SAS Advanced Programmer

SAP2 Macro

SAP206 SAS Case Study

Libname.sas

```
%let path=~/EMC1V2;

libname mc1 "&path/data";

options sasautos=("&path/autocall", sasautos);
```

The Business Problem and Required Deliverables

In this case study, you solve a real-world data problem by applying concepts that you learned in this course. Be aware that there are numerous solutions to this problem, and some can include concepts that are outside the scope of the course.

Business Problem

Orion Star company maintains detailed tables about the products that they offer, the orders placed by customers, and the suppliers for each product. A SAS program has been written to prepare the data to summarize profit for each supplier. The first portion of the program identifies the top suppliers. The second part of the program must be edited to generate a summary report for a particular supplier. Information about the selected supplier, including the ID, name, and country, must be manually replaced in the program.

The program should be modified to automatically generate a summary report for the top five suppliers for a selected method of order type (1=Retail, 2=Catalog, 3=Internet).

Deliverables

There are two deliverables that need to be completed by the end of the case study:

- **replacespace.sas** – A macro program that replaces all spaces in a string with an underscore. The macro should be stored and retrieved using the autocall facility.
- **supplierreport.sas** – A macro program that identifies the top *n* suppliers for either all or a selected order type and generates a summary PDF report for each separate supplier. The macro should automatically create the reports with the only input required being the selected order type and the number of suppliers to include. The macro should be stored and retrieved using the autocall facility.

The Case Study Files

The files used in this case study are included in the SAS Macro 1: Essentials course data. It is assumed that the course data was previously set up in your SAS environment. Be sure to submit the **libname.sas** program in the course files to define the **Path** macro variable and the **MC1** library.

The case study uses the following files from the **case_study** folder:

- **CaseStudyStart.sas**

From the **data** folder:

- **orders** table
- **products** table
- **country_codes** table

Note: Remember that you can use **&path** to reference the location of your files as you write the case study program.

What the SAS Program does

Part 1 of the **CaseStudyStart.sas** program does the following:

- creates a temporary table named **OrderDetail** that joins all rows from the **orders** table with the corresponding information from the **product_dim** table. Profit is calculated for each order where **Order_Type**=*1* (retail sales).
- summarizes **Profit** and identifies **Supplier_ID**, **Supplier_Name**, and **Supplier_Country** for each unique supplier. Results are ordered by descending values of **Profit**.

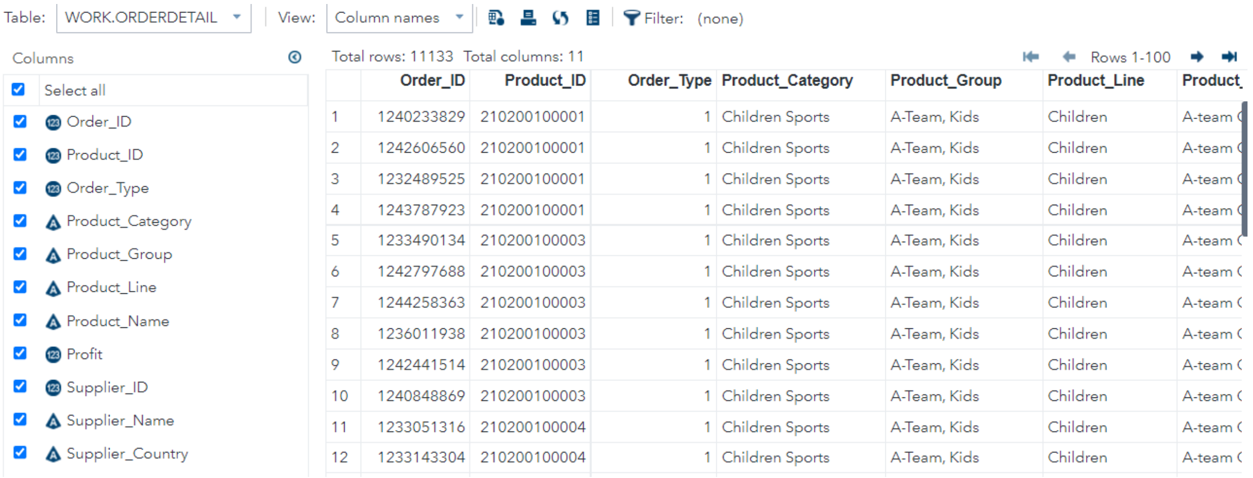
The **country_codes** lookup table is printed so that you can look up the corresponding country name for the top supplier **Supplier_Country**.

Part 2 of the program creates a PDF file for **Supplier_ID**=*1303*, the top supplier identified by running part 1 of the program. The report includes a bar chart that represents the sum of profit for each product category, and a table with the total and average profit for each product group.

The PDF linked below shows the data layout for each input table in the case study. The layout lists the column name, a description, and any value requirements for that column. You will refer to this document as you prepare the data.

MacroCS_Data_Layout.pdfPDF File

```
*********************************************************
*  Concepts used:                            *
*  1) Create a macro function, store as AUTOCALL        *
*  2) Use %SYSFUNC                           *
*  3) Create macro with a parameter                  *
*  4) Use %IF/%THEN based on macro parameter value      *
*  5) Use %PUT to generate custom messages in log       *
*  6) Create a series of macro variables              *
*  7) Use %DO loop and indirect macro variable reference *
*********************************************************;

*********************************************************
* Be sure to first run cre8data.sas and libname.sas     *
*********************************************************;

 /*PART 1*/
 /*Run first two steps of the program to identify the top Supplier_ID and Supplier_Name*/


 /*This step creates a table that joins Orders with Products and calculates Profit per Order*/
 /*Include Retail Sales (order_type=1) only*/
proc sql;
create table OrderDetail as
select Order_ID, o.Product_ID, Order_Type, Product_Category,
      Product_Group, Product_Line, Product_Name,
      (total_retail_price-(costprice_per_unit*quantity)) as Profit,
      Supplier_ID, Supplier_Name, Supplier_Country
   from mc1.orders as o
       left join  mc1.products as p
                on o.Product_ID=p.Product_ID
             where order_type=1;
quit;
```



☑ Select all
☑ Order_ID
☑ Product_ID
☑ Order_Type
☑ Product_Category
☑ Product_Group
☑ Product_Line
☑ Product_Name
☑ Profit
☑ Supplier_ID
☑ Supplier_Name
☑ Supplier_Country

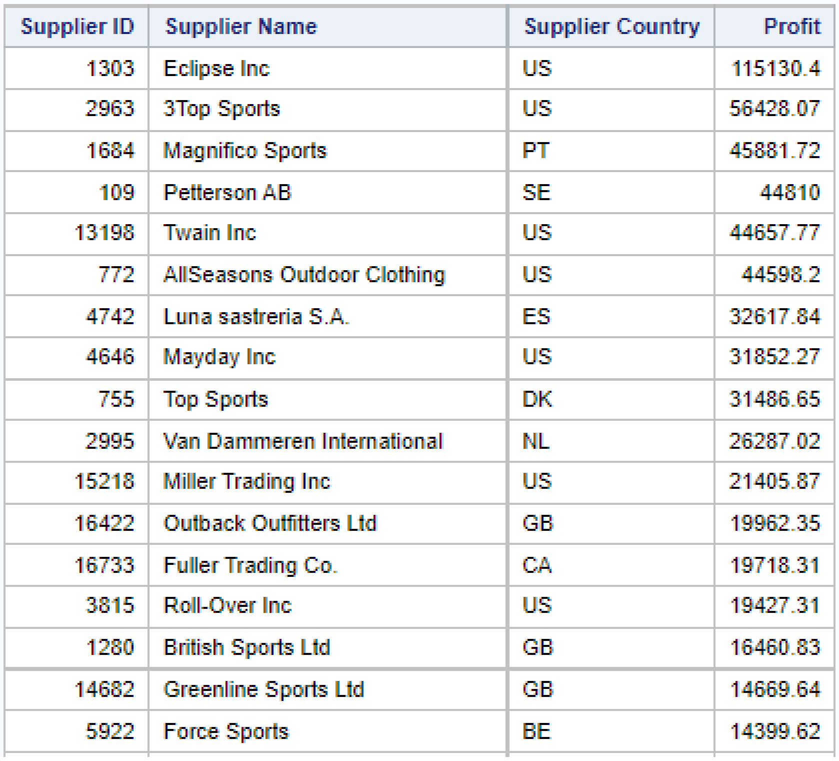
Total rows: 11133 Total columns: 11   Rows 1-100

	Order_ID	Product_ID	Order_Type	Product_Category	Product_Group	Product_Line	Product_
1	1240233829	210200100001	1	Children Sports	A-Team, Kids	Children	A-team
2	1242606560	210200100001	1	Children Sports	A-Team, Kids	Children	A-team
3	1232489525	210200100001	1	Children Sports	A-Team, Kids	Children	A-team
4	1243787923	210200100001	1	Children Sports	A-Team, Kids	Children	A-team
5	1233490134	210200100003	1	Children Sports	A-Team, Kids	Children	A-team
6	1242797688	210200100003	1	Children Sports	A-Team, Kids	Children	A-team
7	1244258363	210200100003	1	Children Sports	A-Team, Kids	Children	A-team
8	1236011938	210200100003	1	Children Sports	A-Team, Kids	Children	A-team
9	1242441514	210200100003	1	Children Sports	A-Team, Kids	Children	A-team
10	1240848869	210200100003	1	Children Sports	A-Team, Kids	Children	A-team
11	1233051316	210200100004	1	Children Sports	A-Team, Kids	Children	A-team
12	1233143304	210200100004	1	Children Sports	A-Team, Kids	Children	A-team

```
NOTE: Table WORK.ORDERDETAIL created, with 11133 rows and 11 columns.
```

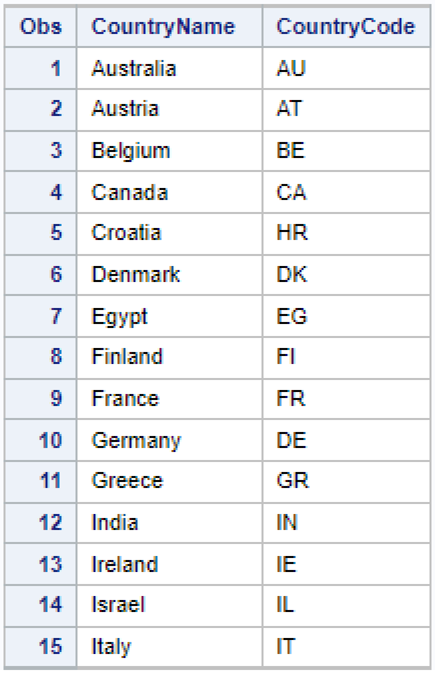
```
/*This step summarizes profit and ranks suppliers*/
proc sql;
select Supplier_ID format=12.,
      Supplier_Name,
      Supplier_Country,
      sum(profit) as Profit
   from OrderDetail
```

```
        group by Supplier_ID, Supplier_Name, Supplier_Country
        order by Profit desc;
quit;
```



Supplier ID	Supplier Name	Supplier Country	Profit
1303	Eclipse Inc	US	115130.4
2963	3Top Sports	US	56428.07
1684	Magnifico Sports	PT	45881.72
109	Petterson AB	SE	44810
13198	Twain Inc	US	44657.77
772	AllSeasons Outdoor Clothing	US	44598.2
4742	Luna sastreria S.A.	ES	32617.84
4646	Mayday Inc	US	31852.27
755	Top Sports	DK	31486.65
2995	Van Dammeren International	NL	26287.02
15218	Miller Trading Inc	US	21405.87
16422	Outback Outfitters Ltd	GB	19962.35
16733	Fuller Trading Co.	CA	19718.31
3815	Roll-Over Inc	US	19427.31
1280	British Sports Ltd	GB	16460.83
14682	Greenline Sports Ltd	GB	14669.64
5922	Force Sports	BE	14399.62

```
/*This step prints the country_codes table to look up the country names*/
proc print data=mc1.country_codes;
run;
```

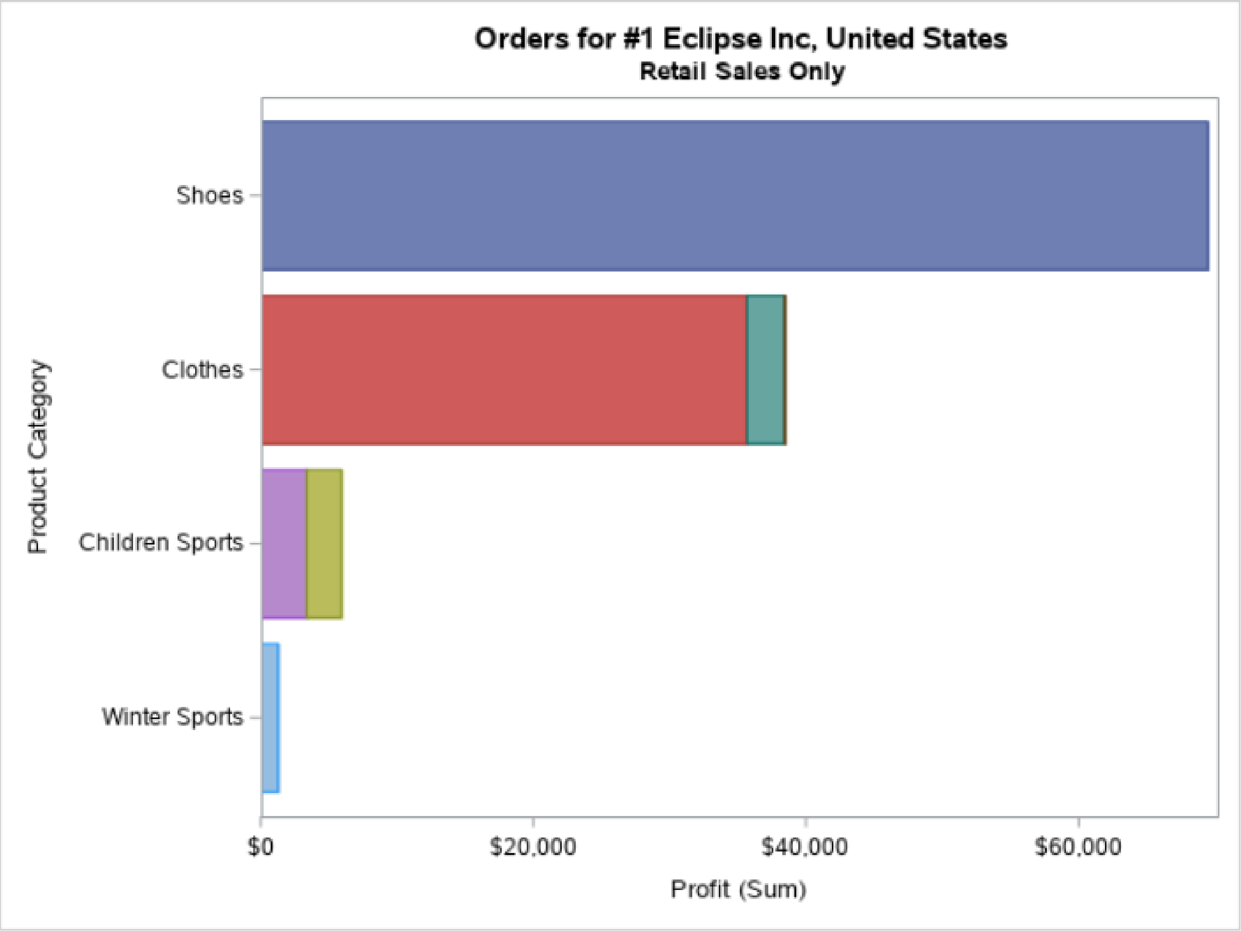


Obs	CountryName	CountryCode
1	Australia	AU
2	Austria	AT
3	Belgium	BE
4	Canada	CA
5	Croatia	HR
6	Denmark	DK
7	Egypt	EG
8	Finland	FI
9	France	FR
10	Germany	DE
11	Greece	GR
12	India	IN
13	Ireland	IE
14	Israel	IL
15	Italy	IT

```
NOTE: There were 32 observations read from the data set MC1.COUNTRY_CODES.
```

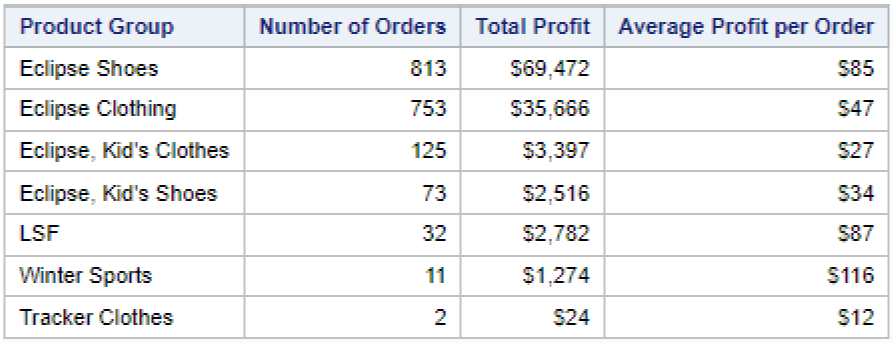
```
/*PART 2*/


 /* In the remainder of the program, the Name, ID and Country for the top supplier
     have been substituted in the appropriate places. Spaces have been replaced with
          underscores in the file name. TITLE2 indicates the type of orders included. */


 /* To modify the program to generate the report for the #2 supplier, update the supplier
          name, ID, country and rank in the appropriate places. */
options nodate;
ods graphics on / imagefmt=png;
ods pdf file="&path/case_study/1_Eclipse_Inc.pdf" style=meadow startpage=no nogtitle;


title "Orders for #1 Eclipse Inc, United States";
title2 "Retail Sales Only";
 /*Create a summary bar chart by Product_Category*/
proc sgplot data=OrderDetail noautolegend ;
```

```
        hbar Product_Category / response=profit stat=sum group=Product_Group
categoryorder=respdesc;

        where Supplier_ID=1303;

        format profit dollar8.;

run;

title;
```



```
NOTE: Writing ODS PDF output to DISK destination
"/home/u58304328/EMC1V2/case_study/1_Eclipse_Inc.pdf", printer "PDF".

NOTE: There were 1809 observations read from the data set
WORK.ORDERDETAIL.
      WHERE Supplier_ID=1303;
```

```
/*Create a summary report of orders and sales for the selected supplier*/

proc sql;

        select Product_Group,
      count(order_id) as NumOrders "Number of Orders",

      sum(profit) as TotalProfit "Total Profit" format=dollar8.,

      avg(profit) as AvgProfit "Average Profit per Order" format=dollar6.

        from OrderDetail

        where Supplier_ID=1303

        group by Product_Group

        order by calculated numorders desc;

quit;


footnote;

ods pdf close;
```



Product Group	Number of Orders	Total Profit	Average Profit per Order
Eclipse Shoes	813	$69,472	$85
Eclipse Clothing	753	$35,666	$47
Eclipse, Kid's Clothes	125	$3,397	$27
Eclipse, Kid's Shoes	73	$2,516	$34
LSF	32	$2,782	$87
Winter Sports	11	$1,274	$116
Tracker Clothes	2	$24	$12

```
NOTE: ODS PDF printed 1 page to
/home/u58304328/EMC1V2/case_study/1_Eclipse_Inc.pdf.
```

/*Macro Case Study: Guided Version

In this version of the case study, you receive the high-level requirements and specific suggestions to solve the business problem. Your job is to familiarize yourself with the CaseStudyStart.sas program and identify what must be edited in the program to identify the top five suppliers and create a separate PDF report for each supplier.

1. Read the comments in the CaseStudyStart.sas program to get familiar with the code and the edits required to generate a report for a different supplier and subset of Order_Type.

2. Create a macro function named %REPLACESPACE that uses the TRANWRD function to replace all spaces in a string with underscores. Save the replacespace.sas macro program in the autocall folder and enable the autocall facility to read it.

a. Create a new program. Start a macro definition named replacespace with a single positional parameter named Text.

b. Use the %SYSFUNC function to execute the TRANWRD function.

c. The first argument is the value of the Text parameter. The second argument is the target character, which is a space. The third argument is the replacement character, which is an underscore.

Note: Use appropriate macro quoting functions (if necessary) to the arguments.

d. Close the macro definition.

e. Save the program as replacespace.sas in the autocall folder.

f. In the CaseStudyStart.sas program, add an OPTIONS statement to indicate that the SASAUTOS search path should include the autocall folder and the SASAUTOS library.

```
*/

*Solution;

%macro replacespace(Text);

        %sysfunc(tranwrd(%quote(&Text),%str( ),_));

%mend replacespace;

/*

%macro replacespace(Text);

%let result=%sysfunc(tranwrd(%quote(&Text),%str( ) ,%str(_)));

%put &=result;

%mend replacespace;

*/

/* %replacespace(Good Morning My Love. I Love You.) */

/* %let have = a,b,c;

%let want = %sysfunc(tranwrd(%quote(&have),%str(,),%str( )));

%put &=want;

WANT=a b c */
```

```
 78         %replacespace(Good Morning My Love. I Love You.)
 RESULT=Good_Morning_My_Love._I_Love_You.
```

/* Modify the code in the CaseStudyStart.sas program to build a macro named %SupplierReport

with a parameter to select a particular Order_Type value.

a. At the top of the program, start a macro definition named SupplierReport with ot as a

positional parameter.

b. At the bottom of the program, end the macro definition with a %MEND statement.

4. Validate the ot parameter value to ensure that it is either 1, 2, or 3. If no value is provided, write a custom message to the log. The message should indicate that a value is required and that the program will stop executing. It should also include a list of valid values. If a value other than 1, 2, or 3 is provided, write a custom error message to the log that prints a list of valid values and stops processing the rest of the program

Note: Be sure to use the MINOPERATOR option in the %MACRO statement to enable the macro IN operator.

a. Add the MINOPERATOR option in the %MACRO statement.

b. Use %IF and %END statements to test whether the parameter is equal to a null value. If it is, use %PUT statements to write a custom error message to the log that indicates that a value is required and that the program will stop executing. It should also include a list of valid values. The error message could appear as follows:

ERROR: You did not specify an Order_Type code (required).

Valid Order_Type values include 1 (retail), 2 (catalog), or 3 (internet).

Program will stop executing

c. Use the %RETURN statement to stop execution and %END to close the %IF block.

d. In the %MACRO statement, add the /MINOPERATOR option to be able to use the IN operator in a macro statement.

e. Use %ELSE, %IF, and %END statements to test whether the parameter is not in the list of 1, 2, or 3. Write a custom error message to the log if an invalid value is provided.

The program should also stop executing. The error message could appear as follows:

ERROR: Valid Order_Type values include 1 (retail), 2 (catalog), or 3 (internet).

Program will stop executing.

f. Use the %RETURN statement to stop execution and %END to close the %IF block.

g. Use %ELSE, %DO, and %END statements to indicate whether the parameter value is valid, and then the rest of the program should run.

5. If a value of 1, 2, or 3 is provided for the parameter, subset the OrderDetail table based on Order_Type.

```
Modify the WHERE statement in the first PROC SQL step to include rows where Order_Type is
equal to the ot macro variable.
*/

/*This step creates a table that joins Orders with Products and calculates Profit per Order*/
%macro SupplierReport(ot) / minoperator;
    proc sql noprint;
        select Order_Type_Code
            into :otlist separated by " "
            from mc1.order_type_codes;
    quit;
    %if &ot =  %then %do;
        %put ERROR: You did not specify an Order_Type code (required).;
        %put ERROR: Valid Order_Type values include 1 (retail), 2 (catalog), or 3 (internet).;
        %put ERROR: Program will stop executing;
        %return;
    %end;
    %if &ot in &otlist %then %do;
        proc sql;
            create table OrderDetail as
            select Order_ID, o.Product_ID, Order_Type, Product_Category,
            Product_Group, Product_Line, Product_Name,
            (total_retail_price-(costprice_per_unit*quantity)) as Profit,
            Supplier_ID, Supplier_Name, Supplier_Country
            from mc1.orders as o
            left join  mc1.products as p
                on o.Product_ID=p.Product_ID
                where order_type=&ot;
        quit;
        %end;

        %else %do;

                %put ERROR: Valid Order_Type values include 1 (retail), 2 (catalog), or 3 (internet).;

                %put ERROR: Program will stop executing;

                %return;

        %end;

%mend SupplierReport;


/* %SupplierReport(1) */

/* %SupplierReport(3) */

/* %SupplierReport(5) */

/* %SupplierReport */


%SupplierReport
```

```
 ERROR: You did not specify an Order_Type code (required).
 ERROR: Valid Order_Type values include 1 (retail), 2 (catalog), or 3
(internet).
 ERROR: Program will stop executing
```

```
%SupplierReport(5)
```

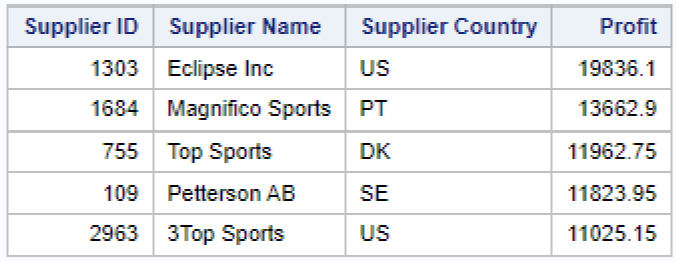
```
 ERROR: Valid Order_Type values include 1 (retail), 2 (catalog), or 3
(internet).
 ERROR: Program will stop executing
```

/*Create a series of macro variables that will store Supplier_ID, Supplier_Name, Supplier_Country, and Profit for each of the top five suppliers. For example, the macro variables TopSupp1, Name1, Country1, and Profit1 will store information about the top supplier; TopSupp2, Name2, Country2, and Profit2 will store information about the secondranked supplier; and so on.

a. Find the second PROC SQL step that identifies the top five suppliers.

b. Add an INTO clause to create the following series of macro variables for the top five suppliers:

```
1) TopSupp1-TopSupp5 to store the Supplier_ID values.

2) Name1-Name5 to store the Supplier_Name values.

3) Country1-Country5 to store the Supplier_Country values.

4) Profit1-Profit5 to store the sum of Profit values.

*/

/*This step summarizes profit and ranks suppliers*/

proc sql outobs=5;

select Supplier_ID format=12.,

       Supplier_Name,

       Supplier_Country,

       sum(profit) as Profit

   into :TopSupp1-, :Name1-, :Country1-, :Profit1-

   from OrderDetail

          group by Supplier_ID, Supplier_Name, Supplier_Country

          order by Profit desc;

quit;
```



Supplier ID	Supplier Name	Supplier Country	Profit
1303	Eclipse Inc	US	19836.1
1684	Magnifico Sports	PT	13662.9
755	Top Sports	DK	11962.75
109	Petterson AB	SE	11823.95
2963	3Top Sports	US	11025.15

```
/*This step summarizes profit and ranks suppliers*/

proc sql;

select Supplier_ID format=12.,

       Supplier_Name,

       Supplier_Country,

       sum(profit) as Profit

   into :TopSupp1-:TopSupp5, :Name1-:Name5, :Country1-:Country5, :Profit1-:Profit5

   from OrderDetail
```

```
        group by Supplier_ID, Supplier_Name, Supplier_Country
        order by Profit desc;
quit;

%mend SupplierReport;

/*Create a series of macro variables named Country_CC where CC is the two-letter
CountryCode value read from the mc1.country_codes table. Assign the corresponding
CountryName value to each macro variable.
a. Write a DATA step that reads the mc1.country_codes table.
b. Use CALL SYMPUTX to create the series of macro variables. The first argument should
concatenate Country_ with the value of CountryCode to create the macro variable names.
The second argument should assign the value from the CountryName column.
*/

 /*This step prints the country_codes table to look up the country names*/
data _null_;
   set mc1.country_codes;
   call symputx(cats('country_',CountryCode), CountryName);
run;



/*8. Use a macro DO loop to repeat Part 2 of the program five times. The first time through the loop,
the program should generate the PDF report for the top supplier. The report should be modified as
follows:
a. The prefix for each PDF file name should be the supplier rank number, 1 through 5.
The name of each PDF file should be the value of Supplier_Name with all spaces replaced with
underscores.
Use the REPLACESPACE custom macro function.
```

1) After the ODS GRAPHICS statement, add a %DO macro statement with an index variable i that starts at 1 and ends at 5.

2) At the end of the program, before the %END statement (this closes the %IF %THEN/%DO block), add another %END statement.

3) After the %MACRO statement, add a %LOCAL statement to ensure that i is written to and read from the local symbol table.

4) In the ODS PDF statement, delete the hardcoded supplier name, 1_Eclipse_Inc (keep

the .pdf extension) and replace it with an expression that does the following:

a) calls the %replacespace macro

b) includes the value of the macro variable i followed by an underscore as the file name prefix.

c) uses an indirect macro variable reference as the parameter for the %replacespace macro.

The indirect reference should substitute the value of the Name1, Name2 (and so on) macro variable.

b. The first title should be the rank of the supplier and then the Supplier_Name value, followed

by the full country name for that particular supplier (for example, Orders for #1 Eclipse Inc, United States).

1) After the ODS statement, add a %LET statement to create a macro variable named CC

that will be the two-letter CountryCode for the supplier being analyzed in the loop.

(For example, when i=1, the value of CC is the CountryCode assigned to the Country1 macro variable.)

This requires an indirect macro variable reference. This macro variable is used later to insert the country name in the title.

2) In the TITLE statement, use the i macro variable to substitute the rank number of the supplier.

3) Use an indirect macro variable reference to substitute the macro variable value for Name1, Name2, and so on.

4) Use an indirect macro variable reference to substitute the full country name.

Remember that the macro variable Country_CC, where CC is the two-letter CountryCode for the

supplier, stores the country name. Use the CC macro variable created earlier as part of the indirect reference.

c. The second title should be one of the following, depending on the value of the ot parameter:

Retail Sales Only, Catalog Sales Only, or Internet Sales Only. To create the second title, use %IF, %THEN, and %END statements to provide unique TITLE2 statements depending on the value of the ot parameter.

d. For the bar chart (PROC SGPLOT step), the data should be subset to include one supplier at a time.

Modify the WHERE statement to use an indirect macro variable reference to substitute the Supplier_ID value.

e. For the report (PROC SQL step), a footnote should be added below the report that includes

the date and time that the report was created. The data should also be subset to include only the top supplier.

1) Add a FOOTNOTE statement before the last PROC SQL step.

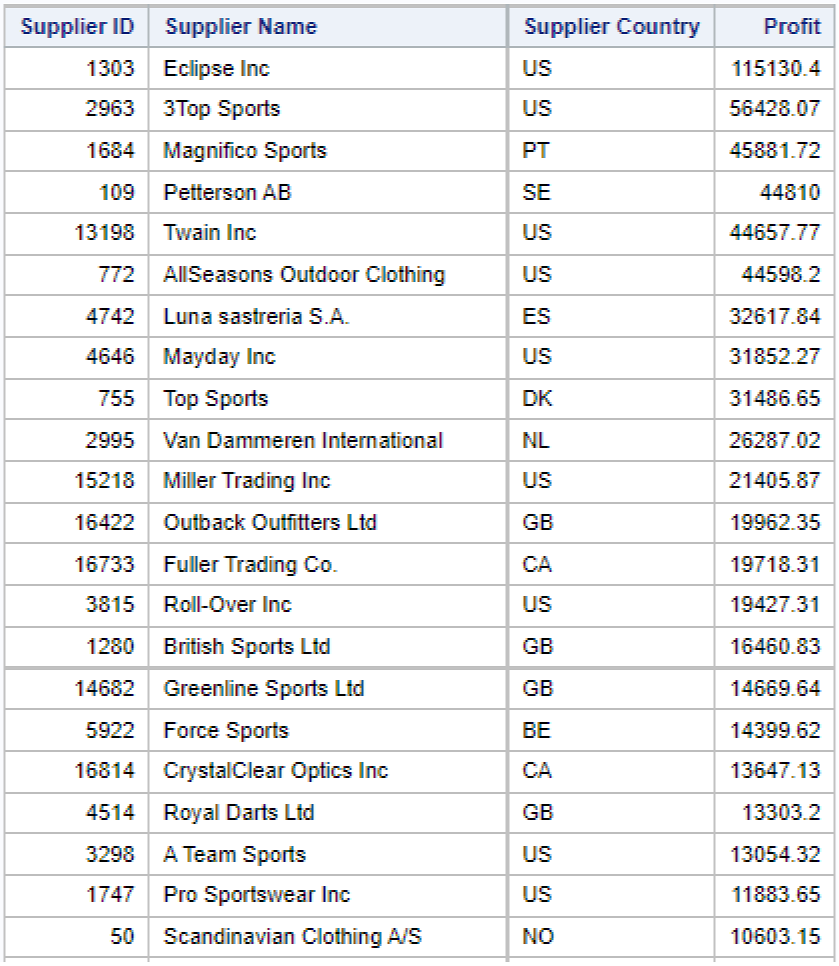
2) Use %SYSFUNC to execute the TODAY() function and format it with an appropriate date format.

3) Use %SYSFUNC again to execute the TIME() function and format it with an appropriate time format.
*/

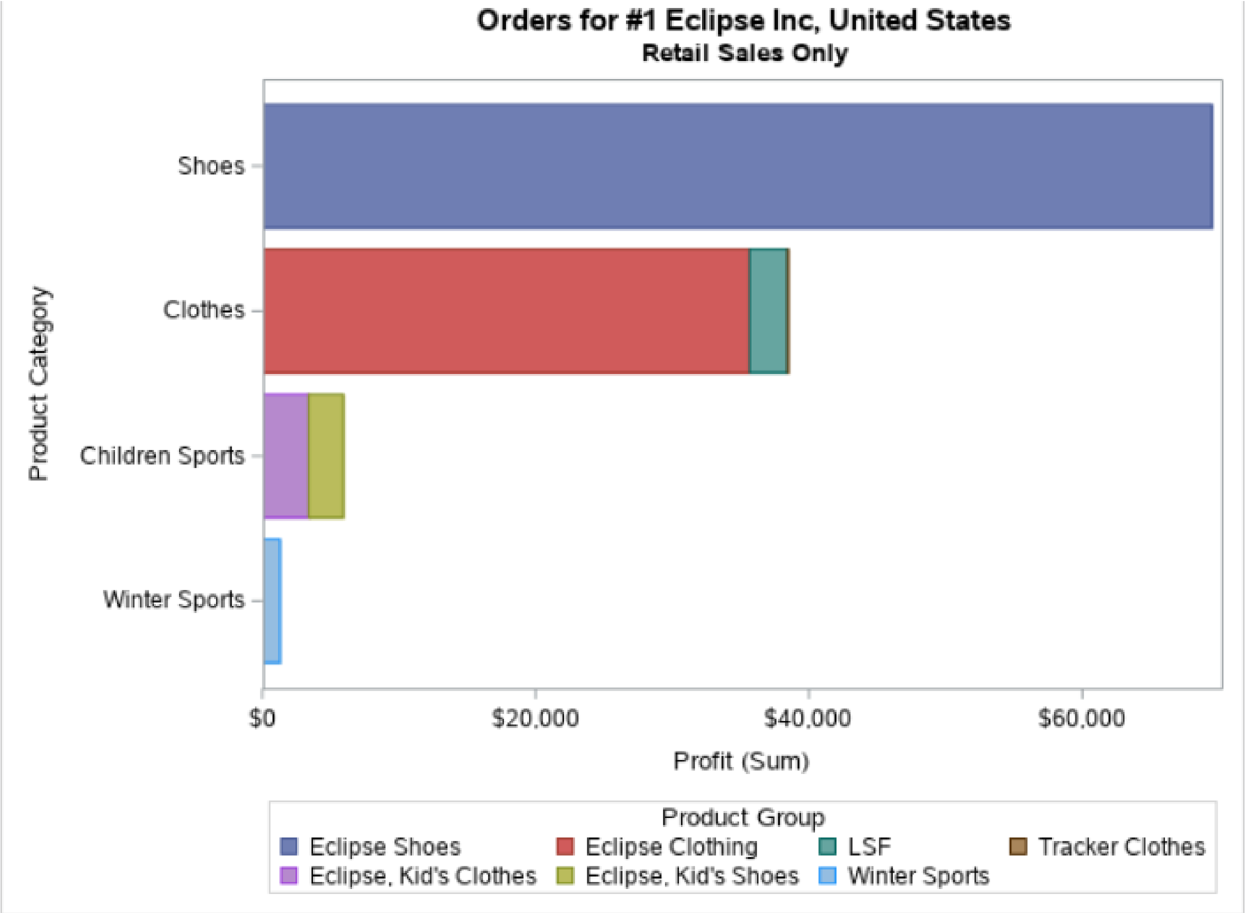
```
 /*PART 2*/

 /* In the remainder of the program, the Name, ID and Country for the top supplier

     have been substituted in the appropriate places. Spaces have been replaced with

            underscores in the file name. TITLE2 indicates the type of orders included. */

 /* To modify the program to generate the report for the #2 supplier, update the supplier

             name, ID, country and rank in the appropriate places. */

options nodate;

ods graphics on / imagefmt=png;


%local i;

%do i=1 %to 5;

        ods pdf file="&path/case_study/%replacespace(&i &&Name &i.pdf") style=meadow
startpage=no nogtitle;

        %let CC=&&Country&i;

        title "Orders for #&i &&Name&i, &&Country_&CC";

        %if &ot=1 %then title2 "Retail Sales Only";

        %else %if &ot=2 %then title2 "Catalog Sales Only";
```

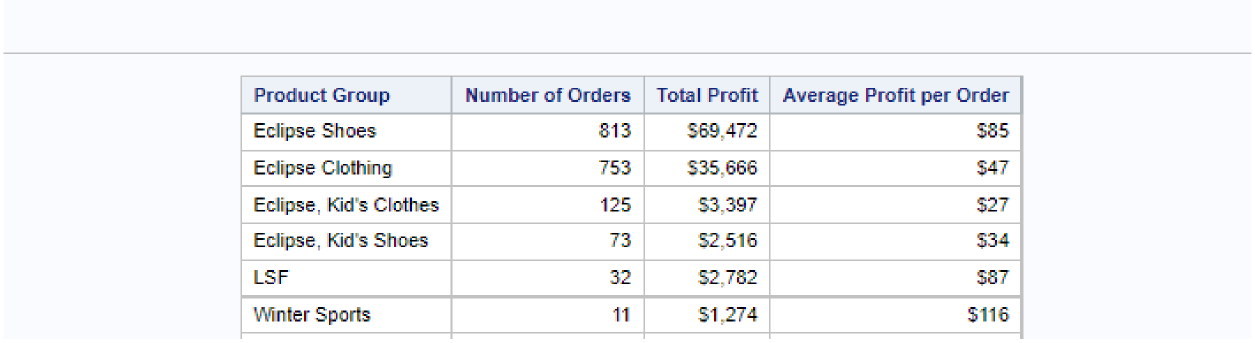
```
          %else %if &ot=3 %then title2 "Internet Sales Only";

          %else title2 "All Sales";

   %end;

 /*Create a summary bar chart by Product_Category*/

proc sgplot data=OrderDetail noautolegend ;

          hbar Product_Category / response=profit stat=sum group=Product_Group
categoryorder=respdesc;

          where Supplier_ID=&&TopSupp&i;

          format profit dollar8.;

run;

title;


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

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

footnote "As of %d at %t";


 /*Create a summary report of orders and sales for the selected supplier*/

proc sql;

          select Product_Group,

        count(order_id) as NumOrders "Number of Orders",

        sum(profit) as TotalProfit "Total Profit" format=dollar8.,

        avg(profit) as AvgProfit "Average Profit per Order" format=dollar6.

          from OrderDetail

          where Supplier_ID=&&TopSupp&i

          group by Product_Group

          order by calculated numorders desc;

quit;


footnote;
```

```
%end;
%end;

ods pdf close;

%mend supplierreport;

/* %SupplierReport(1) */
/* %SupplierReport(2) */
/* %SupplierReport(3) */
/* %SupplierReport(4) */
/* %SupplierReport() */
/* %SupplierReport */

%SupplierReport(2)
```

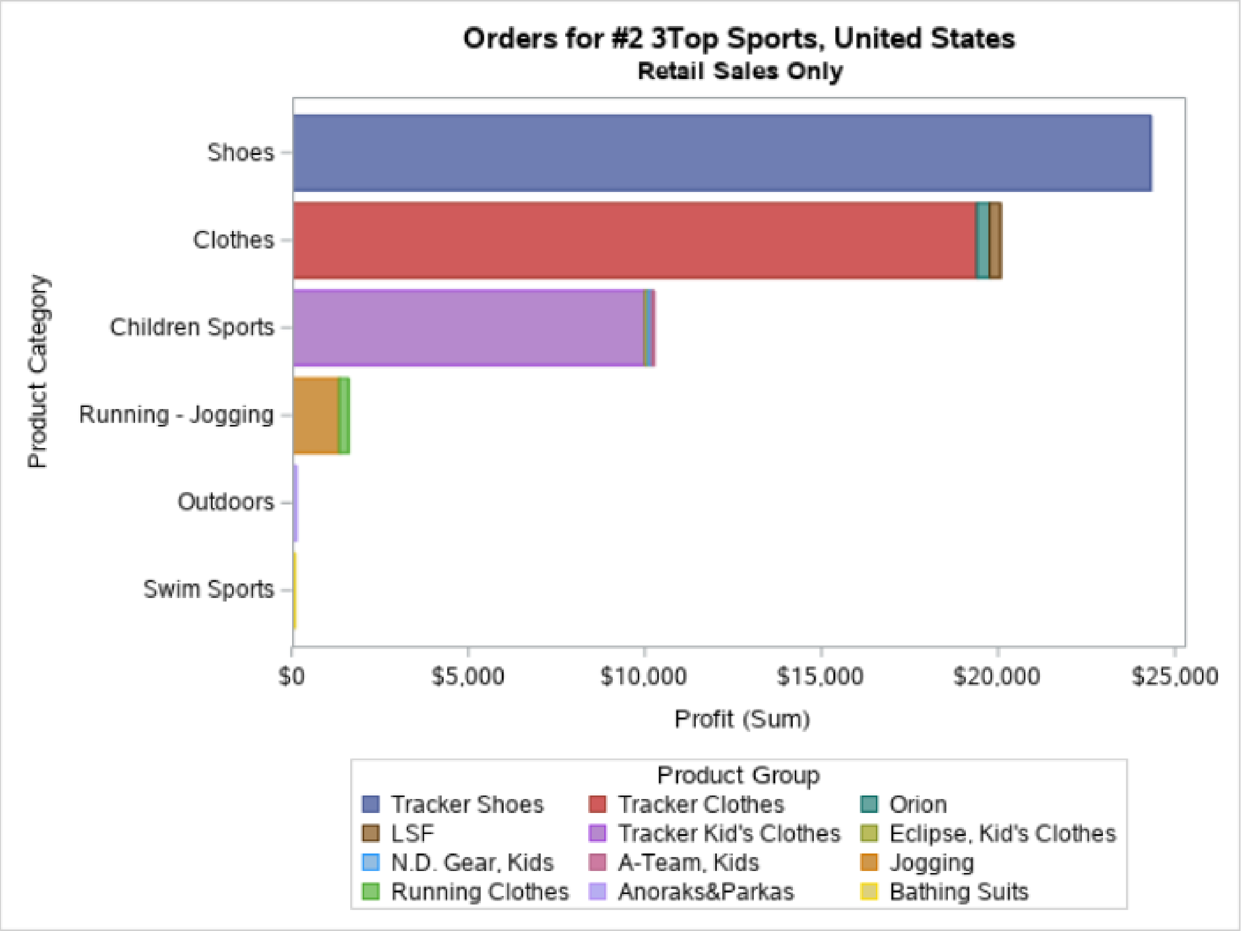


Supplier ID	Supplier Name	Supplier Country	Profit
1303	Eclipse Inc	US	115130.4
2963	3Top Sports	US	56428.07
1684	Magnifico Sports	PT	45881.72
109	Petterson AB	SE	44810
13198	Twain Inc	US	44657.77
772	AllSeasons Outdoor Clothing	US	44598.2
4742	Luna sastreria S.A.	ES	32617.84
4646	Mayday Inc	US	31852.27
755	Top Sports	DK	31486.65
2995	Van Dammeren International	NL	26287.02
15218	Miller Trading Inc	US	21405.87
16422	Outback Outfitters Ltd	GB	19962.35
16733	Fuller Trading Co.	CA	19718.31
3815	Roll-Over Inc	US	19427.31
1280	British Sports Ltd	GB	16460.83
14682	Greenline Sports Ltd	GB	14669.64
5922	Force Sports	BE	14399.62
16814	CrystalClear Optics Inc	CA	13647.13
4514	Royal Darts Ltd	GB	13303.2
3298	A Team Sports	US	13054.32
1747	Pro Sportswear Inc	US	11883.65
50	Scandinavian Clothing A/S	NO	10603.15



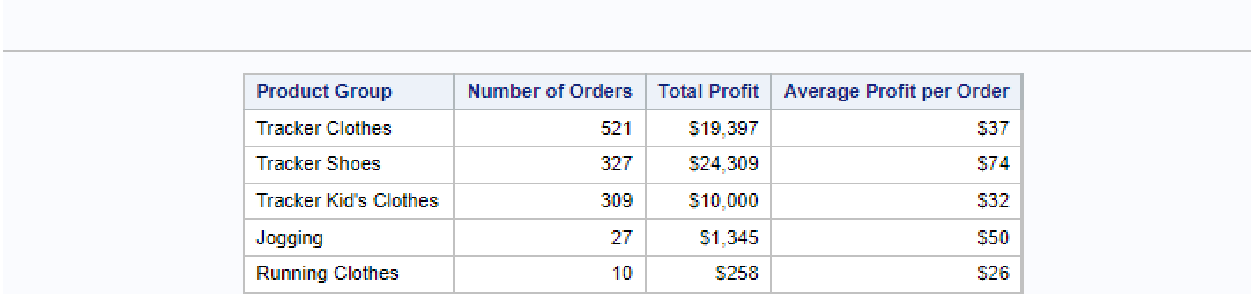


Product Group	Number of Orders	Total Profit	Average Profit per Order
Eclipse Shoes	813	$69,472	$85
Eclipse Clothing	753	$35,666	$47
Eclipse, Kid's Clothes	125	$3,397	$27
Eclipse, Kid's Shoes	73	$2,516	$34
LSF	32	$2,782	$87
Winter Sports	11	$1,274	$116

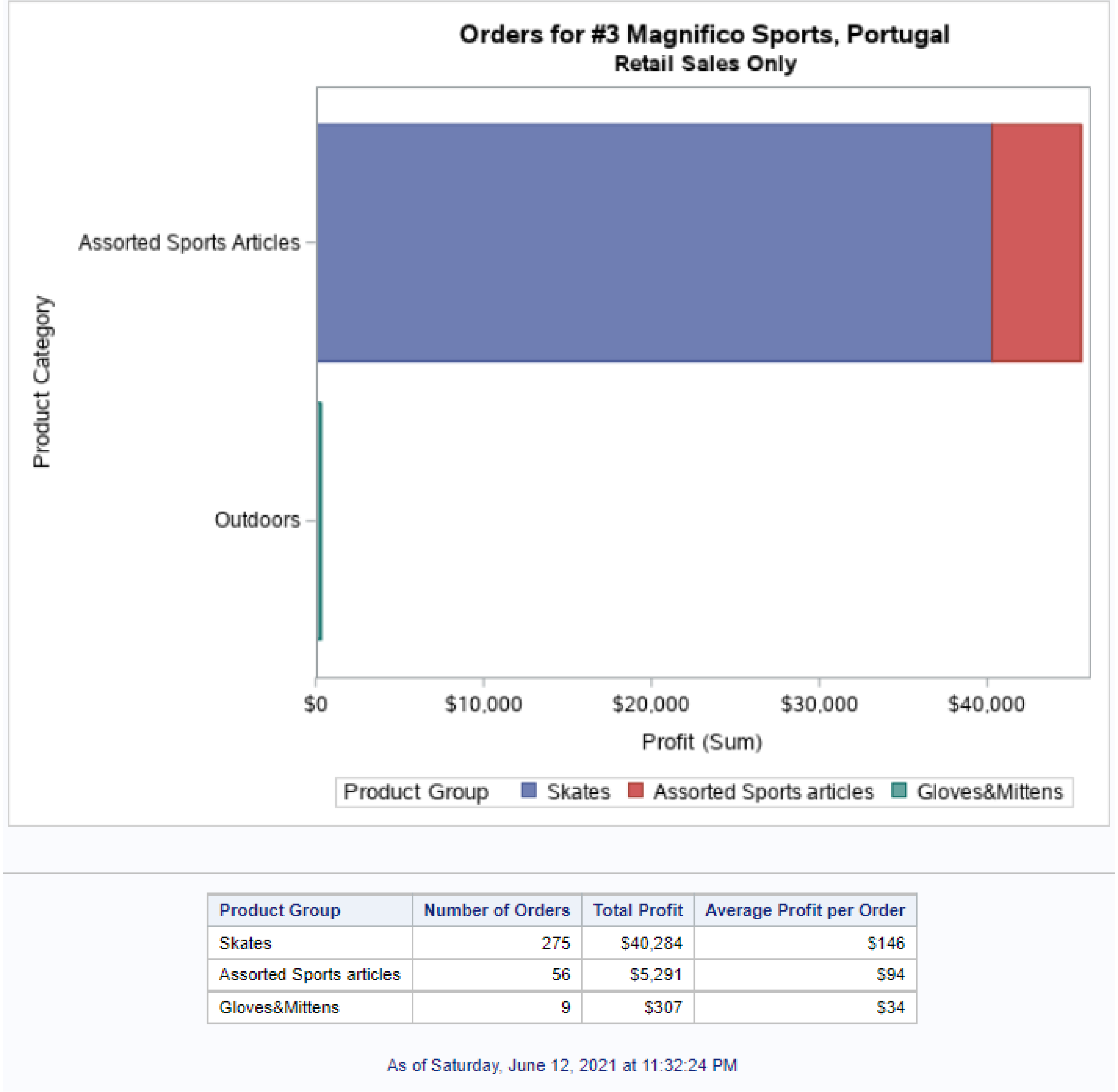


Orders for #2 3Top Sports, United States

Retail Sales Only

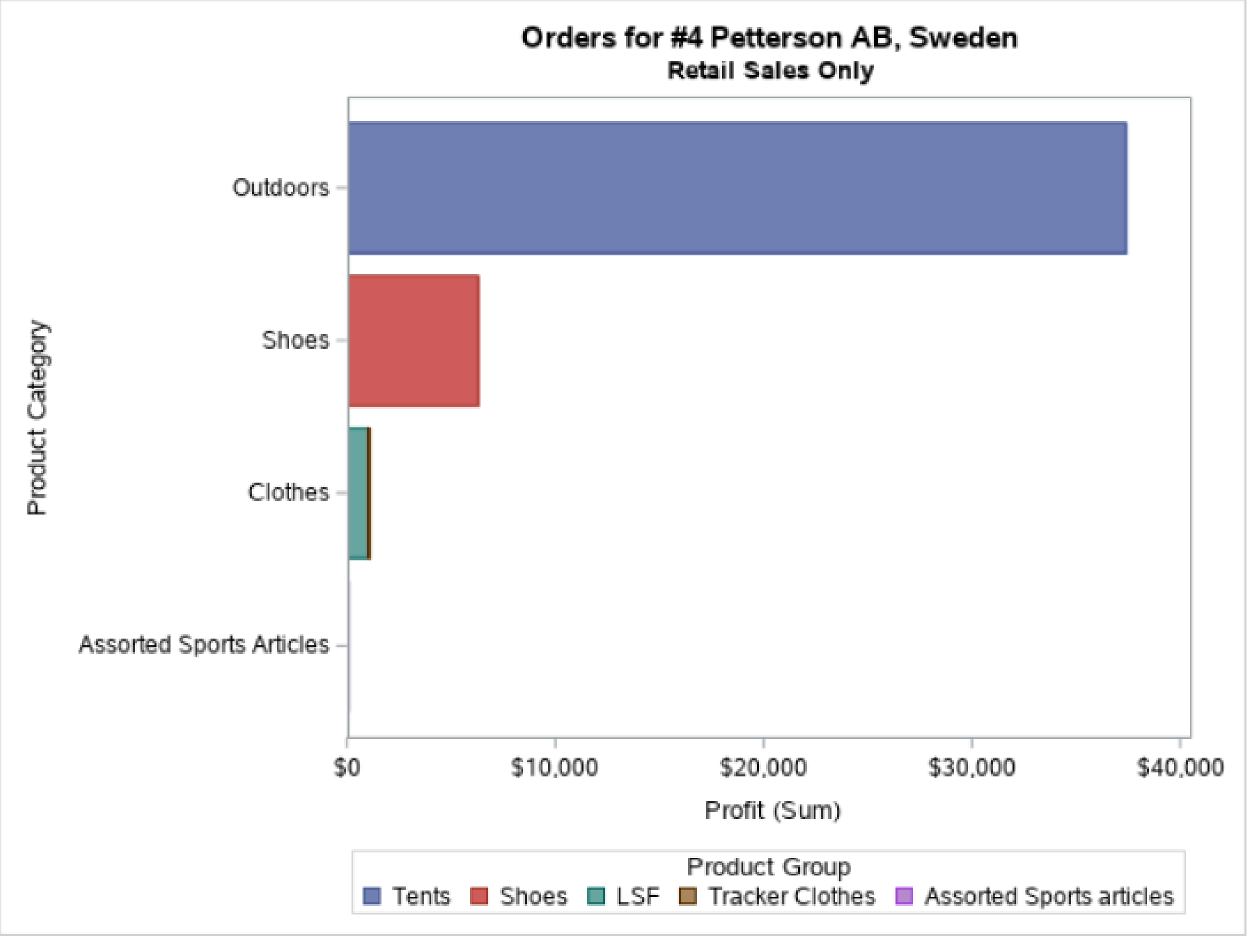


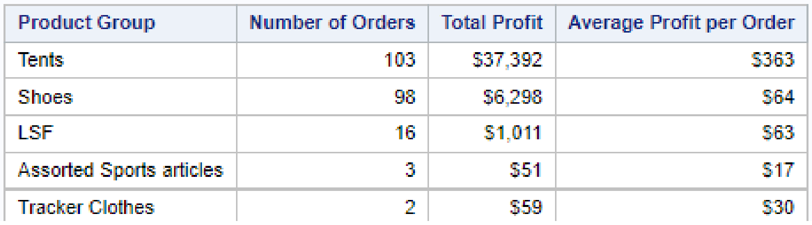
Product Group	Number of Orders	Total Profit	Average Profit per Order
Tracker Clothes	521	$19,397	$37
Tracker Shoes	327	$24,309	$74
Tracker Kid's Clothes	309	$10,000	$32
Jogging	27	$1,345	$50
Running Clothes	10	$258	$26



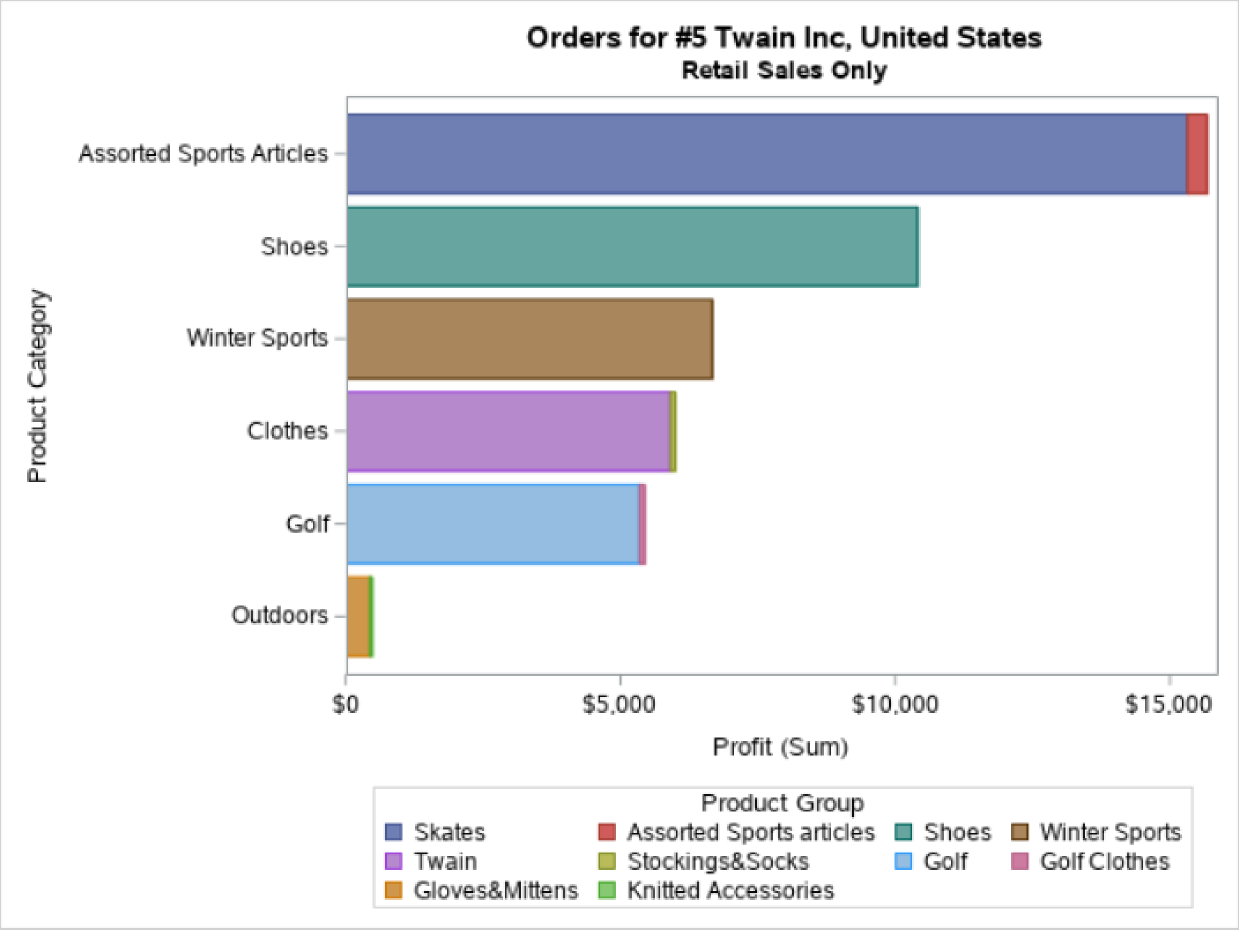
Product Group	Number of Orders	Total Profit	Average Profit per Order
Skates	275	$40,284	$146
Assorted Sports articles	56	$5,291	$94
Gloves&Mittens	9	$307	$34

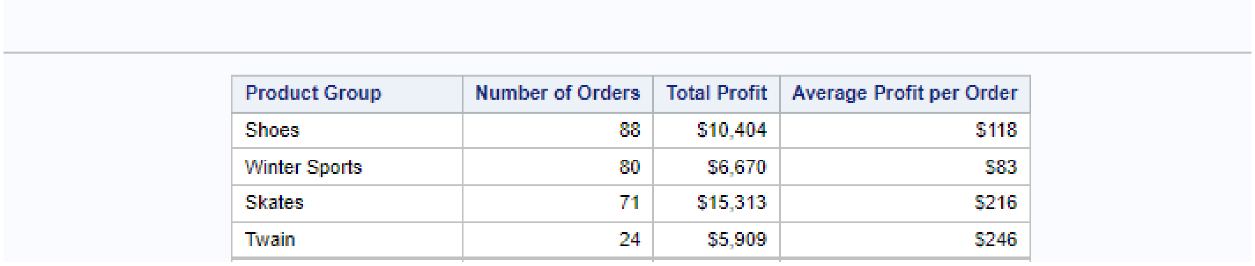
As of Saturday, June 12, 2021 at 11:32:24 PM





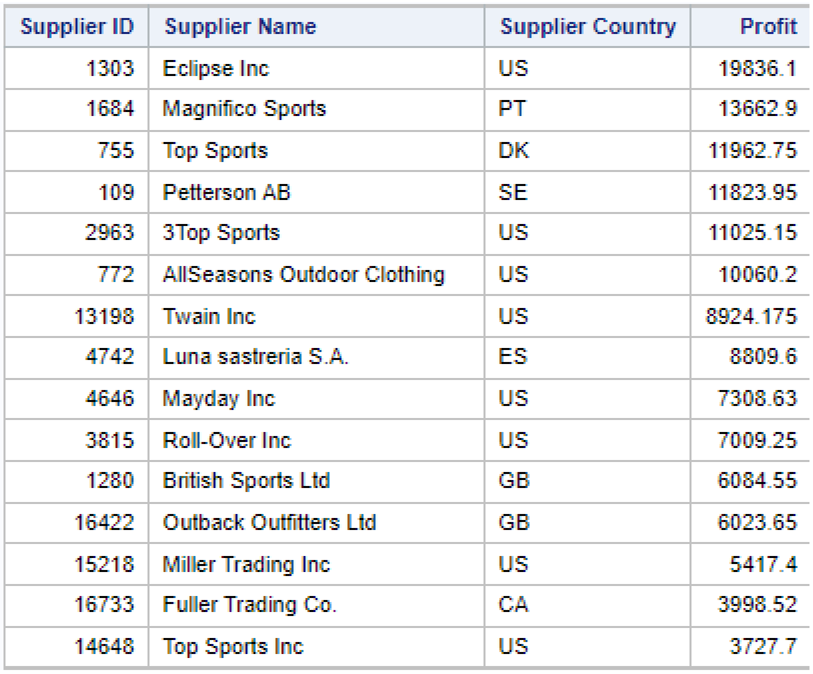
Product Group	Number of Orders	Total Profit	Average Profit per Order
Tents	103	$37,392	$363
Shoes	98	$6,298	$64
LSF	16	$1,011	$63
Assorted Sports articles	3	$51	$17
Tracker Clothes	2	$59	$30



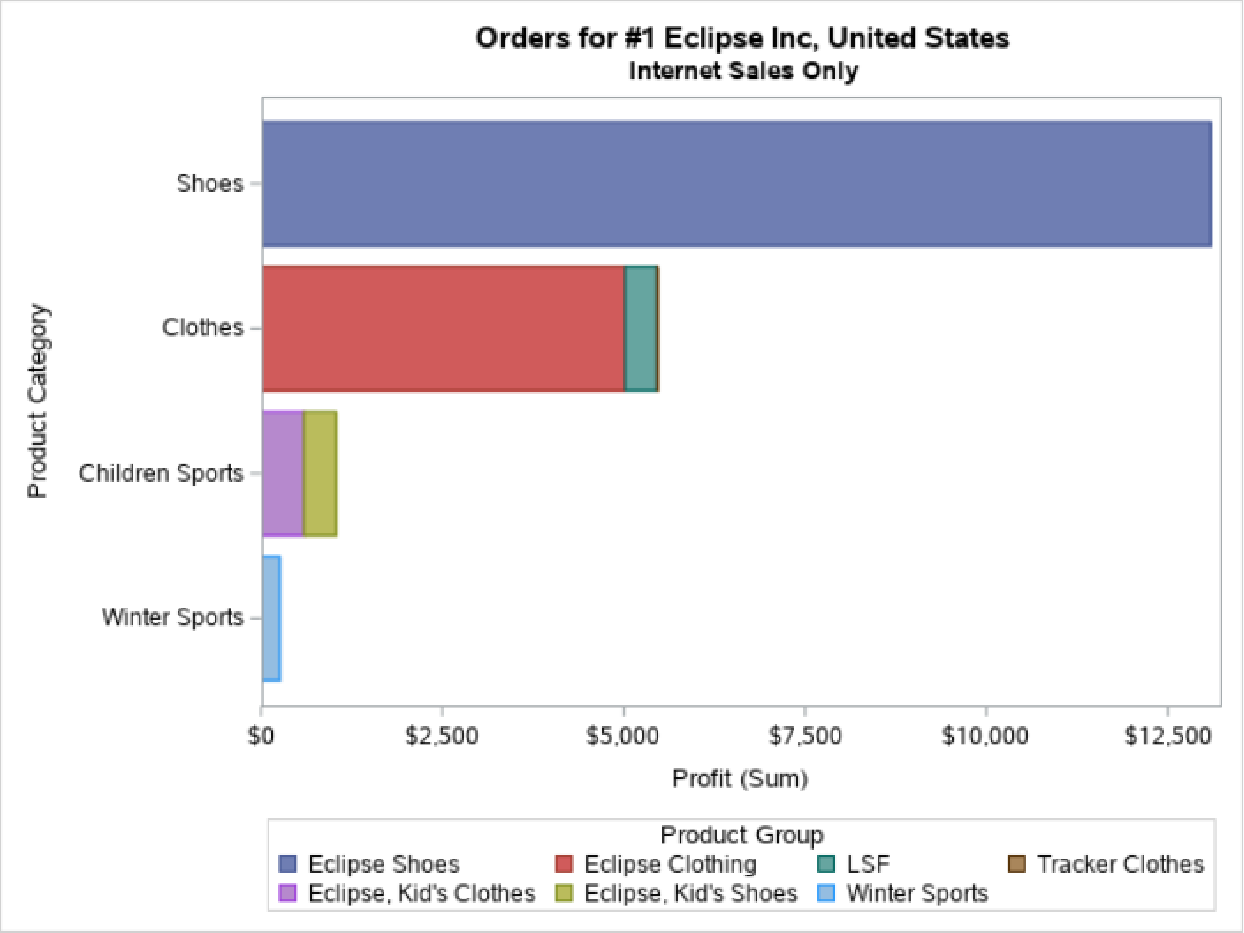


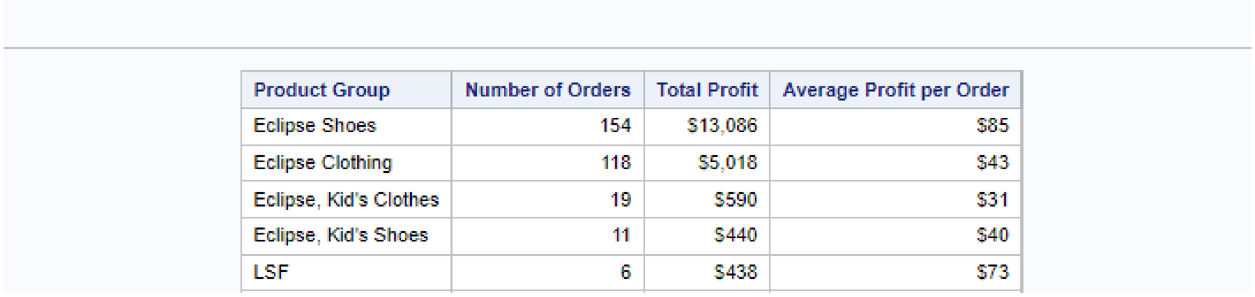
Product Group	Number of Orders	Total Profit	Average Profit per Order
Shoes	88	$10,404	$118
Winter Sports	80	$6,670	$83
Skates	71	$15,313	$216
Twain	24	$5,909	$246

%SupplierReport(2)

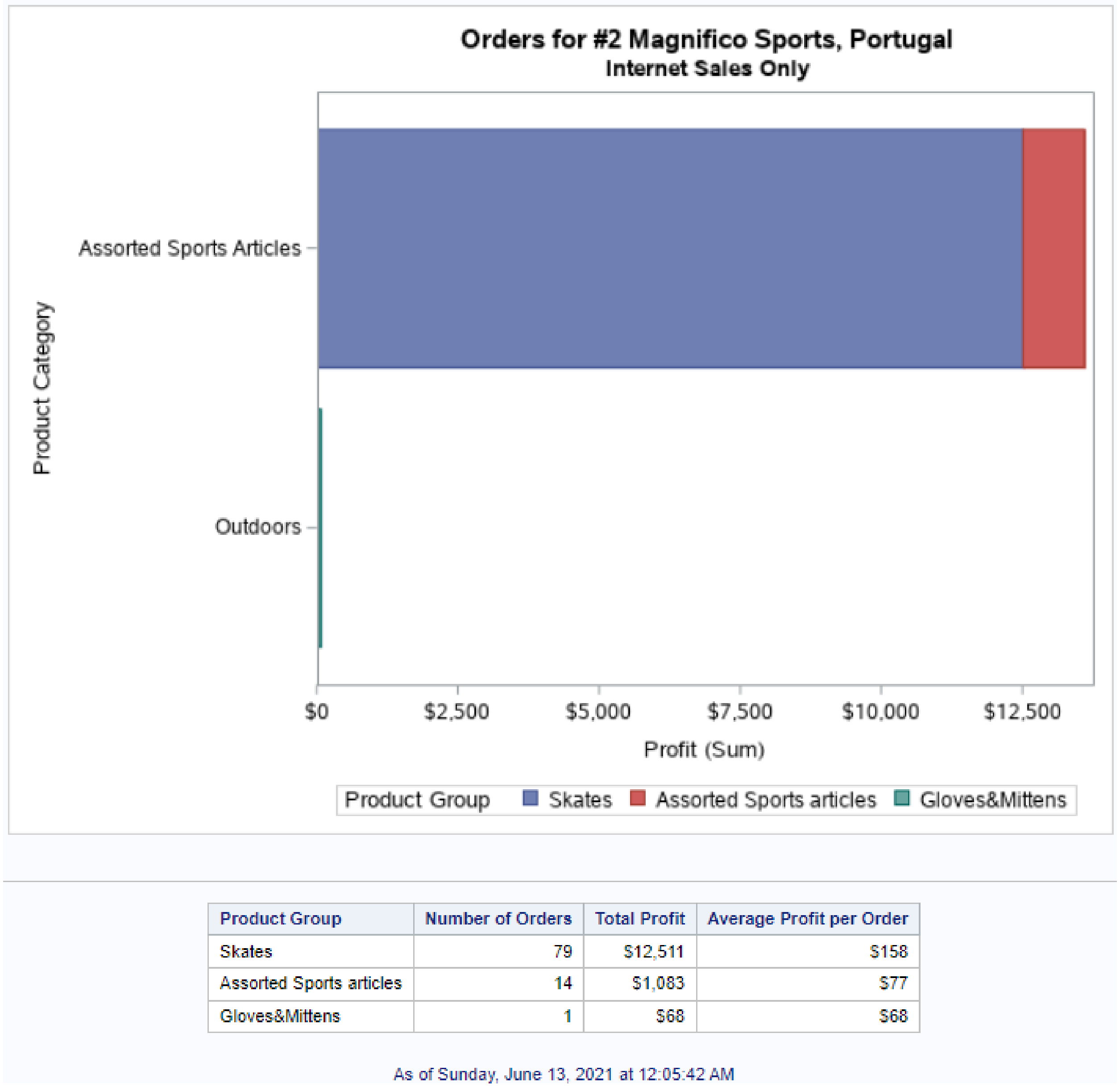


Supplier ID	Supplier Name	Supplier Country	Profit
1303	Eclipse Inc	US	19836.1
1684	Magnifico Sports	PT	13662.9
755	Top Sports	DK	11962.75
109	Petterson AB	SE	11823.95
2963	3Top Sports	US	11025.15
772	AllSeasons Outdoor Clothing	US	10060.2
13198	Twain Inc	US	8924.175
4742	Luna sastreria S.A.	ES	8809.6
4646	Mayday Inc	US	7308.63
3815	Roll-Over Inc	US	7009.25
1280	British Sports Ltd	GB	6084.55
16422	Outback Outfitters Ltd	GB	6023.65
15218	Miller Trading Inc	US	5417.4
16733	Fuller Trading Co.	CA	3998.52
14648	Top Sports Inc	US	3727.7



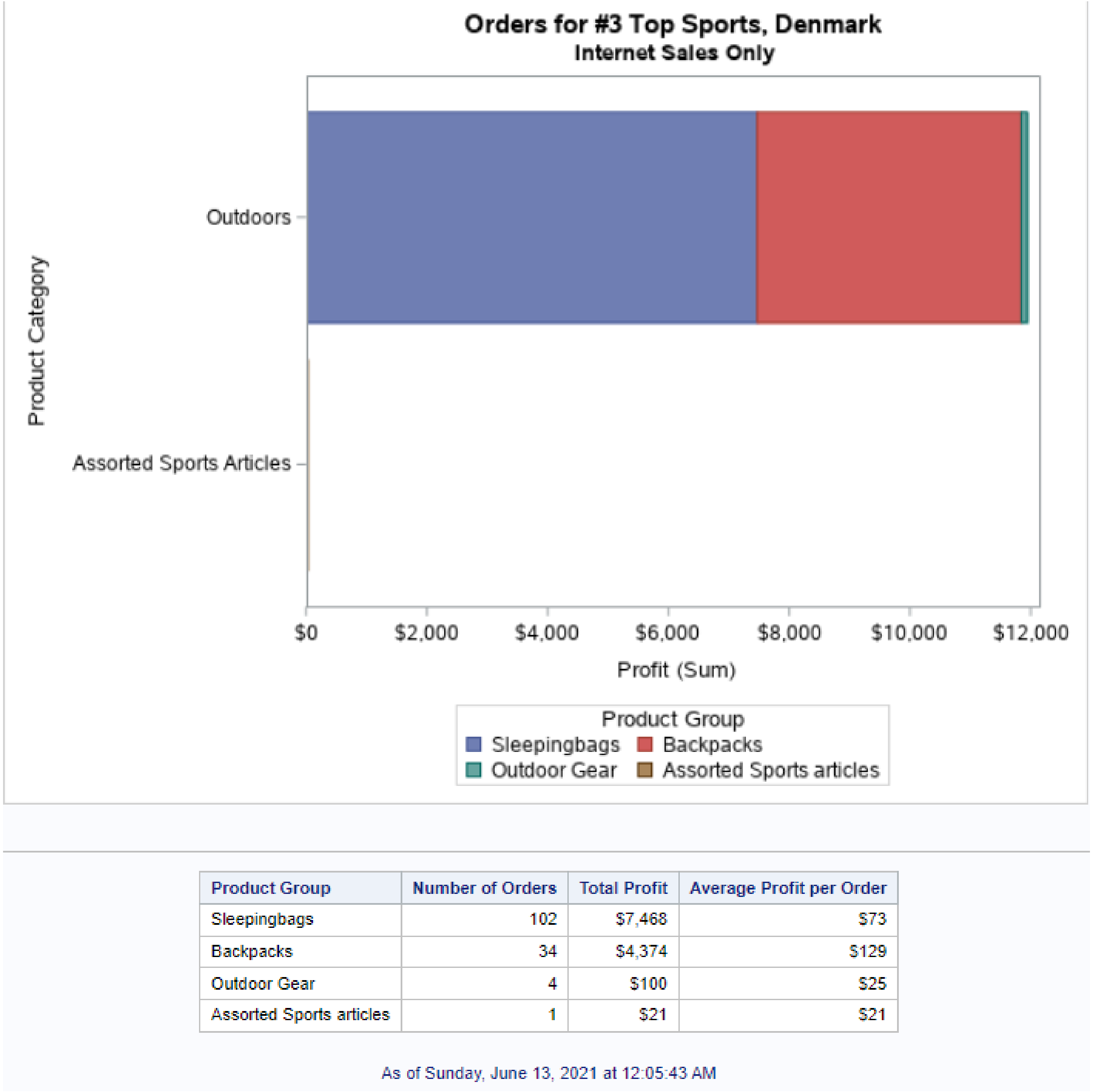


Product Group	Number of Orders	Total Profit	Average Profit per Order
Eclipse Shoes	154	$13,086	$85
Eclipse Clothing	118	$5,018	$43
Eclipse, Kid's Clothes	19	$590	$31
Eclipse, Kid's Shoes	11	$440	$40
LSF	6	$438	$73



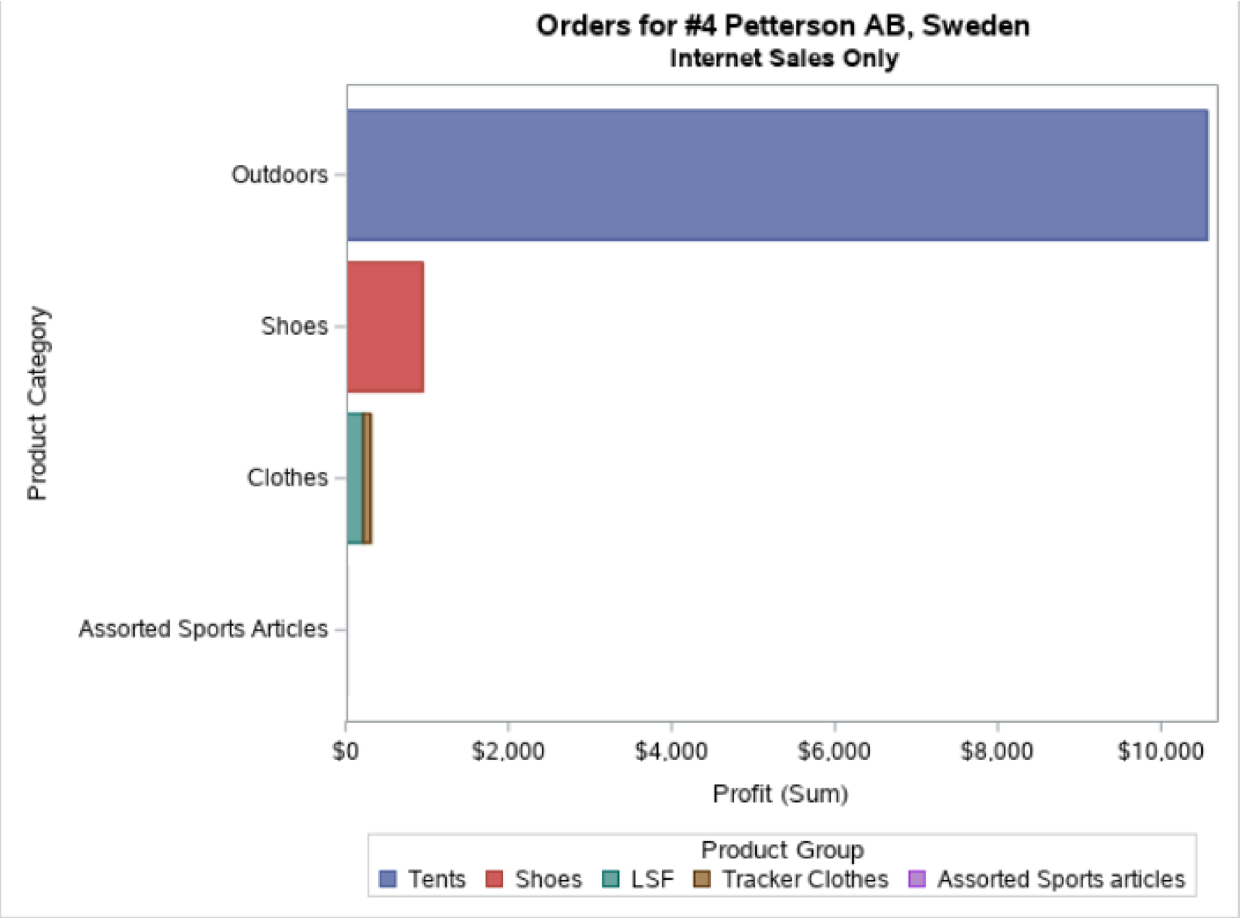
Product Group	Number of Orders	Total Profit	Average Profit per Order
Skates	79	$12,511	$158
Assorted Sports articles	14	$1,083	$77
Gloves&Mittens	1	$68	$68

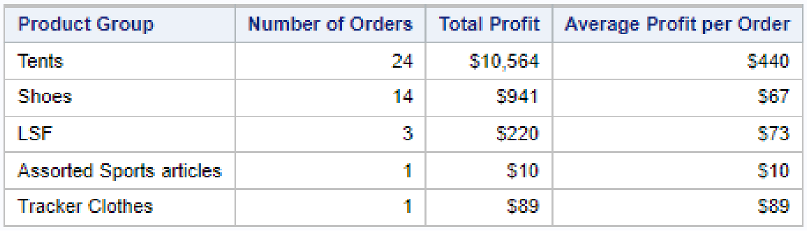
As of Sunday, June 13, 2021 at 12:05:42 AM



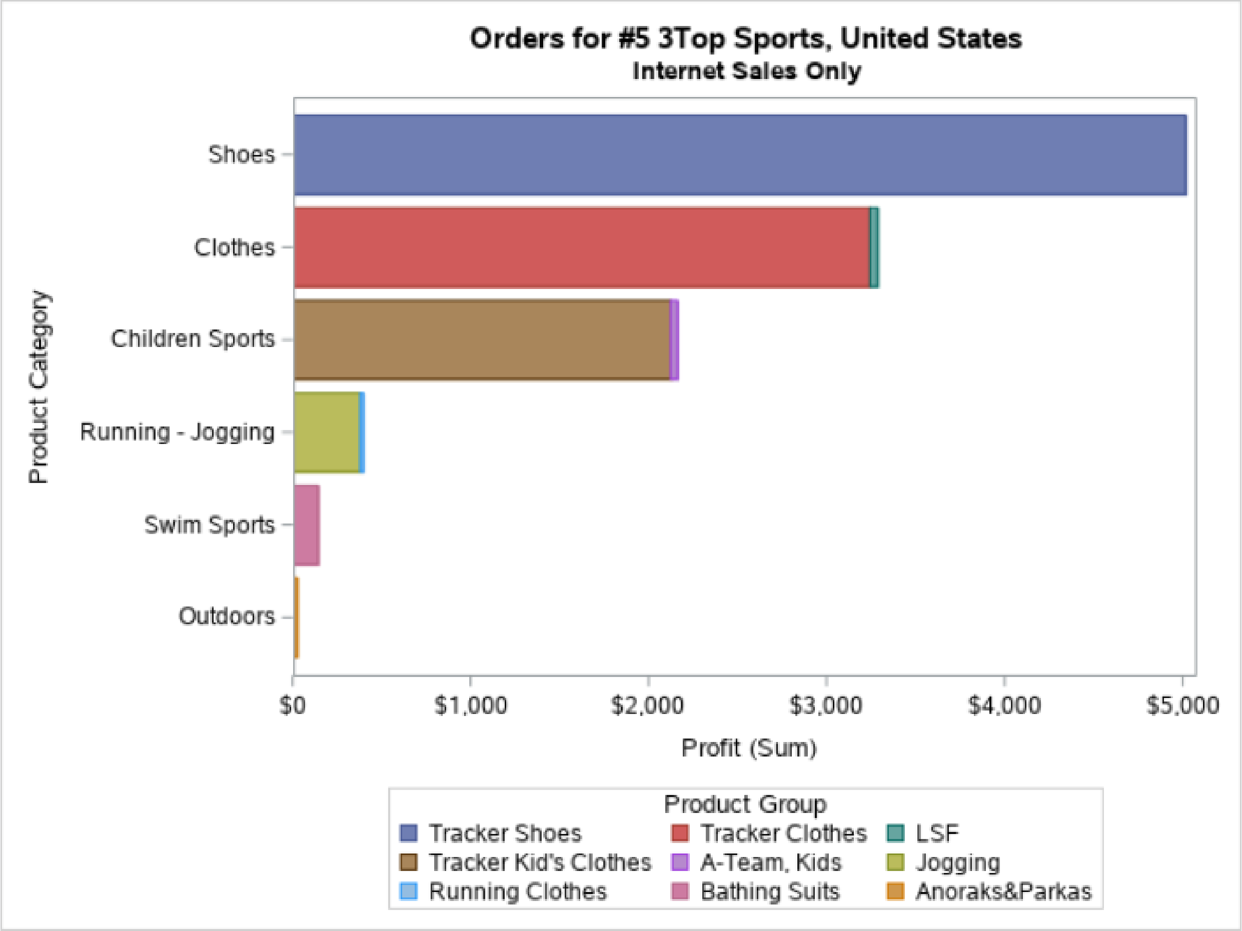
Product Group	Number of Orders	Total Profit	Average Profit per Order
Sleepingbags	102	$7,468	$73
Backpacks	34	$4,374	$129
Outdoor Gear	4	$100	$25
Assorted Sports articles	1	$21	$21

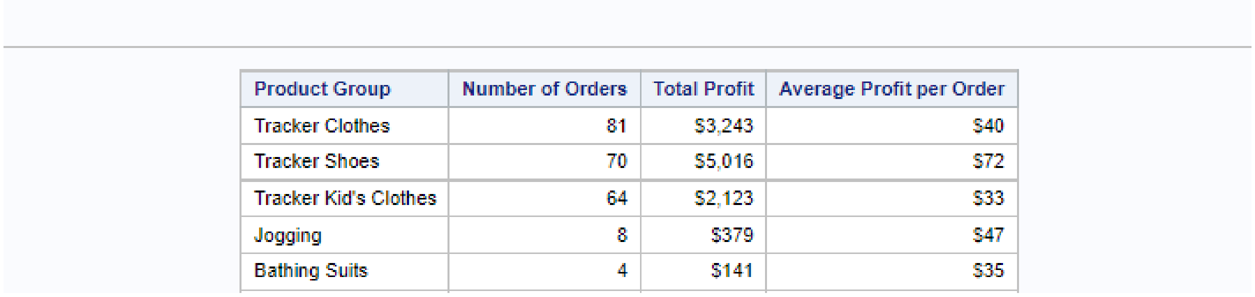
As of Sunday, June 13, 2021 at 12:05:43 AM





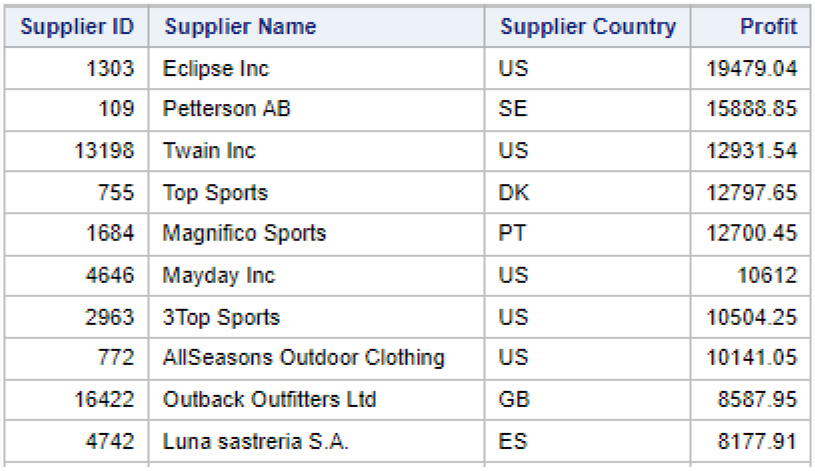
Product Group	Number of Orders	Total Profit	Average Profit per Order
Tents	24	$10,564	$440
Shoes	14	$941	$67
LSF	3	$220	$73
Assorted Sports articles	1	$10	$10
Tracker Clothes	1	$89	$89



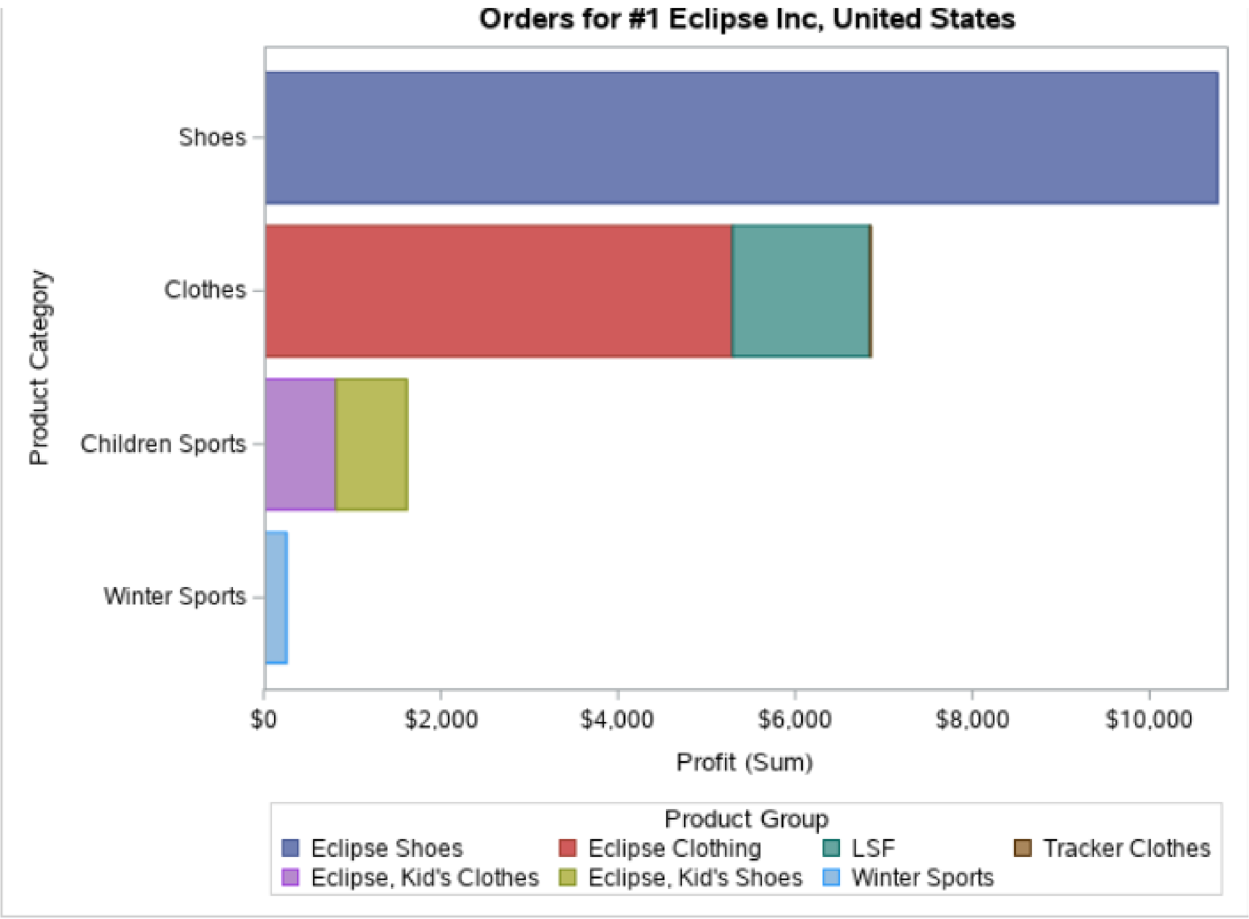


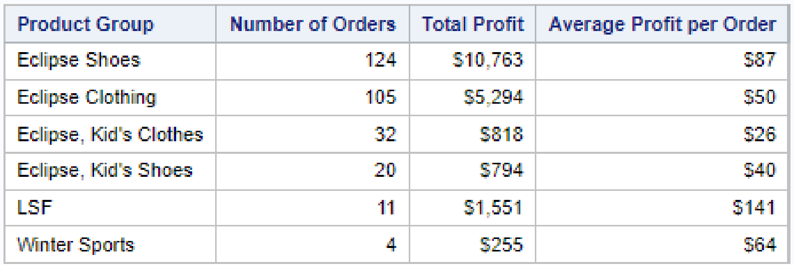
Product Group	Number of Orders	Total Profit	Average Profit per Order
Tracker Clothes	81	$3,243	$40
Tracker Shoes	70	$5,016	$72
Tracker Kid's Clothes	64	$2,123	$33
Jogging	8	$379	$47
Bathing Suits	4	$141	$35

```
%SupplierReport(3)
```

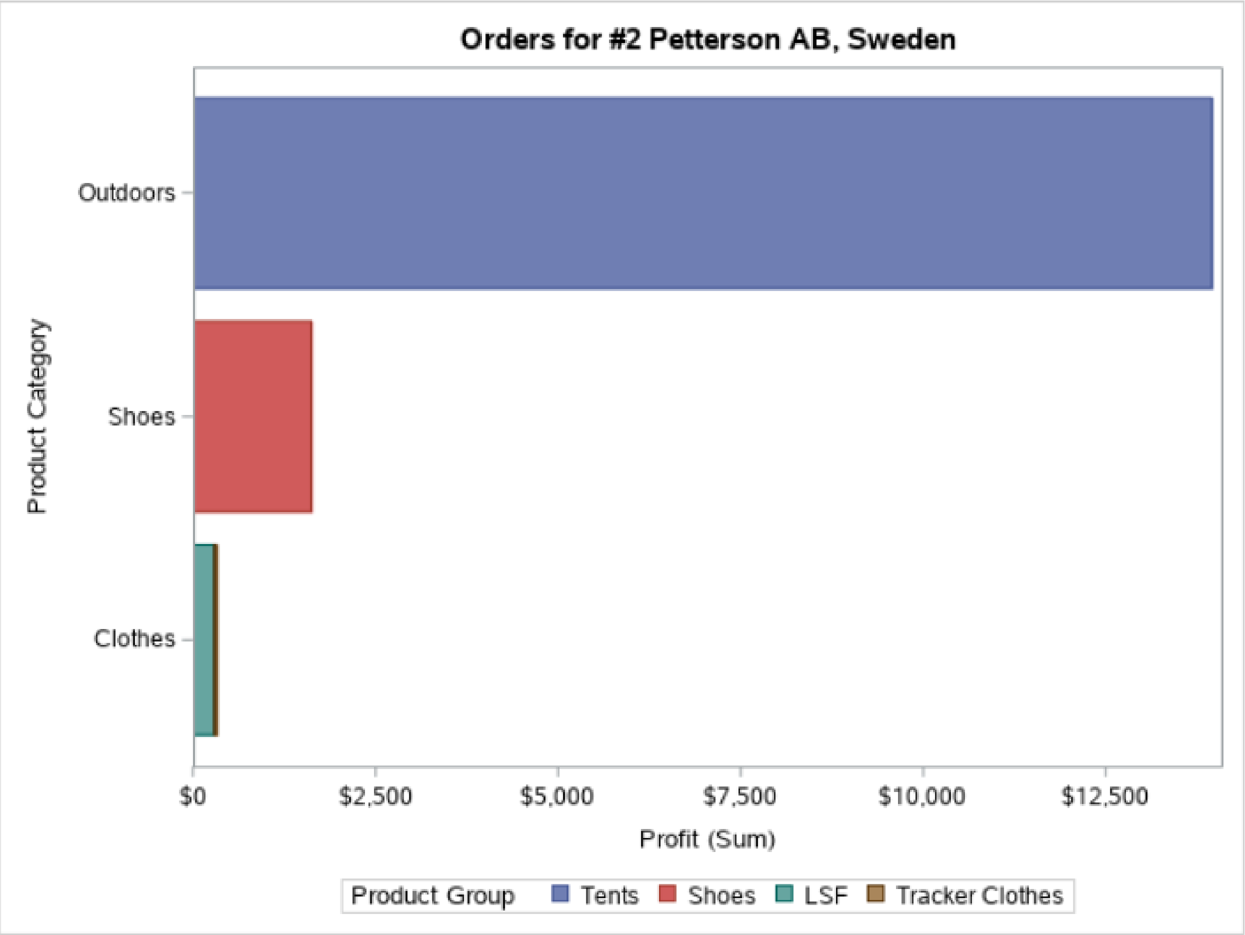


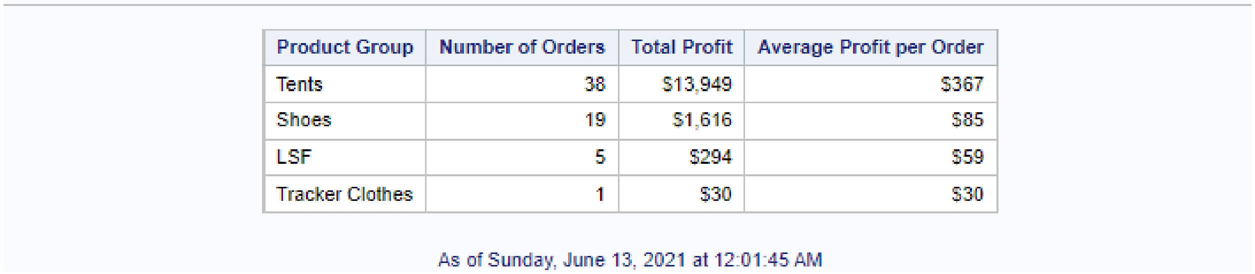
Supplier ID	Supplier Name	Supplier Country	Profit
1303	Eclipse Inc	US	19479.04
109	Petterson AB	SE	15888.85
13198	Twain Inc	US	12931.54
755	Top Sports	DK	12797.65
1684	Magnifico Sports	PT	12700.45
4646	Mayday Inc	US	10612
2963	3Top Sports	US	10504.25
772	AllSeasons Outdoor Clothing	US	10141.05
16422	Outback Outfitters Ltd	GB	8587.95
4742	Luna sastreria S.A.	ES	8177.91


Orders for #1 Eclipse Inc, United States



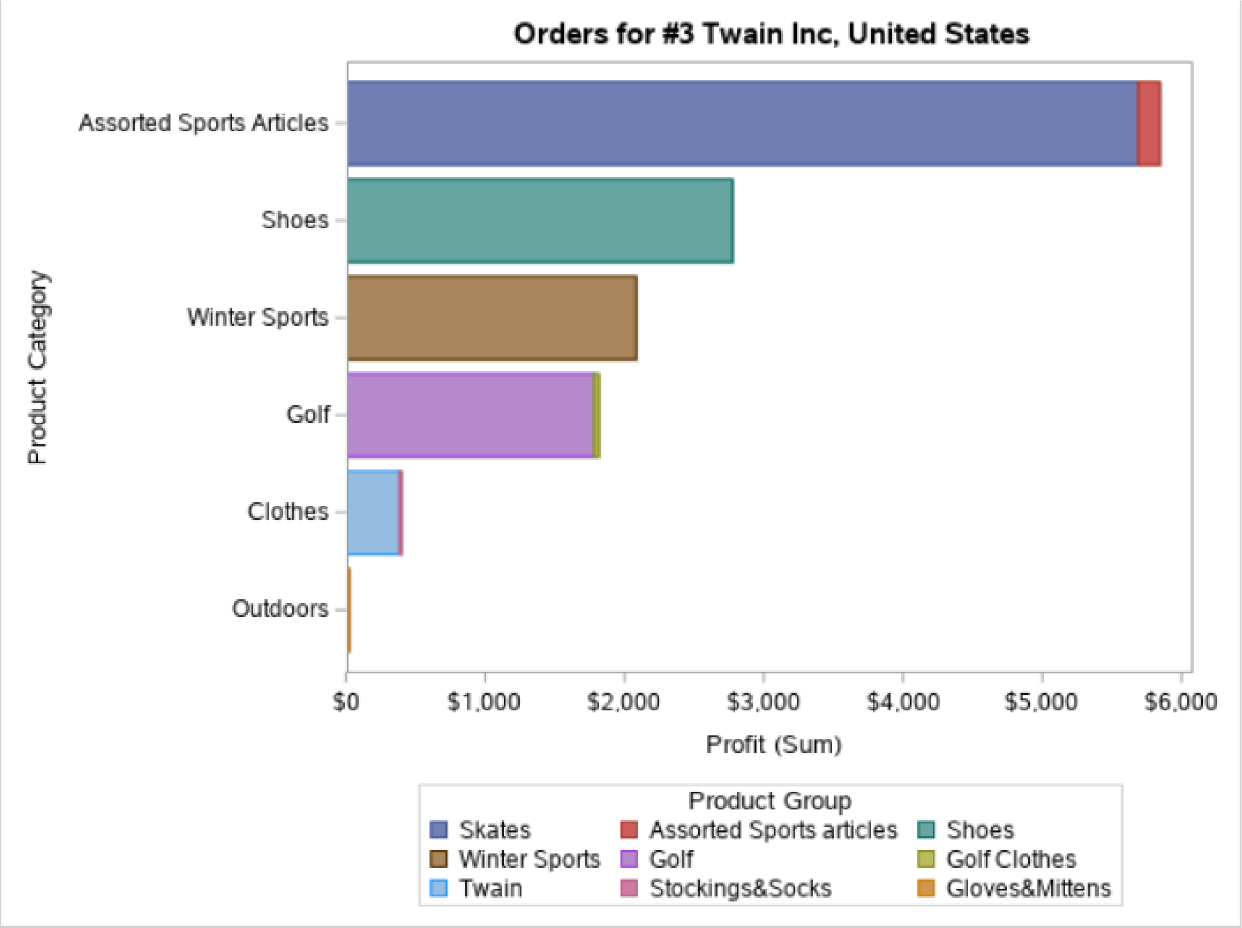
Product Group	Number of Orders	Total Profit	Average Profit per Order
Eclipse Shoes	124	$10,763	$87
Eclipse Clothing	105	$5,294	$50
Eclipse, Kid's Clothes	32	$818	$26
Eclipse, Kid's Shoes	20	$794	$40
LSF	11	$1,551	$141
Winter Sports	4	$255	$64

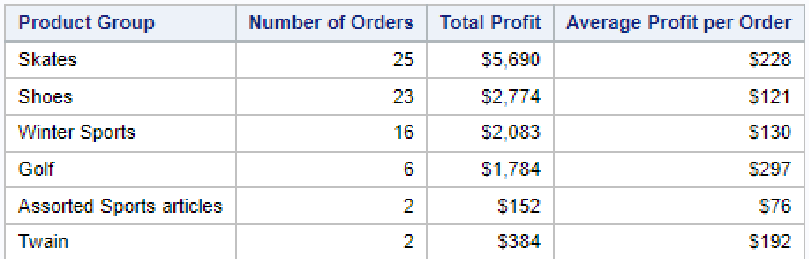




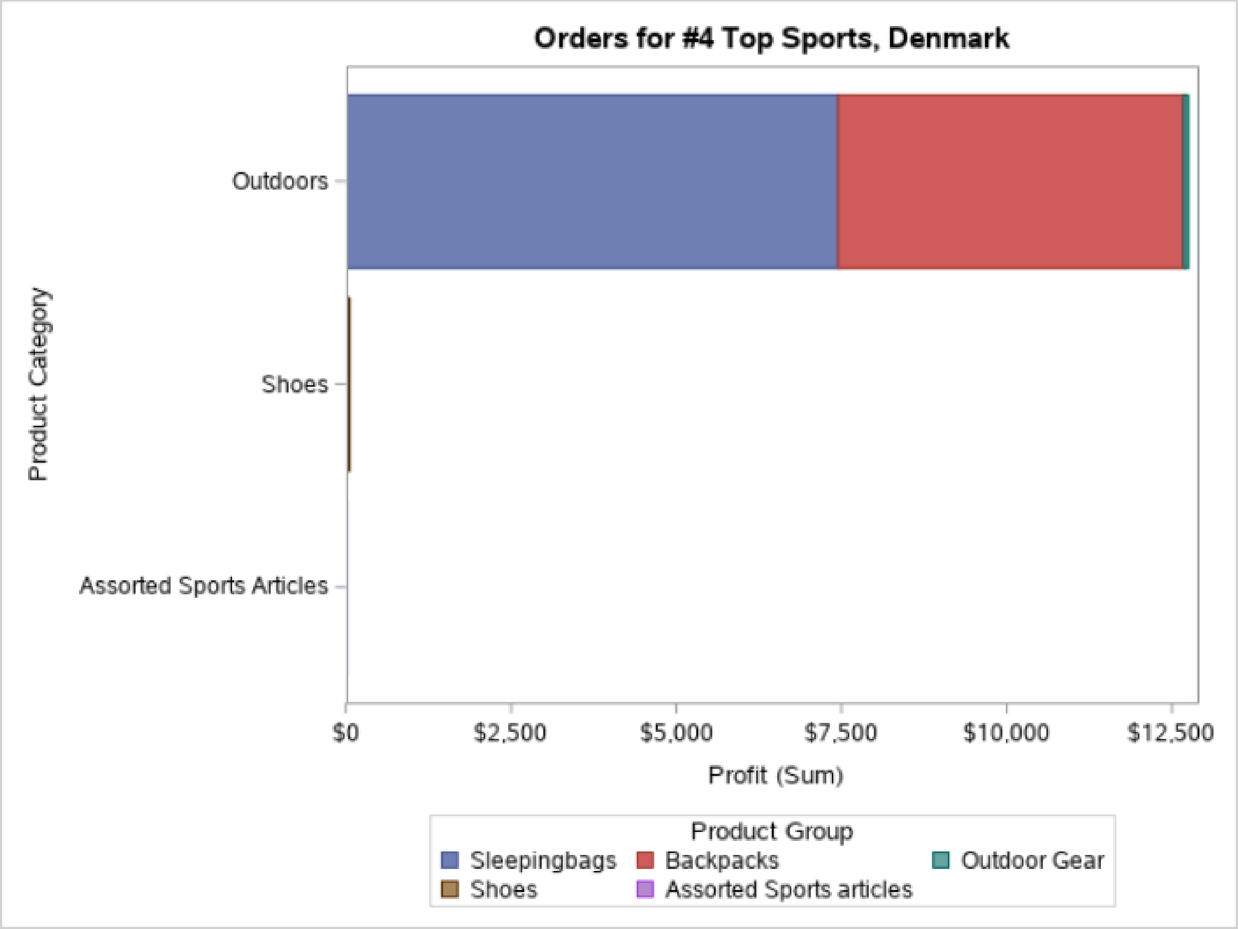
Product Group	Number of Orders	Total Profit	Average Profit per Order
Tents	38	$13,949	$367
Shoes	19	$1,616	$85
LSF	5	$294	$59
Tracker Clothes	1	$30	$30

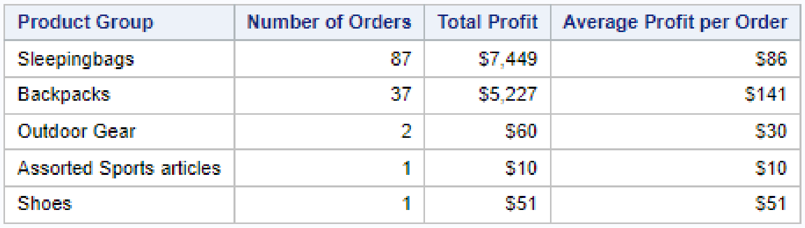
As of Sunday, June 13, 2021 at 12:01:45 AM



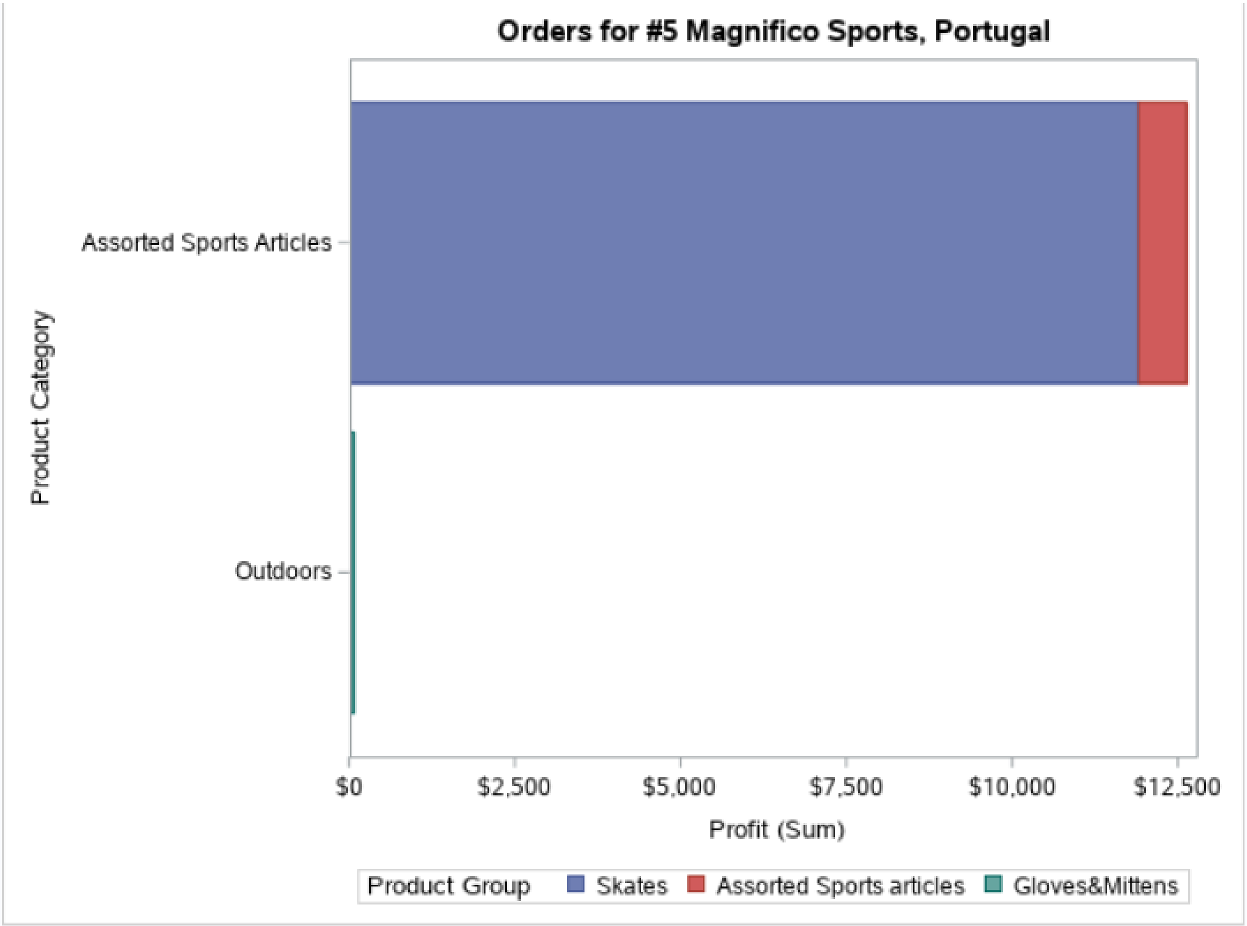


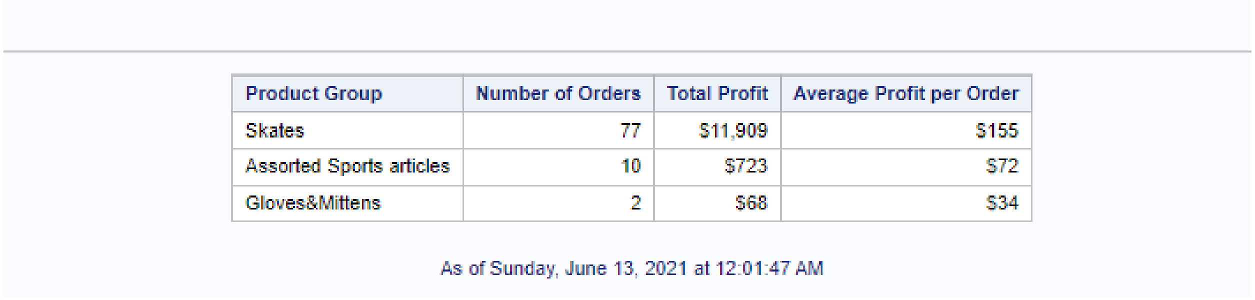
Product Group	Number of Orders	Total Profit	Average Profit per Order
Skates	25	$5,690	$228
Shoes	23	$2,774	$121
Winter Sports	16	$2,083	$130
Golf	6	$1,784	$297
Assorted Sports articles	2	$152	$76
Twain	2	$384	$192





Product Group	Number of Orders	Total Profit	Average Profit per Order
Sleepingbags	87	$7,449	$86
Backpacks	37	$5,227	$141
Outdoor Gear	2	$60	$30
Assorted Sports articles	1	$10	$10
Shoes	1	$51	$51





Product Group	Number of Orders	Total Profit	Average Profit per Order
Skates	77	$11,909	$155
Assorted Sports articles	10	$723	$72
Gloves&Mittens	2	$68	$34

As of Sunday, June 13, 2021 at 12:01:47 AM

*Solution;

```
**********************************************************
*  Concepts used:                                        *
*  1) Create a macro function, store as AUTOCALL         *
*  2) Use %SYSFUNC                                       *
*  3) Create macro with a parameter                      *
*  4) Use %IF/%THEN based on macro parameter value       *
*  5) Use %PUT to generate custom messages in log        *
*  6) Create a series of macro variables                 *
*  7) Use %DO loop and indirect macro variable reference *
**********************************************************;

*****************************************
* Be sure to first run libname.sas      *
*****************************************;

/*Create REPLACESPACE macro to replace spaces with underscores*/
/*Save in autocall folder*/
/*Solution code for REPLACESPACE macro in ReplaceSpace_Solution.sas*/

options sasautos=("&path/autocall", SASAUTOS);

/*Create macro SupplierReport to generate PDF for each of the top 5
suppliers*/

%macro supplierreport(ot) / minoperator;
%local i;
%if &ot= %then %do;
    %put ERROR: You did not specify an Order_Type code (required).;
        %put ERROR- Valid Order_Type values include 1 (retail), 2 (catalog),
or 3 (internet).;
    %put ERROR- Program will stop executing;
    %return;
%end;

%else %if not(&ot in 1 2 3) %then %do;
        %put ERROR: Valid Order_Type values include 1 (retail), 2 (catalog),
or 3 (internet).;
        %put ERROR- Program will stop executing;
        %return;
%end;

%else %do;

/*PART 1*/
/*This step creates the OrderDetail table that joins Orders with Products and
Country_Codes.
  It calculates Profit for each row and include Retail Sales (order_type=1)
only */
        proc sql;
        create table OrderDetail as
                select Order_ID, o.Product_ID, Order_Type, Product_Category,
                   Product_Group, Product_Line, Product_Name,
                   (total_retail_price-(costprice_per_unit*quantity)) as
Profit,
                   Supplier_ID, Supplier_Name, Supplier_Country
                from mc1.orders as o
                        left join mc1.products as p
                        on o.Product_ID=p.Product_ID
                where order_type=&ot;
        quit;

        /*This step summarizes profit and ranks suppliers*/
        /*Generate macro variables for top 5 suppliers ID, Name and sum of 
Profit*/
        proc sql;
        select Supplier_ID format=12.,
               Supplier_Name,
               Supplier_Country,
               sum(profit) as Profit
            into :topsupp1-:topsupp5, :name1-:name5, :country1-:country5, 
:profit1-:profit5
            from OrderDetail
            group by Supplier_ID, Supplier_Name, Supplier_Country
            order by Profit desc;
        quit;

    /*Use CALL SYMPUTX to create a series of macro variables named Country_CC 
where
          CC is the 2-letter CountryCode. Assign the corresponding CountryName 
value.*/
        data _null_;
                set mc1.country_codes;
                call symputx(cats('country_',CountryCode),CountryName);
        run;

        options nodate;
        ods graphics on / imagefmt=png;
        %do i=1 %to 5;

        ods pdf file="&path/case_study/%replacespace(&i &&name&i).pdf" 
style=meadow startpage=no nogtitle notoc;

        %let cc=&&country&i;
        title "Orders for #&i &&name&i, &&country_&cc";
        %if &ot=1 %then %do;
                title2 "Retail Sales Only";
        %end;
        %else %if &ot=2 %then %do;
                title2 "Catalog Sales Only";
        %end;
        %else %if &ot=3 %then %do;
                title2 "Internet Sales Only";
        %end;
        %else %if &ot= %then %do;
                title2 "All Sales";
        %end;

        proc sgplot data=OrderDetail;
                hbar Product_Category / response=profit stat=sum 
group=Product_Group categoryorder=respdesc;
                where Supplier_ID=&&topsupp&i;
                format profit dollar8.;
        run;

        title;
        footnote "As of %sysfunc(today(),weekdate.) at
%sysfunc(time(),timeampm.)";
        proc sql;
                select Product_Group,
                   count(order_id) as NumOrders "Number of Orders",
                   sum(profit) as TotalProfit "Total Profit" format=dollar8.,
                   avg(profit) as AvgProfit "Average Profit per Order"
format=dollar6.
                from OrderDetail
                where Supplier_ID=&&topsupp&i
                group by Product_Group
                order by calculated numorders desc;
        quit;

        footnote;
        %end;
%end;
ods pdf close;
%mend supplierreport;

*%supplierreport(1)
*%supplierreport(2)
*%supplierreport(3)
*%supplierreport(4)
*%supplierreport()
```