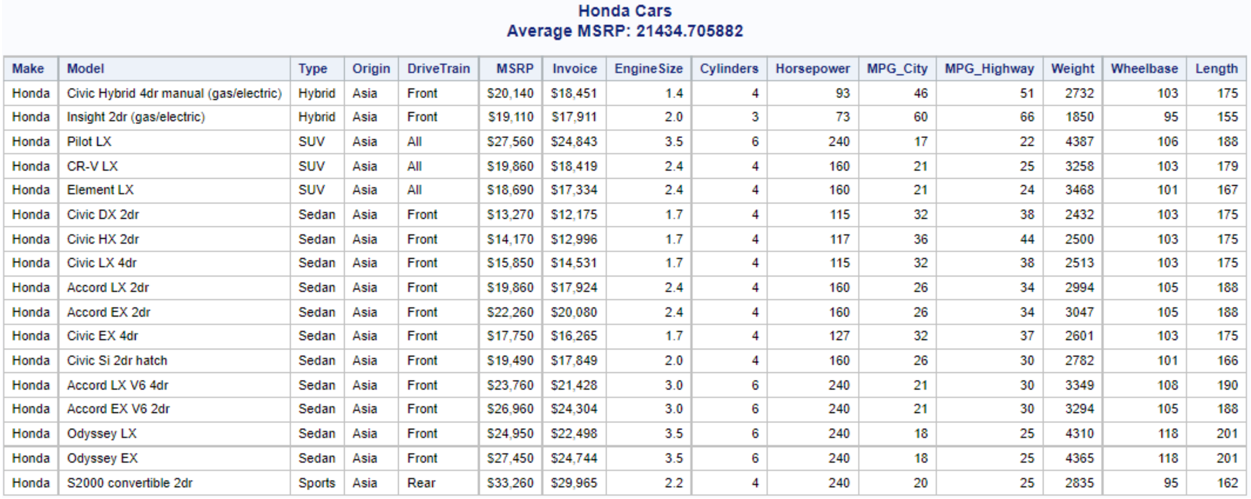
	TYPE	_FREQ_	Mean
1	0	17	$21,435



```
NOTE: There were 17 observations read from the data set SASHELP.CARS.
      WHERE Make='Honda';
NOTE: The data set WORK.CARSSTAT has 1 observations and 3 variables.
```

```
%let make=Honda;


proc means data=sashelp.cars noprint maxdec=0;
        where Make="&make";
        var MSRP;
        output out=CarsStat Mean=Mean;
run;


/* Complete the CALL SYMPUTX statement */
data _null_;
        set CarsStat;
        call symputx("AvgMSRP", Mean);
run;


title "&make Cars";
title2 "Average MSRP: &avgmsrp";
```

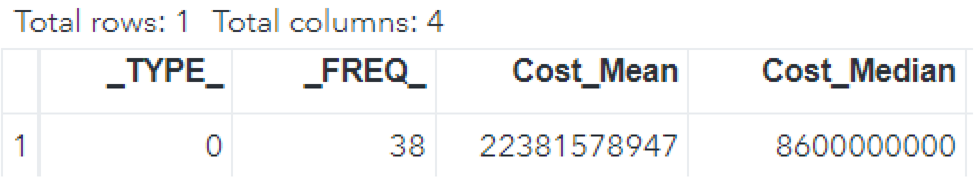
```
proc print data=sashelp.cars noobs;
        where Make="&make";
run;
title;
```



Honda Cars
Average MSRP: 21434.705882

Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
Honda	Civic Hybrid 4dr manual (gas/electric)	Hybrid	Asia	Front	$20,140	$18,451	1.4	4	93	46	51	2732	103	175
Honda	Insight 2dr (gas/electric)	Hybrid	Asia	Front	$19,110	$17,911	2.0	3	73	60	66	1850	95	155
Honda	Pilot LX	SUV	Asia	All	$27,560	$24,843	3.5	6	240	17	22	4387	106	188
Honda	CR-V LX	SUV	Asia	All	$19,860	$18,419	2.4	4	160	21	25	3258	103	179
Honda	Element LX	SUV	Asia	All	$18,690	$17,334	2.4	4	160	21	24	3468	101	167
Honda	Civic DX 2dr	Sedan	Asia	Front	$13,270	$12,175	1.7	4	115	32	38	2432	103	175
Honda	Civic HX 2dr	Sedan	Asia	Front	$14,170	$12,996	1.7	4	117	36	44	2500	103	175
Honda	Civic LX 4dr	Sedan	Asia	Front	$15,850	$14,531	1.7	4	115	32	38	2513	103	175
Honda	Accord LX 2dr	Sedan	Asia	Front	$19,860	$17,924	2.4	4	160	26	34	2994	105	188
Honda	Accord EX 2dr	Sedan	Asia	Front	$22,260	$20,080	2.4	4	160	26	34	3047	105	188
Honda	Civic EX 4dr	Sedan	Asia	Front	$17,750	$16,265	1.7	4	127	32	37	2601	103	175
Honda	Civic Si 2dr hatch	Sedan	Asia	Front	$19,490	$17,849	2.0	4	160	26	30	2782	101	166
Honda	Accord LX V6 4dr	Sedan	Asia	Front	$23,760	$21,428	3.0	6	240	21	30	3349	108	190
Honda	Accord EX V6 2dr	Sedan	Asia	Front	$26,960	$24,304	3.0	6	240	21	30	3294	105	188
Honda	Odyssey LX	Sedan	Asia	Front	$24,950	$22,498	3.5	6	240	18	25	4310	118	201
Honda	Odyssey EX	Sedan	Asia	Front	$27,450	$24,744	3.5	6	240	18	25	4365	118	201
Honda	S2000 convertible 2dr	Sports	Asia	Rear	$33,260	$29,965	2.2	4	240	20	25	2835	95	162

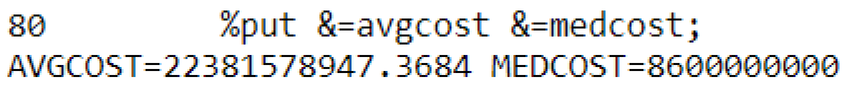
```
/****************************************************
* Using the DATA Step to Create Macro Variables: Demo *
****************************************************/


/* Section 1 */
proc means data=mc1.storm_damage noprint;
   var Cost;
   output out=work.sumdata mean= median= /autoname;
run;
```



Total rows: 1 Total columns: 4

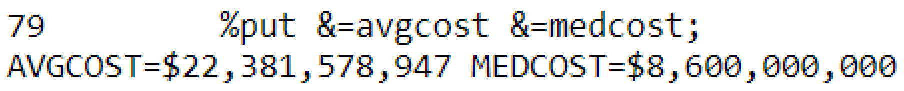
	TYPE	_FREQ_	Cost_Mean	Cost_Median
1	0	38	22381578947	8600000000

```
data _null_;
   set sumdata;
   /*insert CALL SYMPUTX statements */
  call symputx("avgcost", cost_mean);
  call symputx("medcost", cost_median);
run;


%put &=avgcost &=medcost;
```



```
80          %put &=avgcost &=medcost;
AVGCOST=22381578947.3684 MEDCOST=8600000000
```

```
data _null_;
   set sumdata;
   /*insert CALL SYMPUTX statements */
  call symputx("avgcost", put(cost_mean, dollar20.));
  call symputx("medcost", put(cost_median, dollar20.));
run;
%put &=avgcost &=medcost;
```



```
79          %put &=avgcost &=medcost;
AVGCOST=$22,381,578,947 MEDCOST=$8,600,000,000
```

```
/**********************************************************
* Using the DATA Step to Create Maro Variables: Practice #6 *
**********************************************************/
```

/*Level 1 Practice: Creating Macro Variables with the SYMPUTX Routine

Reminder: If you restarted your SAS session,

you must submit the libname.sas program in the EMC1V2 folder to access your practice files.

Open m103p06.sas from the practices folder. Review and submit the program.

Verify that the report title is New Staff: Administration Department and that the sum of Salary is $221,618.

Add a %LET statement to assign the value Administration to a new macro variable named dept, and

replace every occurrence of Administration with a reference to dept.

Submit the modified program and verify that the title and report are the same as the first code you submitted.

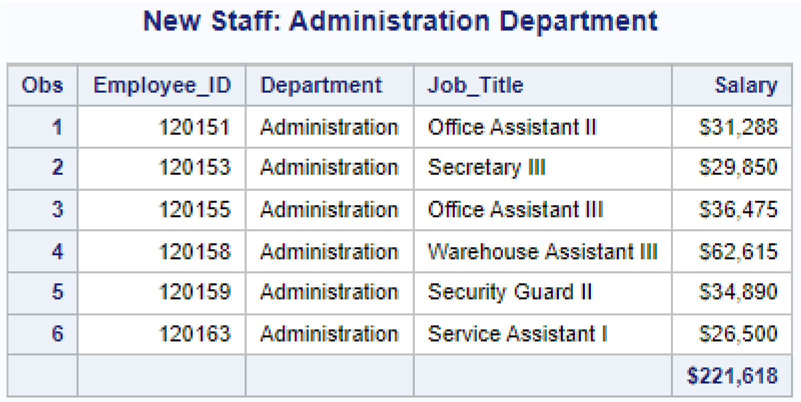
Change the value of dept to Sales and submit the program. Verify that the report title is New Staff: Sales Department.

What is the sum of Salary?

Note: Copy and paste or type the value as it is shown.

*/

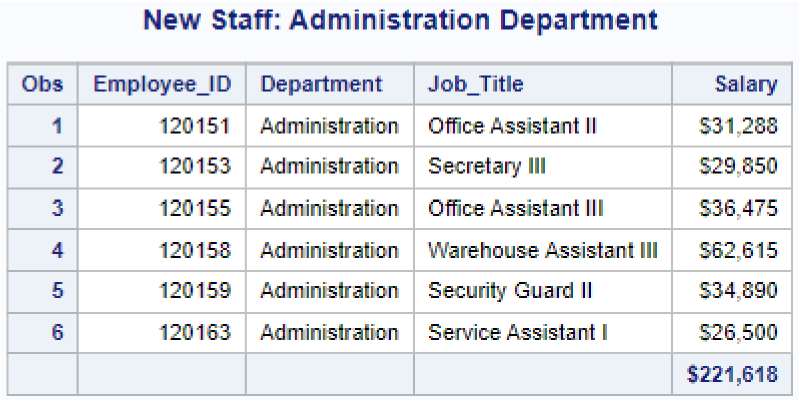
```
data staff;
        keep Employee_ID Department Job_Title Salary;
        set mc1.newhires;
        where Department="Administration";
run;

title "New Staff: Administration Department";
proc print data=staff;
        sum salary;
run;
title;
```



New Staff: Administration Department

Obs	Employee_ID	Department	Job_Title	Salary
1	120151	Administration	Office Assistant II	$31,288
2	120153	Administration	Secretary III	$29,850
3	120155	Administration	Office Assistant III	$36,475
4	120158	Administration	Warehouse Assistant III	$62,615
5	120159	Administration	Security Guard II	$34,890
6	120163	Administration	Service Assistant I	$26,500
				$221,618

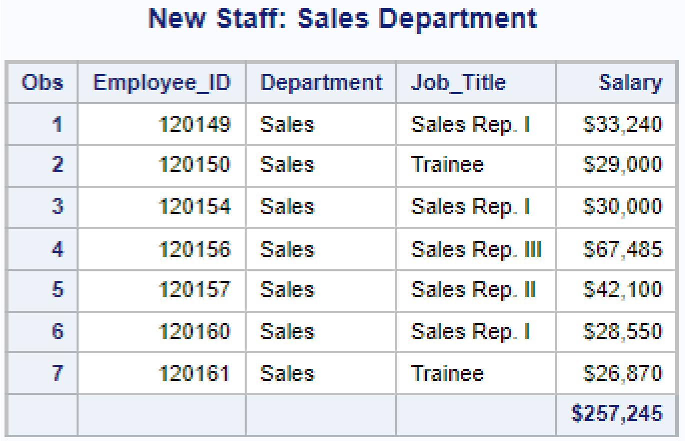
```
%let dept=Administration;
data staff;
        keep Employee_ID Department Job_Title Salary;
        set mc1.newhires;
        where Department=”&dept”;
run;

title "New Staff: &dept Department";
proc print data=staff;
        sum salary;
run;
title;
```



New Staff: Administration Department

Obs	Employee_ID	Department	Job_Title	Salary
1	120151	Administration	Office Assistant II	$31,288
2	120153	Administration	Secretary III	$29,850
3	120155	Administration	Office Assistant III	$36,475
4	120158	Administration	Warehouse Assistant III	$62,615
5	120159	Administration	Security Guard II	$34,890
6	120163	Administration	Service Assistant I	$26,500
				$221,618

```
%let dept=Sales;
data staff;
        keep Employee_ID Department Job_Title Salary;
        set mc1.newhires;
        where Department="&dept";
run;

title "New Staff: &dept Department";
proc print data=staff;
        sum salary;
run;
title;
```



New Staff: Sales Department

Obs	Employee_ID	Department	Job_Title	Salary
1	120149	Sales	Sales Rep. I	$33,240
2	120150	Sales	Trainee	$29,000
3	120154	Sales	Sales Rep. I	$30,000
4	120156	Sales	Sales Rep. III	$67,485
5	120157	Sales	Sales Rep. II	$42,100
6	120160	Sales	Sales Rep. I	$28,550
7	120161	Sales	Trainee	$26,870
				$257,245

/*Modify the DATA step to create a macro variable named avg to store the average salary on the last iteration.

Hint: Use the PUT function and the DOLLAR9. format when assigning the value to avg.

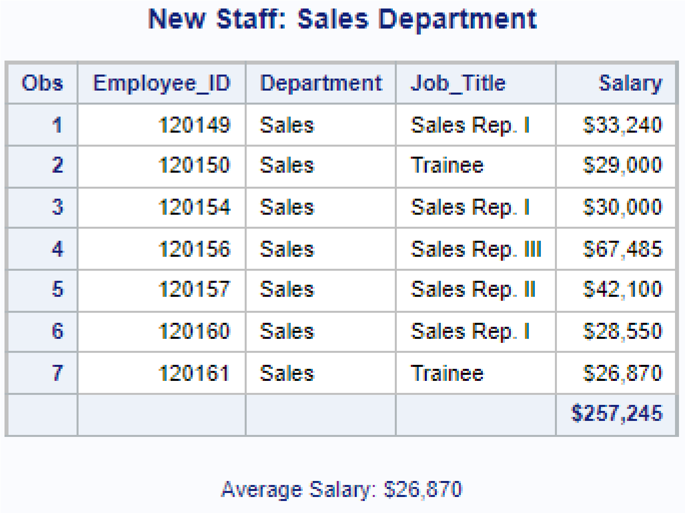
Add a FOOTNOTE statement before the PROC PRINT step to display the value of avg as shown here:
Average Salary: $xx,xxx

Submit the program and review the results.

What text is displayed in the footnote?

*/

```
%let dept=Sales;

data staff;

        keep Employee_ID Department Job_Title Salary;

        set mc1.newhires;

        where Department="&dept";

        call symputx("avg", put(mean(salary), dollar9.));

run;


title "New Staff: &dept Department";

footnote "Average Salary: &avg";

proc print data=staff;

        sum salary;

run;

title;footnote;
```

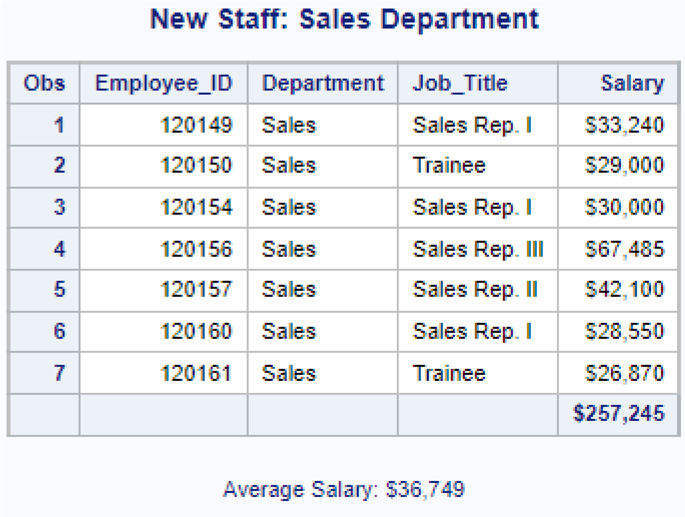


New Staff: Sales Department

Obs	Employee_ID	Department	Job_Title	Salary
1	120149	Sales	Sales Rep. I	$33,240
2	120150	Sales	Trainee	$29,000
3	120154	Sales	Sales Rep. I	$30,000
4	120156	Sales	Sales Rep. III	$67,485
5	120157	Sales	Sales Rep. II	$42,100
6	120160	Sales	Sales Rep. I	$28,550
7	120161	Sales	Trainee	$26,870
				$257,245

Average Salary: $26,870

```
*Solution;

%let dept=Sales;

data staff;

   keep Employee_ID Department Job_Title Salary;

   set mc1.newhires end=last;

   where Department="&dept";

   total+salary;

   if last=1 then

      call symputx("avg",put(total/_n_,dollar9.));

run;


footnote "Average Salary: &avg";

title "New Staff: &dept Department";

proc print data=staff;

   sum salary;

run;

title;footnote;
```



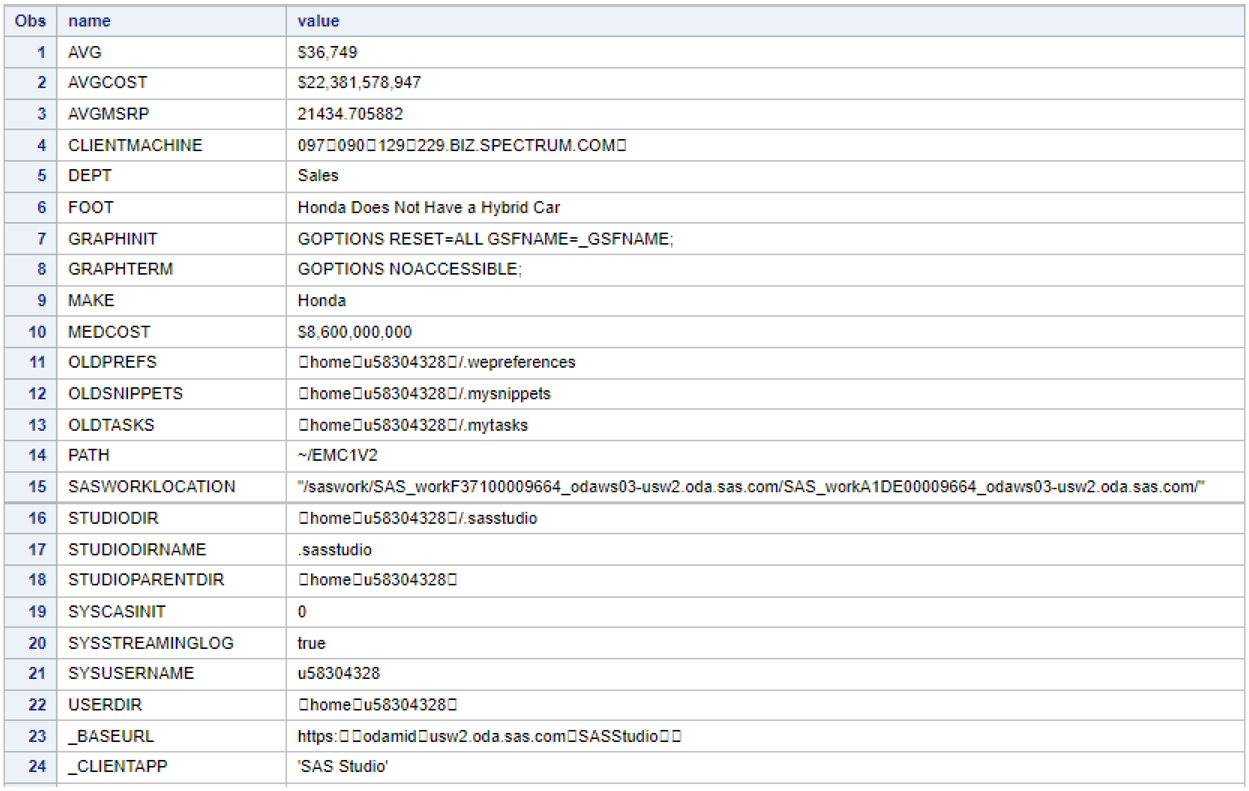
New Staff: Sales Department

Obs	Employee_ID	Department	Job_Title	Salary
1	120149	Sales	Sales Rep. I	$33,240
2	120150	Sales	Trainee	$29,000
3	120154	Sales	Sales Rep. I	$30,000
4	120156	Sales	Sales Rep. III	$67,485
5	120157	Sales	Sales Rep. II	$42,100
6	120160	Sales	Sales Rep. I	$28,550
7	120161	Sales	Trainee	$26,870
				$257,245

Average Salary: $36,749

```
/*********************************************************

* Using the DATA Step to Create Macro Variables: Practice #7 *

*********************************************************/

/*Level 2 Practice: Using a DATA _NULL_ Step to Create a Series of Macro Variables

Reminder: If you restarted your SAS session,

you must submit the libname.sas program in the EMC1V2 folder to access your practice files.

Open m103p07.sas from the practices folder and review the program.

The sashelp.vmacro data source is a dynamic view that contains the name and value of all macro variables

in the current SAS session. Run the program and view the report.

Insert a DATA _NULL_ step before the PROC PRINT step to create a series of macro variables using

the values stored in the mc1.Storm_Ocean_Codes table. Use the values in Ocean to name the macro variables.

Use the values in OceanName as the macro variable values.

Modify the PROC PRINT step to display only macro variables with one-character names.

Submit the entire program and verify that five macro variables were created.

What is the value of the macro variable, S?

*/


proc print data=sashelp.vmacro;
```

```
        var Name Value;
run;
```



Obs	name	value
1	AVG	$36,749
2	AVGCOST	$22,381,578,947
3	AVGMSRP	21434.705882
4	CLIENTMACHINE	097□090□129□229.BIZ.SPECTRUM.COM□
5	DEPT	Sales
6	FOOT	Honda Does Not Have a Hybrid Car
7	GRAPHINIT	GOPTIONS RESET=ALL GSFNAME=_GSFNAME;
8	GRAPHTERM	GOPTIONS NOACCESSIBLE;
9	MAKE	Honda
10	MEDCOST	$8,600,000,000
11	OLDPREFS	□home□u58304328□/.wepreferences
12	OLDSNIPPETS	□home□u58304328□/.mysnippets
13	OLDTASKS	□home□u58304328□/.mytasks
14	PATH	~/EMC1V2
15	SASWORKLOCATION	"/saswork/SAS_workF37100009664_odaws03-usw2.oda.sas.com/SAS_workA1DE00009664_odaws03-usw2.oda.sas.com/"
16	STUDIODIR	□home□u58304328□/.sasstudio
17	STUDIODIRNAME	.sasstudio
18	STUDIOPARENTDIR	□home□u58304328□
19	SYSCASINIT	0
20	SYSSTREAMINGLOG	true
21	SYSUSERNAME	u58304328
22	USERDIR	□home□u58304328□
23	_BASEURL	https:□□odamid□usw2.oda.sas.com□SASStudio□□
24	_CLIENTAPP	'SAS Studio'

/***

* Using the DATA Step to Create Macro Variables: Practice #7 *

***/

/*Level 2 Practice: Using a DATA _NULL_ Step to Create a Series of Macro Variables

Reminder: If you restarted your SAS session,

you must submit the libname.sas program in the EMC1V2 folder to access your practice files.

Open m103p07.sas from the practices folder and review the program.

The sashelp.vmacro data source is a dynamic view that contains the name and value of all macro variables

in the current SAS session. Run the program and view the report.

Insert a DATA _NULL_ step before the PROC PRINT step to create a series of macro variables using

the values stored in the mc1.Storm_Ocean_Codes table. Use the values in Ocean to name the macro variables.

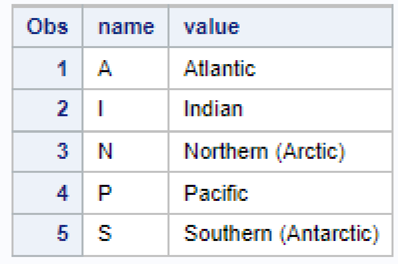
Use the values in OceanName as the macro variable values.

Modify the PROC PRINT step to display only macro variables with one-character names.

Submit the entire program and verify that five macro variables were created.

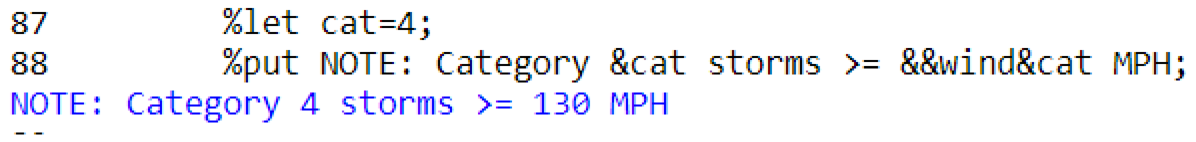
What is the value of the macro variable, S?

```
*/
data _null_;
        set mc1.storm_ocean_codes;
        call symputx(Ocean, OceanName);
run;

proc print data=sashelp.vmacro;
        var Name Value;
        where name like "_";
run;
```



Obs	name	value
1	A	Atlantic
2	I	Indian
3	N	Northern (Arctic)
4	P	Pacific
5	S	Southern (Antarctic)

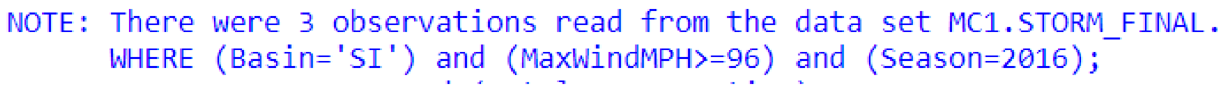
```
*******************************************************;
*  Activity 3.07                                *;
*  1) Notice that wind values have been replaced with    *;
*     &wind&cat. Run the program and review the log.     *;
*  2) Change &wind&cat to &&wind&cat. Run the program and *;
*     review the log. Does the program run successfully?  *;
*******************************************************;

proc sql noprint;
select MinWind
   into :wind1-
   from mc1.storm_cat;
quit;


%let cat=4;
%put NOTE: Category &cat storms >= &&wind&cat MPH;
```



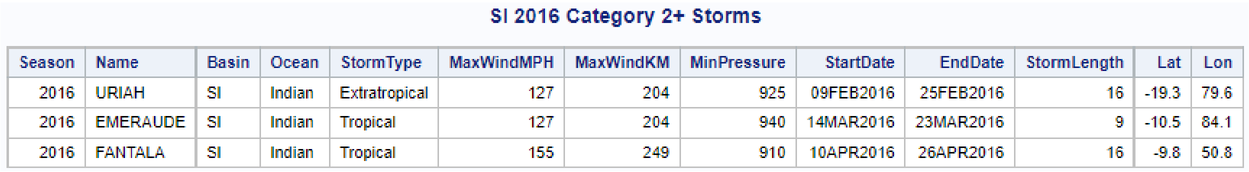
```
87         %let cat=4;
88         %put NOTE: Category &cat storms >= &&wind&cat MPH;
NOTE: Category 4 storms >= 130 MPH
--
```

```
/**********************************************
* Indirect References to Macro Variables: Demo *
**********************************************/


%let year=2016;
%let cat=2;
%let basin=SI;


proc sql noprint;
select MinWind
   into :wind1-
   from mc1.storm_cat;
quit;


data _null_;
   set mc1.storm_basin_codes;
   call symputx(Basin, BasinName);
run;


title1 "&basin &year Category &cat+ Storms";
proc print data=mc1.storm_final noobs;
```

```
          where Basin="&basin" and

                    MaxWindMPH>=&&wind&cat and

                    Season=&year;

run;

title;
```



```
NOTE: There were 3 observations read from the data set MC1.STORM_FINAL.
      WHERE (Basin='SI') and (MaxWindMPH>=96) and (Season=2016);
```

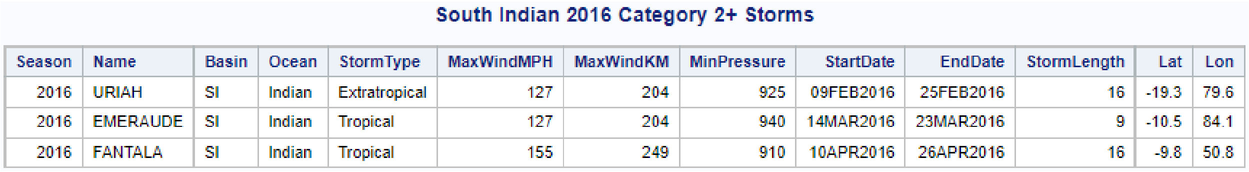


SI 2016 Category 2+ Storms

Season	Name	Basin	Ocean	StormType	MaxWindMPH	MaxWindKM	MinPressure	StartDate	EndDate	StormLength	Lat	Lon
2016	URIAH	SI	Indian	Extratropical	127	204	925	09FEB2016	25FEB2016	16	-19.3	79.6
2016	EMERAUDE	SI	Indian	Tropical	127	204	940	14MAR2016	23MAR2016	9	-10.5	84.1
2016	FANTALA	SI	Indian	Tropical	155	249	910	10APR2016	26APR2016	16	-9.8	50.8

```
/***********************************************

* Indirect References to Macro Variables: Demo *

***********************************************/


%let year=2016;

%let cat=2;

%let basin=SI;


proc sql noprint;

select MinWind

   into :wind1-

   from mc1.storm_cat;

quit;


data _null_;

   set mc1.storm_basin_codes;

   call symputx(Basin, BasinName);

run;
```

```
title1 "&&&basin &year Category &cat+ Storms";
proc print data=mc1.storm_final noobs;
        where Basin="&basin" and
                 MaxWindMPH>=&&wind&cat and
                 Season=&year;
run;
title;
```

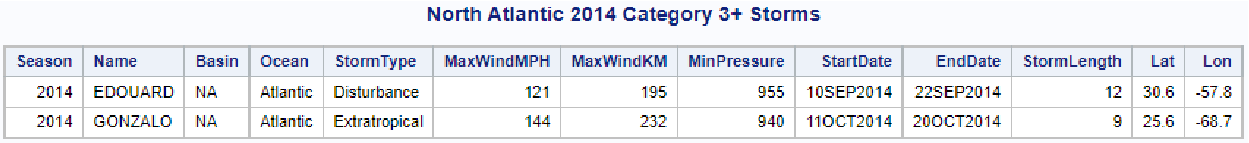


South Indian 2016 Category 2+ Storms

Season	Name	Basin	Ocean	StormType	MaxWindMPH	MaxWindKM	MinPressure	StartDate	EndDate	StormLength	Lat	Lon
2016	URIAH	SI	Indian	Extratropical	127	204	925	09FEB2016	25FEB2016	16	-19.3	79.6
2016	EMERAUDE	SI	Indian	Tropical	127	204	940	14MAR2016	23MAR2016	9	-10.5	84.1
2016	FANTALA	SI	Indian	Tropical	155	249	910	10APR2016	26APR2016	16	-9.8	50.8

```
/********************************************
* Indirect References to Macro Variables: Demo *
********************************************/


%let year=2014;
%let cat=3;
%let basin=NA;


proc sql noprint;
select MinWind
  into :wind1-
  from mc1.storm_cat;
quit;


data _null_;
  set mc1.storm_basin_codes;
  call symputx(Basin, BasinName);
run;
```

```
title1 "&&&basin &year Category &cat+ Storms";

proc print data=mc1.storm_final noobs;

        where Basin="&basin" and

                 MaxWindMPH>=&&wind&cat and

                 Season=&year;

run;

title;
```

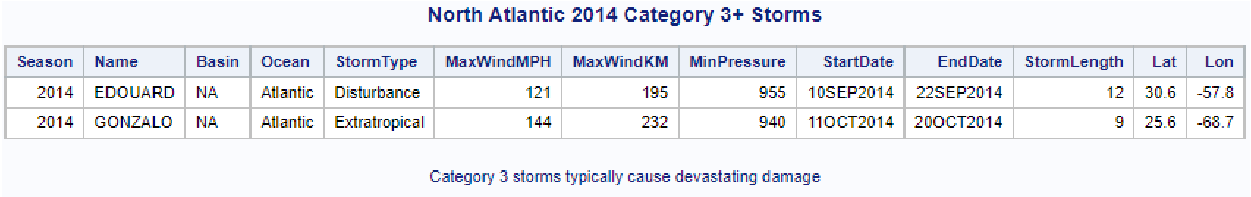


North Atlantic 2014 Category 3+ Storms

Season	Name	Basin	Ocean	StormType	MaxWindMPH	MaxWindKM	MinPressure	StartDate	EndDate	StormLength	Lat	Lon
2014	EDOUARD	NA	Atlantic	Disturbance	121	195	955	10SEP2014	22SEP2014	12	30.6	-57.8
2014	GONZALO	NA	Atlantic	Extratropical	144	232	940	11OCT2014	20OCT2014	9	25.6	-68.7

```
/*********************************************

* Indirect References to Macro Variables: Demo *

*********************************************/


%let year=2014;

%let cat=3;

%let basin=NA;


proc sql noprint;

select MinWind, Damage

   into :wind1-, :damage1-

   from mc1.storm_cat;

quit;


data _null_;

   set mc1.storm_basin_codes;

   call symputx(Basin, BasinName);

run;
```

```
title1 "&&&basin &year Category &cat+ Storms";

footnote "Category &cat storms typically cause %lowcase(&&damage&cat)";

proc print data=mc1.storm_final noobs;

           where Basin="&basin" and

                      MaxWindMPH>=&&wind&cat and

                      Season=&year;

run;

title; footnote;
```

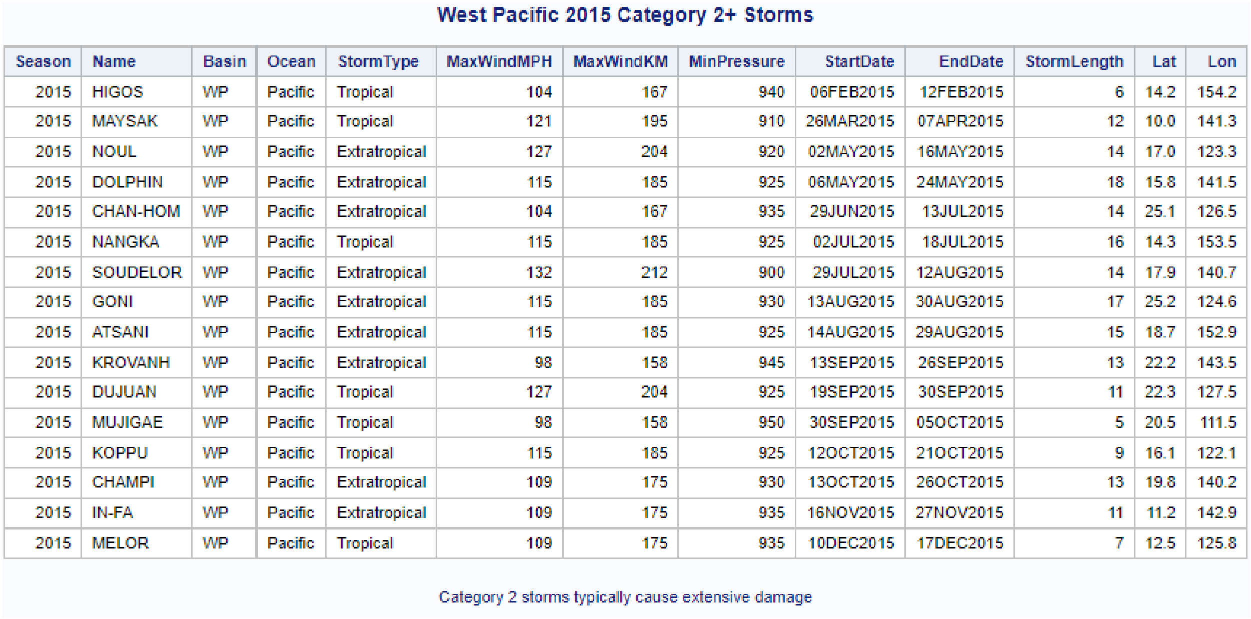


North Atlantic 2014 Category 3+ Storms

Season	Name	Basin	Ocean	StormType	MaxWindMPH	MaxWindKM	MinPressure	StartDate	EndDate	StormLength	Lat	Lon
2014	EDOUARD	NA	Atlantic	Disturbance	121	195	955	10SEP2014	22SEP2014	12	30.6	-57.8
2014	GONZALO	NA	Atlantic	Extratropical	144	232	940	11OCT2014	20OCT2014	9	25.6	-68.7

Category 3 storms typically cause devastating damage

```
/**********************************************

* Indirect References to Macro Variables: Demo *

**********************************************/


%let year=2015;

%let cat=2;

%let basin=WP;


proc sql noprint;

select MinWind, Damage

   into :wind1-, :damage1-

   from mc1.storm_cat;

quit;


data _null_;

   set mc1.storm_basin_codes;
   call symputx(Basin, BasinName);
run;

title1 "&&&basin &year Category &cat+ Storms";
footnote "Category &cat storms typically cause %lowcase(&&damage&cat)";
proc print data=mc1.storm_final noobs;
        where Basin="&basin" and
              MaxWindMPH>=&&wind&cat and
              Season=&year;
run;
title; footnote;
```



West Pacific 2015 Category 2+ Storms

Season	Name	Basin	Ocean	StormType	MaxWindMPH	MaxWindKM	MinPressure	StartDate	EndDate	StormLength	Lat	Lon
2015	HIGOS	WP	Pacific	Tropical	104	167	940	06FEB2015	12FEB2015	6	14.2	154.2
2015	MAYSAK	WP	Pacific	Tropical	121	195	910	26MAR2015	07APR2015	12	10.0	141.3
2015	NOUL	WP	Pacific	Extratropical	127	204	920	02MAY2015	16MAY2015	14	17.0	123.3
2015	DOLPHIN	WP	Pacific	Extratropical	115	185	925	06MAY2015	24MAY2015	18	15.8	141.5
2015	CHAN-HOM	WP	Pacific	Extratropical	104	167	935	29JUN2015	13JUL2015	14	25.1	126.5
2015	NANGKA	WP	Pacific	Tropical	115	185	925	02JUL2015	18JUL2015	16	14.3	153.5
2015	SOUDELOR	WP	Pacific	Extratropical	132	212	900	29JUL2015	12AUG2015	14	17.9	140.7
2015	GONI	WP	Pacific	Extratropical	115	185	930	13AUG2015	30AUG2015	17	25.2	124.6
2015	ATSANI	WP	Pacific	Extratropical	115	185	925	14AUG2015	29AUG2015	15	18.7	152.9
2015	KROVANH	WP	Pacific	Extratropical	98	158	945	13SEP2015	26SEP2015	13	22.2	143.5
2015	DUJUAN	WP	Pacific	Tropical	127	204	925	19SEP2015	30SEP2015	11	22.3	127.5
2015	MUJIGAE	WP	Pacific	Tropical	98	158	950	30SEP2015	05OCT2015	5	20.5	111.5
2015	KOPPU	WP	Pacific	Tropical	115	185	925	12OCT2015	21OCT2015	9	16.1	122.1
2015	CHAMPI	WP	Pacific	Extratropical	109	175	930	13OCT2015	26OCT2015	13	19.8	140.2
2015	IN-FA	WP	Pacific	Extratropical	109	175	935	16NOV2015	27NOV2015	11	11.2	142.9
2015	MELOR	WP	Pacific	Tropical	109	175	935	10DEC2015	17DEC2015	7	12.5	125.8

Category 2 storms typically cause extensive damage

```
/*****************************************************
* Indirect References to Macro Variables: Practice #8 *
*****************************************************/
/*Level 1 Practice: Using Indirect References to Macro Variables
Reminder: If you restarted your SAS session, you must submit the libname.sas program in the EMC1V2 folder
to access your practice files.
```

Open m103p08.sas from the practices folder. Submit the code and review the results.

Explore the mc1.order_type_codes table.

The Order_Type_Code column contains the values 1, 2, and 3, and

the Order_Type column identifies the type of order associated with the code. Close the table.

At the top of the program, insert an SQL query to access mc1.order_type_codes and

create a series of macro variables named Type1, Type2, and Type3.

Assign the value of the Order_Type variable associated with each order type.

Suppress the PROC SQL output, and add a %PUT statement to write the values of Type1, Type2, and Type3 to the log.

Submit the PROC SQL step and the %PUT statement. Review the log.

Verify that the log contains TYPE1=Retail Store TYPE2=Catalog TYPE3=Internet.

Modify the TITLE statement to use an indirect macro variable reference to the Type variable that

corresponds to the value of code.

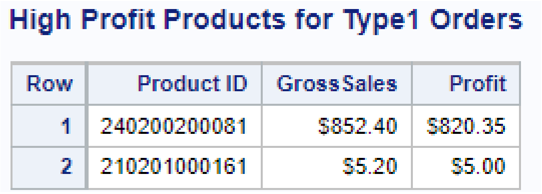
Submit the entire program and review the results. Verify that the report contains the same two rows

generated by the original report and that the title is High Profit Products for Retail Store Orders.

Modify the %LET statement to assign a value of 3 to the macro variable, code. Resubmit the program and review the results.

What is the report title?

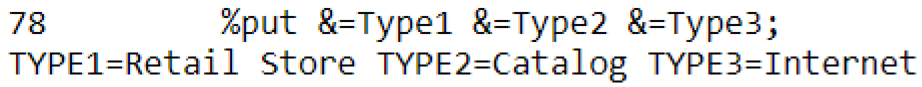
```
*/

%let code=1;

title "High Profit Products for Type&code Orders";

proc sql number;

select Product_ID format=z12.,

      Sum(Total_Retail_Price) format=dollar10.2 as GrossSales,

      Sum(Total_Retail_Price-CostPrice_Per_Unit) format=dollar10.2 as Profit

   from mc1.orders

   where Order_Type=&code

   group by Product_ID

   having profit /grosssales > .95
   order by Profit desc;
quit;
title;
```



High Profit Products for Type1 Orders

Row	Product ID	Gross Sales	Profit
1	240200200081	$852.40	$820.35
2	210201000161	$5.20	$5.00

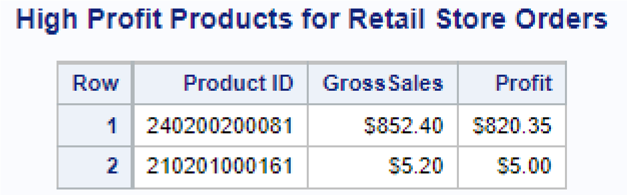
```
proc sql noprint;
select Order_Type
          into :Type1-
          from mc1.order_type_codes;
quit;
%put &=Type1 &=Type2 &=Type3;
```



```
78          %put &=Type1 &=Type2 &=Type3;
TYPE1=Retail Store TYPE2=Catalog TYPE3=Internet
```

```
proc sql noprint;
select Order_Type
          into :Type1-
          from mc1.order_type_codes;
quit;
%put &=Type1 &=Type2 &=Type3;


%let code=1;
title "High Profit Products for &&Type&code Orders";
proc sql number;
select Product_ID format=z12.,
     Sum(Total_Retail_Price) format=dollar10.2 as GrossSales,
     Sum(Total_Retail_Price-CostPrice_Per_Unit) format=dollar10.2 as Profit
   from mc1.orders
```

```
  where Order_Type=&code
  group by Product_ID
  having profit /grosssales > .95
  order by Profit desc;
quit;
title;
```

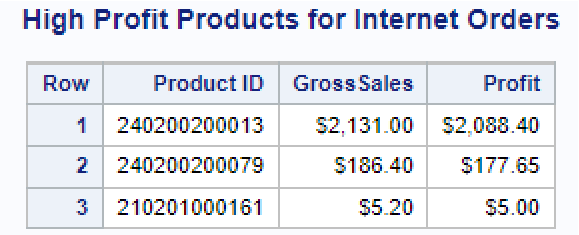


High Profit Products for Retail Store Orders

Row	Product ID	Gross Sales	Profit
1	240200200081	$852.40	$820.35
2	210201000161	$5.20	$5.00

```
proc sql noprint;
select Order_Type
        into :Type1-
        from mc1.order_type_codes;
quit;
%put &=Type1 &=Type2 &=Type3;

%let code=3;
title "High Profit Products for &&Type&code Orders";
proc sql number;
select Product_ID format=z12.,
    Sum(Total_Retail_Price) format=dollar10.2 as GrossSales,
    Sum(Total_Retail_Price-CostPrice_Per_Unit) format=dollar10.2 as Profit
  from mc1.orders
  where Order_Type=&code
  group by Product_ID
  having profit /grosssales > .95
  order by Profit desc;
quit;
```

```
title;
```

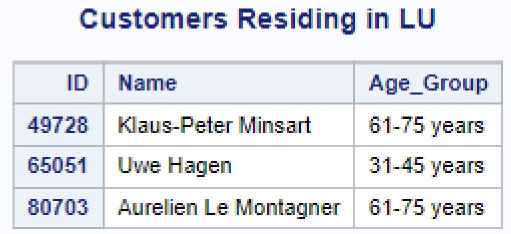


High Profit Products for Internet Orders

Row	Product ID	GrossSales	Profit
1	240200200013	$2,131.00	$2,088.40
2	240200200079	$186.40	$177.65
3	210201000161	$5.20	$5.00

```
/****************************************************

* Indirect References to Macro Variables: Practice #9 *

****************************************************/

/*Level 2 Practice: Using Indirect References to Macro Variables

Reminder: If you restarted your SAS session, you must submit the libname.sas program in the EMC1V2
folder

to access your practice files.

Open m103p09.sas from the practices folder and review the code. Submit the program and review the
results.

Verify that the report is titled Customers Residing in LU and that there are three rows in the report.

At the top of the program, insert a DATA _NULL_ step to create a series of macro variables from the
mc1.country_codes table.

Use the value in the CountryCode column as the macro variable name, and

use the value of the corresponding CountryName column as the macro variable's value.

Submit the DATA _NULL_ step to create the macro variables.

Modify the TITLE statement to include the country name based on the value of the code macro variable.

Verify that the title is Customers Residing in Luxembourg.

Modify the %LET statement to assign a value of ZA to the macro variable, code.

Submit the program and review the results.

How many customers are from South Africa?

Note: Type a numeric value for your answer.

*/

%let code=LU;

title "Customers Residing in &code";
```

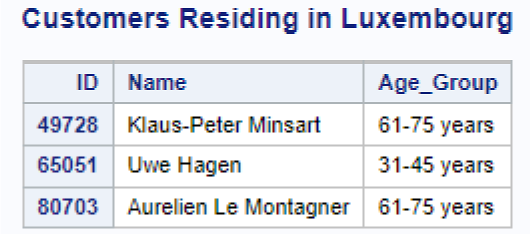
```
proc print data=mc1.customers;
   id ID;
   var Name Age_Group;
   where Country="&code";
run;
title;
```



Customers Residing in LU

ID	Name	Age_Group
49728	Klaus-Peter Minsart	61-75 years
65051	Uwe Hagen	31-45 years
80703	Aurelien Le Montagner	61-75 years

```
data _null_;
           set mc1.country_codes;
           call symputx(CountryCode, CountryName);
run;


%let code=LU;
title "Customers Residing in &&&code";
proc print data=mc1.customers;
   id ID;
   var Name Age_Group;
   where Country="&code";
run;
title;
```

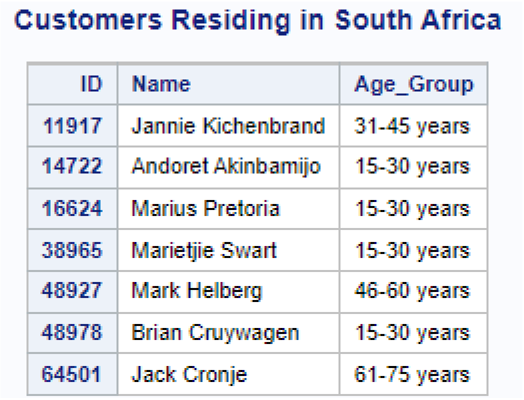


Customers Residing in Luxembourg

ID	Name	Age_Group
49728	Klaus-Peter Minsart	61-75 years
65051	Uwe Hagen	31-45 years
80703	Aurelien Le Montagner	61-75 years

```
%let code=ZA;
```

```
title "Customers Residing in &&&code";

proc print data=mc1.customers;

   id ID;

   var Name Age_Group;

   where Country="&code";

run;

title;
```



Customers Residing in South Africa

ID	Name	Age_Group
11917	Jannie Kichenbrand	31-45 years
14722	Andoret Akinbamijo	15-30 years
16624	Marius Pretoria	15-30 years
38965	Marietjie Swart	15-30 years
48927	Mark Helberg	46-60 years
48978	Brian Cruywagen	15-30 years
64501	Jack Cronje	61-75 years