Sales Analysis of AdventureWorks Products

Zhongda (David) Yang

Project Overview

This project utilizes SAS to analyze sales data from the AdventureWorks dataset, focusing on product sales for 2013 and 2014. It is structured into four phases: Data Import, Data Cleaning, Joining and Merging, and Data Analysis. The process involves importing product and sales details, cleaning and preparing the data, merging related datasets for a unified analysis, and conducting in-depth analysis to uncover sales trends, performance metrics, and customer preferences. The analysis yields insights into sales by product color, the impact of unit prices on sales, and the total sales of specific product categories, enhanced with visualizations to illustrate key findings and trends.

Phase 1: Data Import

SAS code for Data Import

```
libname project '/home/u63568166/BAN130 Notebook/project';

proc import datafile='/home/u63568166/BAN130 Notebook/project/AdventureWorks(2).xlsx'
out=project.Product
dbms=xlsx;
sheet=Product;
run;

proc import datafile='/home/u63568166/BAN130 Notebook/project/AdventureWorks(2).xlsx'
out=project.salesorderdetail
dbms=xlsx;
sheet=salesorderdetail;
run;
```

In this initial phase, the SAS environment is set up with a library reference named "project", pointing to a specific directory on the server. Two sheets “Product” and “SalesOrderDetail” in the dataset “AdventureWorks(2)” were then imported into SAS using Proc Import.

Phase 2: Data Cleaning

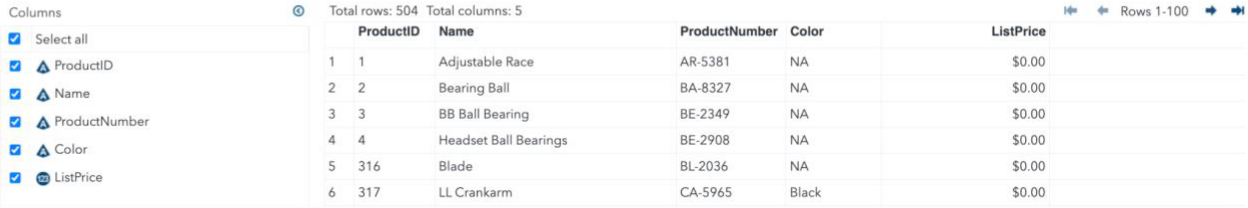
SAS code for Data Cleaning “Product”

```
data project.product_clean;
set project.product (keep=ProductID Name ProductNumber Color ListPrice);
if color=' ' then color='NA';
ListPrice_1=input(ListPrice,9.2);
format listprice_1 dollar9.2;
drop ListPrice;
rename Listprice_1=ListPrice;
run;
```

The processes completed include:

- Creating a Product_Clean dataset from the Product dataset, including only ProductID, Name, ProductNumber, Color, and ListPrice.
- Replacing all missing values in the Color column with 'NA'.
- Setting the ListPrice column format to numeric and formatting it to include a dollar sign with two decimal places.
- Dropping all unnecessary columns.

Output of Data Cleaning for “Product” is as follows:



Total rows: 504 Total columns: 5

	ProductID	Name	ProductNumber	Color	ListPrice
1	1	Adjustable Race	AR-5381	NA	$0.00
2	2	Bearing Ball	BA-8327	NA	$0.00
3	3	BB Ball Bearing	BE-2349	NA	$0.00
4	4	Headset Ball Bearings	BE-2908	NA	$0.00
5	316	Blade	BL-2036	NA	$0.00
6	317	LL Crankarm	CA-5965	Black	$0.00

SAS code for Data Cleaning “SalesOrderDetail”

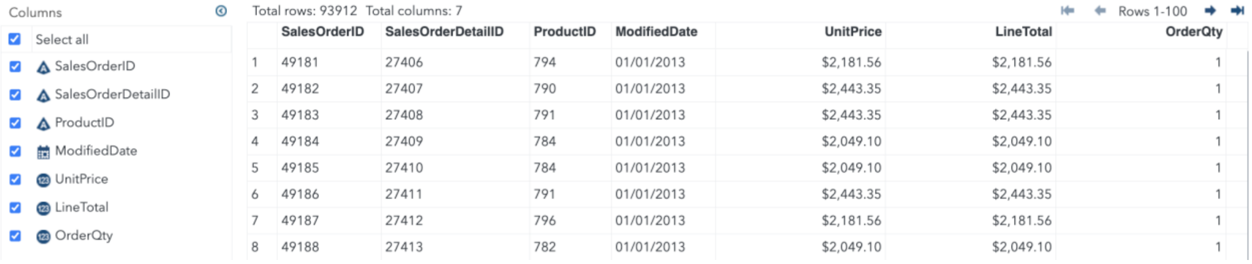
```
data project.SalesOrderDetail_Clean;
set project.salesorderdetail (keep=SalesOrderID SalesOrderDetailID OrderQty ProductID
UnitPrice LineTotal ModifiedDate);
date_part=input(substr(modifieddate,1,10),yymmdd10.);
price=input(unitprice,15.4);
line=input(linetotal,15.6);
order=input(orderqty,5.);
if year(date_part) in (2013,2014);
```

```
format date_part mmddyy10. price dollar9.2 line dollar15.2;
drop modifieddate unitprice linetotal orderqty;
rename date_part=ModifiedDate price=UnitPrice line=LineTotal order=OrderQty;
run;
```

The processes completed include:

- A SalesOrderDetail_Clean dataset was created from the SalesOrderDetail dataset, including only SalesOrderID, SalesOrderDetailID, OrderQty, ProductID, UnitPrice, LineTotal, and ModifiedDate.
- The formats of ModifiedDate, UnitPrice, LineTotal, and OrderQty columns were set to numeric.
- Only dates from the years 2013 and 2014 were included in the ModifiedDate.
- The ModifiedDate was formatted to the mmddyy10. format.
- UnitPrice and LineTotal were formatted to display a dollar sign with two decimal places.
- All unnecessary columns were dropped.

Output of Data Cleaning for "SalesOrderDetail" is as follows:



Total rows: 93912 Total columns: 7

	SalesOrderID	SalesOrderDetailID	ProductID	ModifiedDate	UnitPrice	LineTotal	OrderQty
1	49181	27406	794	01/01/2013	$2,181.56	$2,181.56	1
2	49182	27407	790	01/01/2013	$2,443.35	$2,443.35	1
3	49183	27408	791	01/01/2013	$2,443.35	$2,443.35	1
4	49184	27409	784	01/01/2013	$2,049.10	$2,049.10	1
5	49185	27410	784	01/01/2013	$2,049.10	$2,049.10	1
6	49186	27411	791	01/01/2013	$2,443.35	$2,443.35	1
7	49187	27412	796	01/01/2013	$2,181.56	$2,181.56	1
8	49188	27413	782	01/01/2013	$2,049.10	$2,049.10	1

Phase 3: Joining and Merging

SAS code for SalesDetails

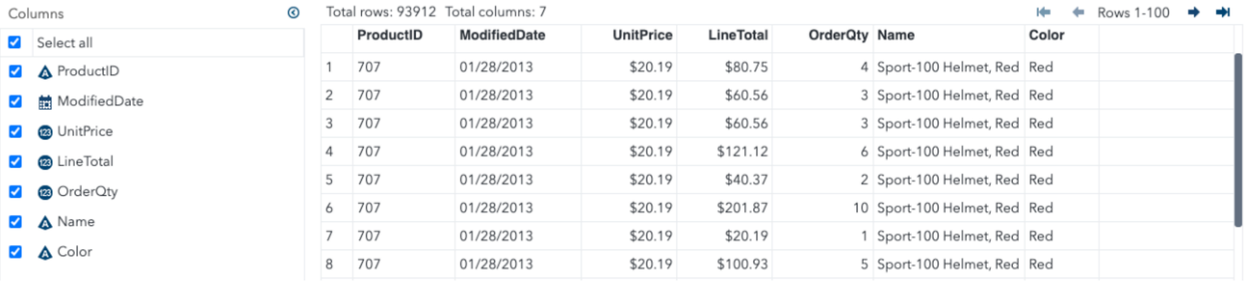
```
proc sort data=project.product_clean;
by productID;
run;

proc sort data=project.SalesOrderDetail_Clean;
by productID;
run;

data project.SalesDetails;
merge project.SalesOrderDetail_Clean(in=a) project.Product_Clean(in=b);
by ProductID;
```

```
if a;
drop SalesOrderID SalesOrderDetailID ProductNumber ListPrice;
run;
```

In this step, we combined two preprocessed datasets from Phase 2, using ‘ProductID’ as the key variable. To complete the merge, we first sorted each dataset by ‘ProductID’. Next, we used the MERGE function in a data step to create the ‘SalesDetails’ dataset. We positioned ‘a’ (representing ‘SalesOrderDetail_Clean’) after the IF function to achieve a left join with Product_Clean. Subsequently, we dropped four columns that were not relevant to our analysis. The resulting dataset is outlined below.



☑ Select all
☑ ProductID
☑ ModifiedDate
☑ UnitPrice
☑ LineTotal
☑ OrderQty
☑ Name
☑ Color

Total rows: 93912 Total columns: 7

	ProductID	ModifiedDate	UnitPrice	LineTotal	OrderQty	Name	Color
1	707	01/28/2013	$20.19	$80.75	4	Sport-100 Helmet, Red	Red
2	707	01/28/2013	$20.19	$60.56	3	Sport-100 Helmet, Red	Red
3	707	01/28/2013	$20.19	$60.56	3	Sport-100 Helmet, Red	Red
4	707	01/28/2013	$20.19	$121.12	6	Sport-100 Helmet, Red	Red
5	707	01/28/2013	$20.19	$40.37	2	Sport-100 Helmet, Red	Red
6	707	01/28/2013	$20.19	$201.87	10	Sport-100 Helmet, Red	Red
7	707	01/28/2013	$20.19	$20.19	1	Sport-100 Helmet, Red	Red
8	707	01/28/2013	$20.19	$100.93	5	Sport-100 Helmet, Red	Red

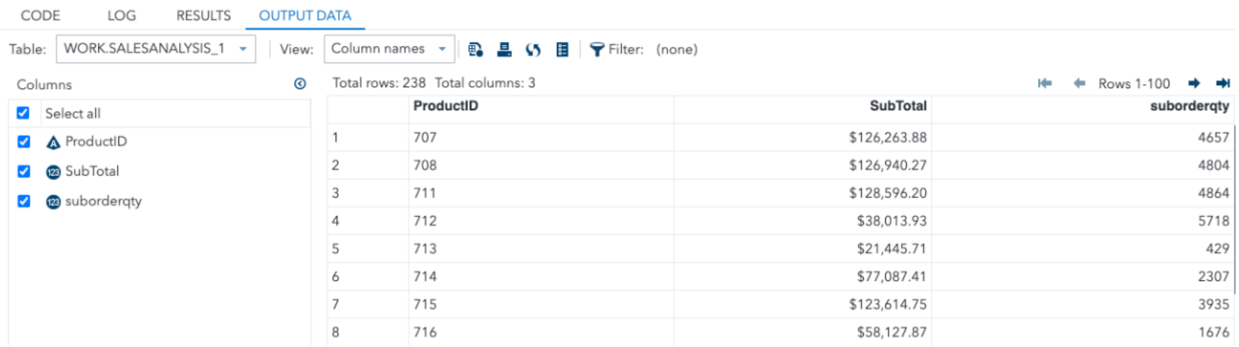
SAS code for SalesAnalysis

```
proc means data=project.SalesDetails noprint;
by productid;
output out=SalesAnalysis_1(drop= _TYPE_ _FREQ_) sum(linetotal)=SubTotal
sum(orderqty)=suborderqty;
run;

data salesanalysis_2;
set project.SalesDetails;
by productid;
if first.productid;
run;

data project.salesanalysis;
merge SalesAnalysis_2 SalesAnalysis_1;
by productid;
run;
```

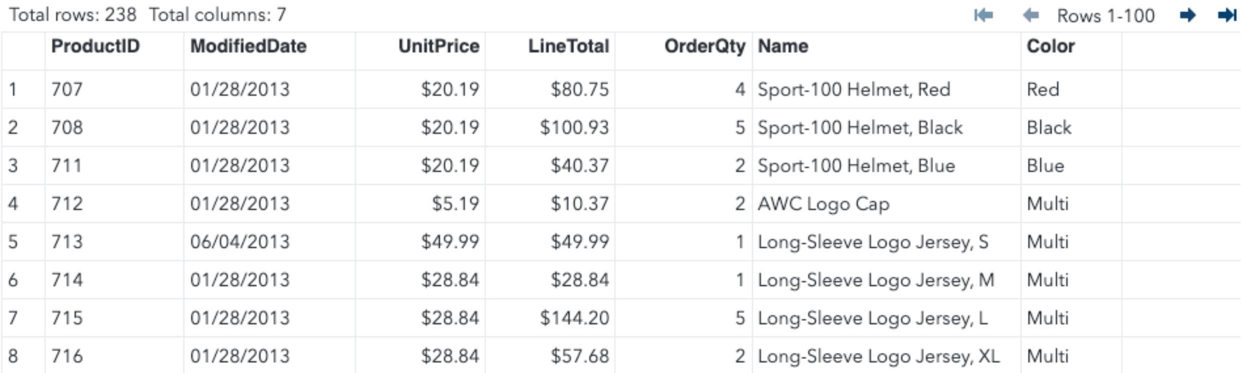
Grouping the ‘SalesDetails’ dataset by ‘ProductID’ variable and calculating the total sales and quantity for each group can be easily achieved in SQL using the GROUP BY function. However, in SAS we need to complete this in 3 steps. First we employed ‘PROC MEANS’ to obtain ‘SubTotal’ and ‘SubOrderQty’ for each ‘ProductID’. The resulting output, named ‘SalesAnalysis_1’, is presented below.



Total rows: 238 Total columns: 3

	ProductID	SubTotal	suborderqty
1	707	$126,263.88	4657
2	708	$126,940.27	4804
3	711	$128,596.20	4864
4	712	$38,013.93	5718
5	713	$21,445.71	429
6	714	$77,087.41	2307
7	715	$123,614.75	3935
8	716	$58,127.87	1676

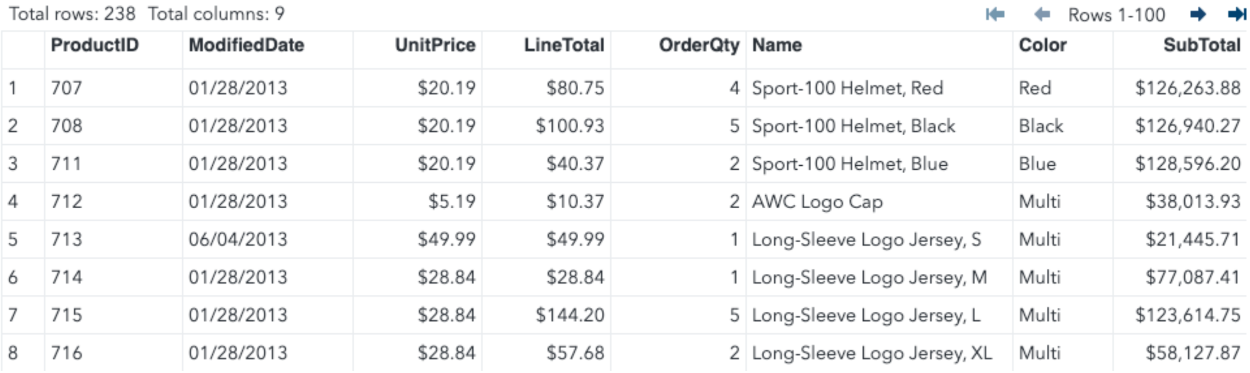
Next, from ‘SalesDetails’ we extracted the first entry of each ‘ProductID’. This step aims to align the number of rows in the newly created dataset ‘SalesAnalysis_2’ with ‘SalesAnalysis_1’. We could also choose the last entry or any other within each ‘ProductID’ group, as we are focusing on the total value per group. Variables with multiple values in a group, like ‘OderQty’, are not pertinent to our analysis in this context. ‘SalesAnalysis_2’ is as follows.



Total rows: 238 Total columns: 7

	ProductID	ModifiedDate	UnitPrice	LineTotal	OrderQty	Name	Color
1	707	01/28/2013	$20.19	$80.75	4	Sport-100 Helmet, Red	Red
2	708	01/28/2013	$20.19	$100.93	5	Sport-100 Helmet, Black	Black
3	711	01/28/2013	$20.19	$40.37	2	Sport-100 Helmet, Blue	Blue
4	712	01/28/2013	$5.19	$10.37	2	AWC Logo Cap	Multi
5	713	06/04/2013	$49.99	$49.99	1	Long-Sleeve Logo Jersey, S	Multi
6	714	01/28/2013	$28.84	$28.84	1	Long-Sleeve Logo Jersey, M	Multi
7	715	01/28/2013	$28.84	$144.20	5	Long-Sleeve Logo Jersey, L	Multi
8	716	01/28/2013	$28.84	$57.68	2	Long-Sleeve Logo Jersey, XL	Multi

Finally, we merged ‘SalesAnalysis_1’ and ‘SalesAnalysis_2’ by ‘ProductID’ and obtained ‘SalesAnalysis’ dataset, which is the base of our subsequent analysis.



Total rows: 238 Total columns: 9

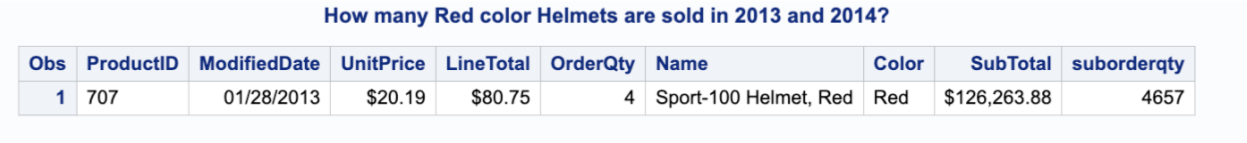
	ProductID	ModifiedDate	UnitPrice	LineTotal	OrderQty	Name	Color	SubTotal
1	707	01/28/2013	$20.19	$80.75	4	Sport-100 Helmet, Red	Red	$126,263.88
2	708	01/28/2013	$20.19	$100.93	5	Sport-100 Helmet, Black	Black	$126,940.27
3	711	01/28/2013	$20.19	$40.37	2	Sport-100 Helmet, Blue	Blue	$128,596.20
4	712	01/28/2013	$5.19	$10.37	2	AWC Logo Cap	Multi	$38,013.93
5	713	06/04/2013	$49.99	$49.99	1	Long-Sleeve Logo Jersey, S	Multi	$21,445.71
6	714	01/28/2013	$28.84	$28.84	1	Long-Sleeve Logo Jersey, M	Multi	$77,087.41
7	715	01/28/2013	$28.84	$144.20	5	Long-Sleeve Logo Jersey, L	Multi	$123,614.75
8	716	01/28/2013	$28.84	$57.68	2	Long-Sleeve Logo Jersey, XL	Multi	$58,127.87

Phase 4: Data Analysis

SAS code for Question: How many Red color Helmets are sold in 2013 and 2014?

```
data project.red_helmet;
set project.salesanalysis;
where upcase(color)='RED' and upcase(name) like '%HELMET%';
run;

title 'How many Red color Helmets are sold in 2013 and 2014?';
proc print data=project.red_helmet;
run;
```



How many Red color Helmets are sold in 2013 and 2014?

Obs	ProductID	ModifiedDate	UnitPrice	LineTotal	OrderQty	Name	Color	SubTotal	suborderqty
1	707	01/28/2013	$20.19	$80.75	4	Sport-100 Helmet, Red	Red	$126,263.88	4657

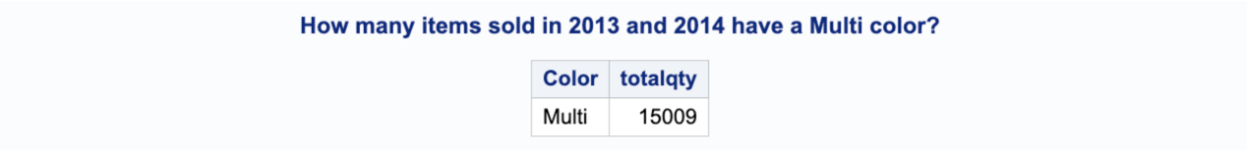
In order to identify the count of Red color Helmets sold in the years 2013 and 2014, we first filtered the dataset ‘SalesAnalysis’ by selecting rows where the color is 'RED' (regardless of case) and the product name contains the term 'HELMET'. This filtered data is then stored in a new dataset named ‘Red_Helmet’ within the project library. Later we utilized the PROC PRINT procedure to display the details of the ‘Red_Helmet’ dataset, where SubOrderQty represented the number of Red color Helmets that were sold in 2013 and 2014, which is 4657.

SAS code for Question: How many items sold in 2013 and 2014 have a Multi color?

```
data project.multi_color;
set project.salesanalysis;
where upcase(color)='MULTI';
totalqty+suborderqty;
run;

proc sort data=project.multi_color;
by descending totalqty;
run;

title 'How many items sold in 2013 and 2014 have a Multi color?';
proc print data=project.multi_color(obs=1 keep=color totalqty) noobs;
run;
```



How many items sold in 2013 and 2014 have a Multi color?

Color	totalqty
Multi	15009

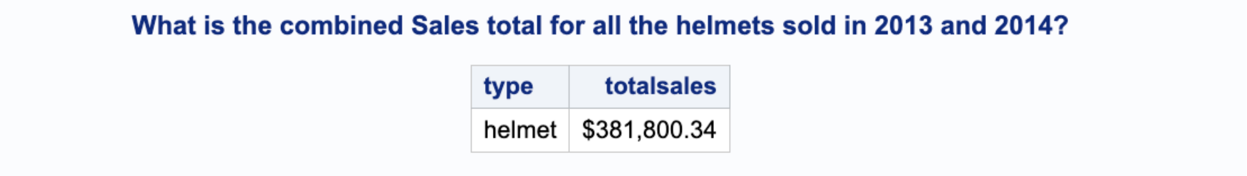
By using the dataset ‘SalesAnalysis’ again, we filtered the data to include only rows where the color is 'MULTI' (regardless of case). And a cumulative total of the SubOrderQty is calculated as TotalQty for answering the question. Subsequently, the dataset is sorted in descending order based on the total quantity. Later we utilized the PROC PRINT procedure to display the first observation of the ‘Multi_Color’ dataset, showcasing the color and total quantity of Multi color items sold in 2013 and 2014, which is 15009.

SAS code for Question: What is the combined Sales total for all the helmets sold in 2013 and 2014?

```
data project.helmet_sales;
set project.salesanalysis;
where upcase(name) like '%HELMET%';
type='helmet';
totalsales+subtotal;
format totalsales dollar12.2;
run;

proc sort data=project.helmet_sales;
by descending totalsales;
run;

title 'What is the combined Sales total for all the helmets sold in 2013 and 2014?';
proc print data=project.helmet_sales (obs=1 keep=type totalsales) noobs;
run;
```



What is the combined Sales total for all the helmets sold in 2013 and 2014?

type	totalsales
helmet	$381,800.34

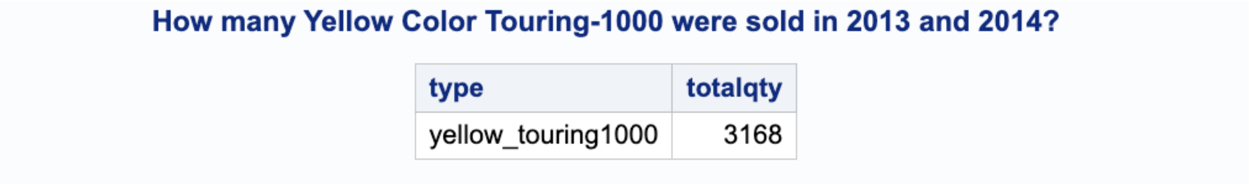
Again, we created a dataset named ‘Helmet_Sales’ within the project library by filtering the ‘SalesAnalysis’ dataset and selecting only rows where the product name includes the term 'HELMET' in any case. We introduced a new variable, type, assigned as 'helmet,' and accumulated the SubTotal sales into the TotalSales variable. To enhance readability, the TotalSales variable is formatted with a dollar sign and two decimal places. Subsequently, the dataset is sorted in descending order based on the total sales. The PROC PRINT procedure is used to display the first observation of the ‘Helmet_Sales’ dataset, providing the combined sales total for all helmets sold in 2013 and 2014, which is $381,800.34.

SAS code for Question: How many Yellow Color Touring-1000 were sold in 2013 and 2014?

```
data project.yellow_touring1000;
set project.salesanalysis;
where upcase(color)='YELLOW' and upcase(name) like '%TOURING-1000%';
type='yellow_touring1000';
totalqty+suborderqty;
run;

proc sort data=project.yellow_touring1000;
by descending totalqty;
run;

title 'How many Yellow Color Touring-1000 where sold in 2013 and 2014?';
proc print data=project.yellow_touring1000(obs=1 keep=type totalqty) noobs;
run;
```



How many Yellow Color Touring-1000 were sold in 2013 and 2014?

type	totalqty
yellow_touring1000	3168

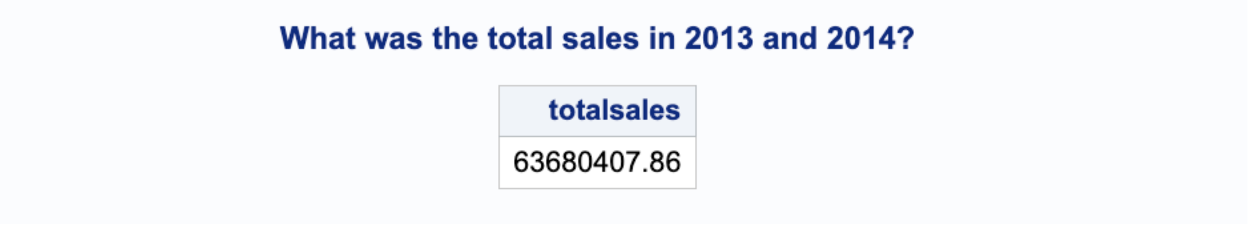
Similar approach was adopted here, which we created a dataset named ‘Yellow_Touring1000’ within the project library by filtering the ‘SalesAnalysis’ dataset and selecting rows where the color is 'YELLOW' and the product name contains 'TOURING-1000' in any case. A new variable, type, is introduced and assigned as 'yellow_touring1000.' The SubOrderQty values are accumulated into the TotalQty variable. Following this, the dataset is sorted in descending order based on the total quantity sold. PROC PRINT procedure is employed to display the first observation of the ‘Yellow_Touring1000’ dataset, and answered the question that 3168 Yellow Color Touring-1000 were sold in 2013 and 2014.

SAS code for Question: What was the total sales in 2013 and 2014?

```
data project.total_sales;
set project.salesanalysis;
totalsales+subtotal;
run;

proc sort data=project.total_sales;
by descending totalsales;
run;
```

```
title 'What was the total sales in 2013 and 2014?';
proc print data=project.total_sales (obs=1 keep=totalsales) noobs;
run;
```



What was the total sales in 2013 and 2014?

totalsales
63680407.86

For the last question in phase 4, we also created a dataset named ‘Total_Sales’ by extracting and filtering relevant information from the dataset ‘SalesAnalysis’. The variable TotalSales was utilized to accumulate the total sales, incremented by the subtotal for each observation. Following this, we organized the dataset in descending order based on the TotalSales variable. PROC PRINT procedure is employed to display the first observation of the ‘Total_Sales’ dataset, that a total sales of $63680408.96 in 2013 and 2014 was shown.

Chart1 (Which color got the highest sales in the sales of 2013 and 2014?)
SAS Code

```
proc sort data=project.SalesAnalysis;
by Color;
run;

proc means data=project.SalesAnalysis noprint;
by Color;
output out=Popular_Color (drop= _TYPE_ _FREQ_) sum(SubTotal)=TotalSales
sum(SubOrderQty)=TotalQty;
run;

title 'Which color got the highest sales in the sales of 2013 and 2014?';
proc sgplot data=Popular_Color;
vbar Color / response=TotalSales datalabel datalabelattrs=(weight=bold) fillattrs=(color=grey);
run;
```

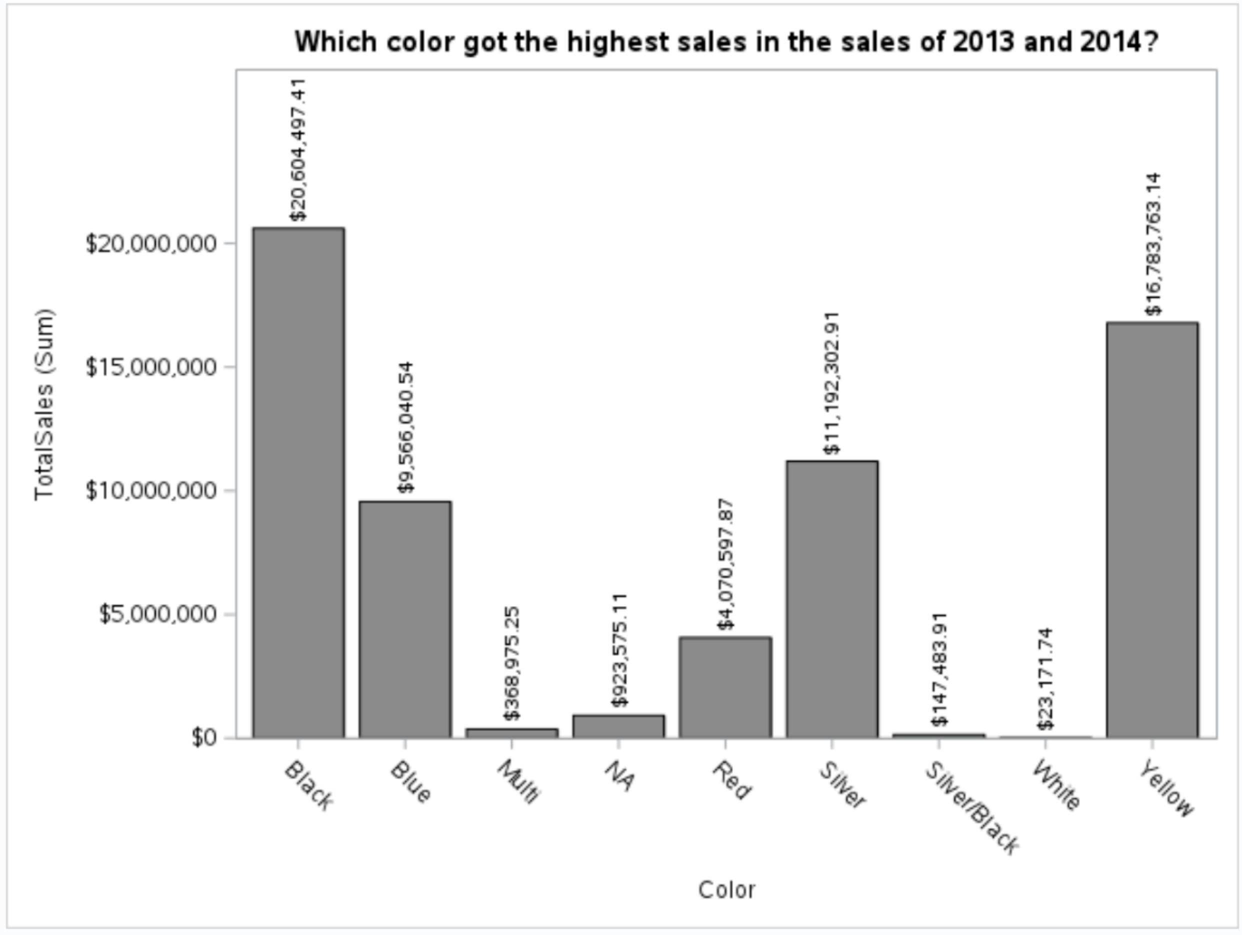
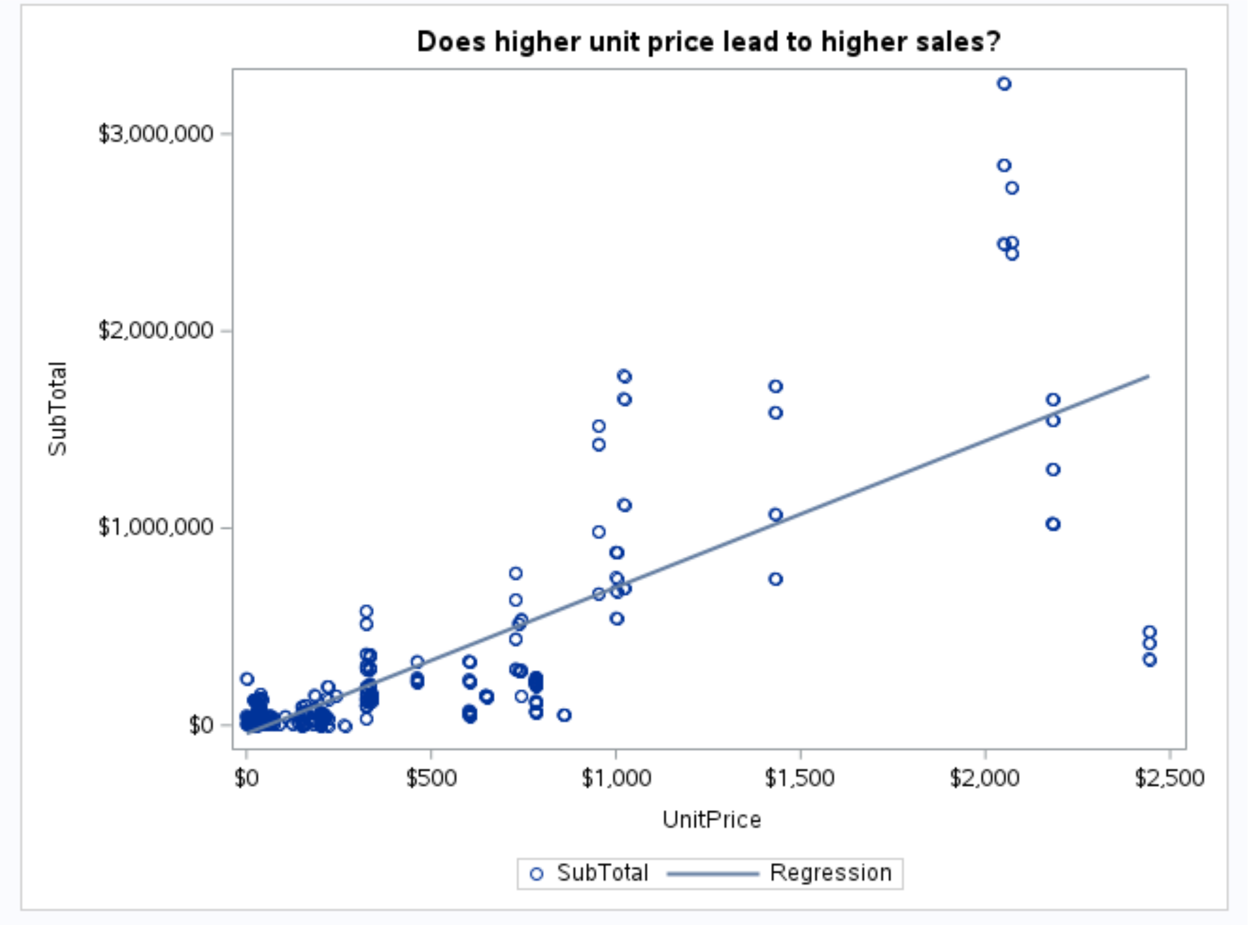


Chart 1 aims to identify and visualize the color that achieved the highest sales in the years 2013 and 2014. The process begins with sorting the ‘SalesAnalysis’ dataset by the Color variable using the PROC SORT procedure. Subsequently, the PROC MEANS procedure is employed to compute the total sales (TotalSales) and total quantity (TotalQty) for each color group. The results are stored in a new dataset named ‘Popular_Color’. Finally, the PROC SGPLOT procedure generates a vertical bar chart (vbar) based on the Color variable, with the height of each bar representing the total sales. Data labels are added to the bars, and the chart is filled with a grey color for better visibility.

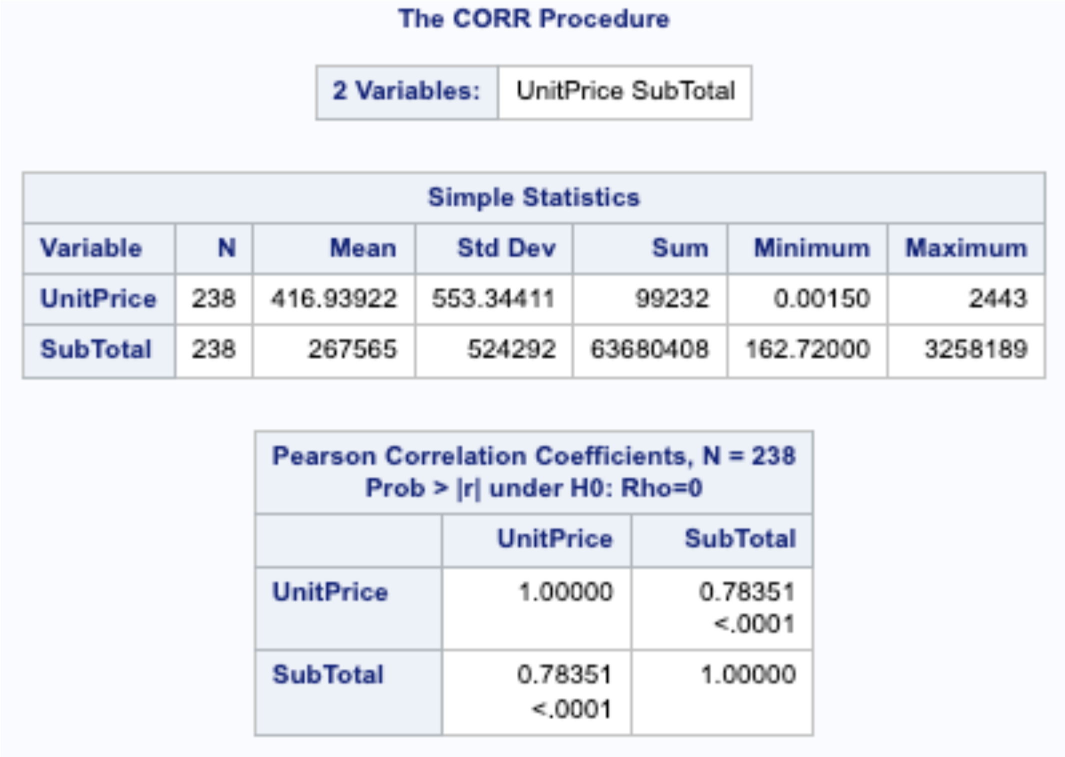
Chart2 (Does higher unit price lead to higher sales?)
SAS code

```
title 'Does higher unit price lead to higher sales?';
proc sgplot data=project.salesanalysis;
scatter x=unitprice y=subtotal;
reg x=unitprice y=subtotal;
run;
```



We want to explore the relationship between total sales and unit price of a product. The scatterplot with a regression line we created suggests a positive correlation between these two variables. To verify this observation, we also conducted a ‘PROC CORR’ analysis.

```
proc corr data=project.salesanalysis;
var UnitPrice SubTotal;
run;
```



The CORR Procedure

2 Variables:	UnitPrice SubTotal

Simple Statistics						
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
UnitPrice	238	416.93922	553.34411	99232	0.00150	2443
SubTotal	238	267565	524292	63680408	162.72000	3258189

Pearson Correlation Coefficients, N = 238 Prob > |r| under H0: Rho=0		
	UnitPrice	SubTotal
UnitPrice	1.00000	0.78351 <.0001
SubTotal	0.78351 <.0001	1.00000

Given that the correlation coefficient is 0.78351 and p-value is less than 0.0001, we can conclude that there exists a strong positive correlation between two variables. This indicates that higher unit prices are generally associated with increased total sales.

Conclusion

The AdventureWorks Sales Analysis project successfully navigated through extensive data manipulation and analysis to unearth valuable insights into the sales performance of AdventureWorks products in 2013 and 2014. Through a structured approach that included data import, cleaning, merging, and rigorous analysis, we were able to highlight significant trends and patterns that can inform strategic decisions.

Key findings from the analysis include:

- Red Helmets Popularity: A notable demand for red helmets, underscoring the importance of color in consumer preferences.
- Multi-Color Items Sales: Items with multi-color options exhibited significant sales volumes, suggesting that product variety may enhance appeal.

- Impact of Unit Price on Sales: A strong positive correlation between unit price and total sales was identified, indicating that higher-priced items do not deter sales, possibly reflecting a perception of higher quality or value among consumers.
- Sales Trends by Color: The analysis of sales by color revealed which colors are more popular among consumers, offering insights into potential inventory and marketing strategies.
- Performance of Specific Products: Detailed insights into specific products, like the Yellow Color Touring-1000, provided targeted data on consumer preferences and product performance.

These insights not only offer a snapshot of past performance but also provide a valuable foundation for predictive modeling and future strategy development. By understanding the dynamics of sales in relation to product features such as color, price, and type, AdventureWorks can tailor its inventory, marketing, and sales strategies to better meet consumer demands and enhance profitability.