

BAN130: Programming for Analytics

Adventure Works Product Sales Analysis

BAN130: Programming for Analytics

Introduction

The SAS Online Demand Studio platform is used to develop the Adventure Works Product Sales Analysis Project Report. It is divided into four parts: data import, cleaning, joining, merging, and data analysis.

Output and results focus on information gathered from the output obtained by writing SAS programs by Group 10. To launch, we imported the data from the provided Excel file into SAS using Proc Import. The SalesOrderDetail and Product worksheet in the Excel file is imported as the SalesOrderDetail and Product dataset respectively in SAS.



Dataset Description:

This Project report includes analysis carried out on two datasets:

Product

SalesOrderDetail

Both the datasets are present in AdventureWorks (xlsx) Excel file

Product Dataset :

Column Name	Description
ProductID	Unique product identification number
Name	Name of the product
ProductNumber	Product Number
MakeFlag	0 = Product is purchased, 1 = Product is manufactured in-house.
FinishedGoodsFlag	0 = Product is not a salable item. 1 = Product is salable.

BAN130: Programming for Analytics

Color	Product Color
SafetyStockLevel	Minimum inventory quantity
ReorderPoint	Inventory level that triggers a purchase order or work order
StandardCost	Standard cost of the product
ListPrice	Listed Price of Product
Size	Selling price
SizeUnitMeasureCode	Product size
WeightUnitMeasureCode	Unit of measure for Size column
Weight	Product weight
DaysToManufacture	Number of days required to manufacture the product
ProductLine	R = Road, M = Mountain, T = Touring, S = Standard
Class	H = High, M = Medium, L = Low
Style	W = Womens, M = Mens, U = Universal
ProductSubcategoryID	Unique subcategory ID of the Product
ProductModelID	Product is a member of this product subcategory. Foreign key to ProductSubCategory.ProductSubCategoryID.
SellStartDate	Product is a member of this product model. Foreign key to ProductModel.ProductModelID.
SellEndDate	Date the product was available for sale.
DiscontinuedDate	Date the product was no longer available for sale.
rowguid	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
ModifiedDate	Date and time the record was last updated.

BAN130: Programming for Analytics

SalesOrderDetail Dataset :

Column Name	Description
SalesOrderID	Primary key. Foreign key to SalesOrderHeader.SalesOrderID.
SalesOrderDetailID	Primary key. One incremental unique number per product sold.
CarrierTrackingNumber	Shipment tracking number supplied by the shipper.
OrderQty	Quantity ordered per product.
ProductID	Unique ID defining the product
SpecialOfferID	Promotional code. Foreign key to SpecialOffer.SpecialOfferID.
UnitPrice	Selling price of a single product.
UnitPriceDiscount	Discount amount
LineTotal	Per product subtotal. Computed as $\text{UnitPrice} * (1 - \text{UnitPriceDiscount}) * \text{OrderQty}$.
rowguid	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample
ModifiedDate	Date and time the record was last updated.

BAN130: Programming for Analytics

1. Data Import

```
libname mylib '/home/u55491482/sasuser.v94';

proc import out=mylib.Product

datafile="/home/u55491482/sasuser.v94/AdventureWorks.xlsx"

dbms=xlsx;

sheet = 'Product';

getnames= yes;

run;


proc import out=mylib.SalesOrderDetail

datafile="/home/u55491482/sasuser.v94/AdventureWorks.xlsx"

dbms=xlsx;

sheet = 'SalesOrderDetail';

getnames= yes;

run;
```

2. Data Cleaning

This phase requires you to clean your data before the data analysis phase.

§ Product_Clean:

- Create a Product_Clean dataset from Product dataset by bringing in only ProductID, Name, ProductNumber, Color and ListPrice

BAN130: Programming for Analytics

Code:

```
title "Product_Clean Dataset with only 5 Variables";

data mylib.Product_Clean;

    set mylib.Product (keep= ProductID Name ProductNumber Color ListPrice);

run;

proc print data=mylib.Product_Clean(obs=10);

run;
```

Output:

Product_Clean Dataset with only 5 Variables					
Obs	ProductID	Name	ProductNumber	Color	ListPrice
1	1	Adjustable Race	AR-5381		0
2	2	Bearing Ball	BA-8327		0
3	3	BB Ball Bearing	BE-2349		0
4	4	Headset Ball Bearings	BE-2908		0
5	316	Blade	BL-2036		0
6	317	LL Crankarm	CA-5965	Black	0
7	318	ML Crankarm	CA-6738	Black	0
8	319	HL Crankarm	CA-7457	Black	0
9	320	Chainring Bolts	CB-2903	Silver	0
10	321	Chainring Nut	CN-6137	Silver	0

BAN130: Programming for Analytics

- All the missing values in Color column should be replaced by 'NA'

Code:

```
title "Missing Values replaced by 'NA'";  
data mylib.Product_Clean;  
    set mylib.Product (keep= ProductID Name ProductNumber Color ListPrice);  
    if missing(Color) then do Color= 'NA';  
end;  
run;  
  
proc print data=mylib.Product_Clean(obs=10);  
run;
```

Output:

Missing Values replaced by 'NA'					
Obs	ProductID	Name	ProductNumber	Color	ListPrice
1	1	Adjustable Race	AR-5381	NA	0
2	2	Bearing Ball	BA-8327	NA	0
3	3	BB Ball Bearing	BE-2349	NA	0
4	4	Headset Ball Bearings	BE-2908	NA	0
5	316	Blade	BL-2036	NA	0
6	317	LL Crankarm	CA-5965	Black	0
7	318	ML Crankarm	CA-6738	Black	0
8	319	HL Crankarm	CA-7457	Black	0
9	320	Chainring Bolts	CB-2903	Silver	0
10	321	Chainring Nut	CN-6137	Silver	0

BAN130: Programming for Analytics

- ListPrice column should be numeric (final column name should be ListPrice) and format should have a dollar sign with 2 decimal places.

Code:

```
title "ListPrice from Character to Numeric";

data mylib.Product_Clean;

set mylib.Product(keep= ProductID Name ProductNumber Color ListPrice);

if missing(Color) then Color = 'NA';

ListPrice_num = input(ListPrice,8.);

drop ListPrice;

rename ListPrice_num=ListPrice;

run;


data mylib.Product_Clean;

set mylib.Product_Clean;

informat ListPrice 8.2;

format ListPrice dollar8.2;

label ListPrice="ListPrice";

run;


proc contents data=mylib.Product_Clean varnum;

run;


proc print data=mylib.Product_Clean (obs=10);

run;
```


BAN130: Programming for Analytics

Output:

ListPrice from Character to Numeric

The CONTENTS Procedure

Before

Variables in Creation Order						
#	Variable	Type	Len	Format	Informat	Label
1	ProductID	Char	3	\$3.	\$3.	ProductID
2	Name	Char	32	\$32.	\$32.	Name
3	ProductNumber	Char	10	\$10.	\$10.	ProductNumber
4	Color	Char	12	\$12.	\$12.	Color
5	ListPrice	Char	7	\$7.	\$7.	ListPrice

After

Variables in Creation Order						
#	Variable	Type	Len	Format	Informat	Label
1	ProductID	Char	3	\$3.	\$3.	ProductID
2	Name	Char	32	\$32.	\$32.	Name
3	ProductNumber	Char	10	\$10.	\$10.	ProductNumber
4	Color	Char	12	\$12.	\$12.	Color
5	ListPrice	Num	8	DOLLAR8.2	8.2	ListPrice

- No unnecessary columns should be part of the Product_Clean dataset. Please see below expected output.

Output:

Project 130.sas x

CODE LOG RESULTS OUTPUT DATA

Table: MYLIB.PRODUCT_CLEAN View: Column names Filter: (none)

Columns

☒ Select all

☒ ProductID

☒ Name

☒ ProductNumber

☒ Color

☒ ListPrice

Property Value

Label

Name

Length

Type

Format

Informat

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

7

318

ML Crankarm

CA-6738

Black

\$0.00

8

319

HL Crankarm

CA-7457

Black

\$0.00

9

320

Chainring Bolts

CB-2903

Silver

\$0.00

10

321

Chainring Nut

CN-6137

Silver

\$0.00

11

322

Chainring

CR-7833

Black

\$0.00

12

323

Crown Race

CR-9981

NA

\$0.00

13

324

Chain Stays

CS-2812

NA

\$0.00

14

325

Decal 1

DC-8732

NA

\$0.00

15

326

Decal 2

DC-9824

NA

\$0.00

16

327

Down Tube

DT-2377

NA

\$0.00

17

328

Mountain End Caps

EC-M092

NA

\$0.00

18

329

Road End Caps

EC-R098

NA

\$0.00

Messages: 1 User: u59416473

BAN130: Programming for Analytics

§ SalesOrderDetail_Clean:

- Create SalesOrderDetail_Clean dataset from SalesOrderDetail dataset by bringing in only SalesOrderID SalesOrderDetailID OrderQty ProductID UnitPrice LineTotal and ModifiedDate
- ModifiedDate should be numeric with column name ModifiedDate
- UnitPrice should be numeric with column name UnitPrice
- LineTotal should be numeric with column name LineTotal
- OrderQty should be numeric with column name OrderQty
- Include date for year 2013 and 2014 in ModifiedDate only
- ModifiedDate should be mmddyy10. Format
- UnitPrice and LineTotal should have a dollar with 2 decimal places
- No unnecessary columns should be part of the SalesOrderDetail_Clean dataset. Please see expected output below:

Code:

```
title "SalesOrderDetail_Clean Dataset";  
data mylib.SalesOrderDetail_Clean;  
set mylib.SalesOrderDetail(keep= SalesOrderID SalesOrderDetailID OrderQty ProductID UnitPrice  
LineTotal ModifiedDate);  
ModifiedDate_num = input(ModifiedDate,anydtdte21.);  
UnitPrice_num = input(UnitPrice,8.);  
LineTotal_num = input(LineTotal,8.);  
OrderQty_num = input(OrderQty,8.);  
drop ModifiedDate UnitPrice LineTotal OrderQty;  
rename ModifiedDate_num=ModifiedDate  
UnitPrice_num=UnitPrice  
LineTotal_num=LineTotal  
OrderQty_num=OrderQty;
```

BAN130: Programming for Analytics

```
run;

data mylib.SalesOrderDetail_Clean label;
set mylib.SalesOrderDetail_Clean;

informat ModifiedDate anydtdte21. UnitPrice 8. LineTotal 8. OrderQty 8.;

format ModifiedDate mmddyy10. UnitPrice dollar10.2 LineTotal dollar10.2 OrderQty 8.;

label ModifiedDate="ModifiedDate"

UnitPrice="UnitPrice"

LineTotal="LineTotal"

OrderQty="OrderQty";

where year(ModifiedDate) in (2013,2014);

run;
```

Output:

The screenshot shows the SAS Output Data window for 'Project 130.sas'. The table is named 'MYLIB.SALESORDERDETAIL_CLEAN' and contains 93912 rows and 7 columns. The columns are: SalesOrderID, SalesOrderDetailID, ProductID, ModifiedDate, UnitPrice, LineTotal, and OrderQty. The table is displayed with a grid view, showing rows 1 through 100. The first 10 rows are visible, showing data for various sales orders and products. The table is filtered by 'none'.

	SalesOrderID	SalesOrderDetailID	ProductID	ModifiedDate	UnitPrice	LineTotal
1	49181	27406	794	01/01/2013	\$2,181.56	\$2,181.56
2	49182	27407	790	01/01/2013	\$2,443.35	\$2,443.35
3	49183	27408	791	01/01/2013	\$2,443.35	\$2,443.35
4	49184	27409	784	01/01/2013	\$2,049.10	\$2,049.10
5	49185	27410	784	01/01/2013	\$2,049.10	\$2,049.10
6	49186	27411	791	01/01/2013	\$2,443.35	\$2,443.35
7	49187	27412	796	01/01/2013	\$2,181.56	\$2,181.56
8	49188	27413	782	01/01/2013	\$2,049.10	\$2,049.10
9	49189	27414	769	01/01/2013	\$782.99	\$782.99
10	49190	27415	798	01/01/2013	\$1,000.44	\$1,000.44
11	49191	27416	800	01/02/2013	\$1,000.44	\$1,000.44
12	49192	27417	800	01/02/2013	\$1,000.44	\$1,000.44
13	49193	27418	783	01/02/2013	\$2,049.10	\$2,049.10
14	49194	27419	766	01/02/2013	\$782.99	\$782.99
15	49195	27420	759	01/02/2013	\$782.99	\$782.99
16	49196	27421	780	01/02/2013	\$2,071.42	\$2,071.42
17	49197	27422	783	01/02/2013	\$2,049.10	\$2,049.10
18	49198	27423	770	01/02/2013	\$782.99	\$782.99

BAN130: Programming for Analytics

3. Joining and Merging

- This phase requires you to join / merge your datasets to create a dataset for analysis.

§ SalesDetails:

- Create a SalesDetails dataset by joining SalesOrderDetail_Clean and Product_Clean datasets
- Use ProductID column for joining the tables
- SalesDetails table should contain all the observations from SalesOrderDetail_Clean table along with columns from Product_Clean
- Drop SalesOrderID SalesOrderDetailID ProductNumber and ListPrice from the result dataset. Please see expected output below:

Code:

```
proc sort data=mylib.salesorderdetail_clean;
by ProductID;
run;

proc sort data=mylib.Product_Clean;
by ProductID;
run;

data mylib.SalesDetails;
merge mylib.SalesOrderDetail_Clean(in=SalesOrderDetail_Clean)
      mylib.Product_Clean(in=Product_Clean);
by ProductID;
if SalesOrderDetail_Clean=1 and Product_Clean=1;
```

BAN130: Programming for Analytics

```
drop SalesOrderID SalesOrderDetailID ProductNumber ListPrice;  
  
run;  
  
title "SalesDetails Dataset";  
proc print data=mylib.SalesDetails (obs=5);  
run;
```

Output:

SalesDetails Dataset							
Obs	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

§ SalesAnalysis:

- Create a SalesAnalysis dataset from SalesDetails dataset that groups all the products by ProductID (hint: research on obtaining a total for each by group in SAS)
- Create a SubTotal column in SalesAnalysis that provides an aggregate sum of each product by its ProductID.
- SubTotal column should have a dollar and 2 decimal places.

BAN130: Programming for Analytics

Code:

```
proc sql;
create table mylib.SalesAnalysis as
select ProductID, ModifiedDate,
max(UnitPrice) as UnitPrice format dollar12.2,
sum(LineTotal) as LineTotal format dollar12.2,
sum(OrderQty) as OrderQty format 8.,
Name,
Color,
sum(LineTotal) as SubTotal format dollar12.2
from mylib.SalesDetails group by ProductID,ModifiedDate,Name,Color;
quit;
```

Output:

	ProductID	ModifiedDate	UnitPrice	LineTotal	OrderQty	Name	Color	SubTotal
1	784	01/01/2013	\$2,049.10	\$4,098.20	2	Mountain-200 Black, 46	Black	\$4,098.20
2	782	01/01/2013	\$2,049.10	\$2,049.10	1	Mountain-200 Black, 38	Black	\$2,049.10
3	798	01/01/2013	\$1,000.44	\$1,000.44	1	Road-550-W Yellow, 40	Yellow	\$1,000.44
4	791	01/01/2013	\$2,443.35	\$4,886.70	2	Road-250 Red, 52	Red	\$4,886.70
5	769	01/01/2013	\$782.99	\$782.99	1	Road-650 Black, 48	Black	\$782.99
6	796	01/01/2013	\$2,181.56	\$2,181.56	1	Road-250 Black, 58	Black	\$2,181.56
7	790	01/01/2013	\$2,443.35	\$2,443.35	1	Road-250 Red, 48	Red	\$2,443.35
8	794	01/01/2013	\$2,181.56	\$2,181.56	1	Road-250 Black, 48	Black	\$2,181.56
9	770	01/02/2013	\$782.99	\$782.99	1	Road-650 Black, 52	Black	\$782.99
10	783	01/02/2013	\$2,049.10	\$4,098.20	2	Mountain-200 Black, 42	Black	\$4,098.20
11	800	01/02/2013	\$1,000.44	\$2,000.87	2	Road-550-W Yellow, 44	Yellow	\$2,000.87
12	780	01/02/2013	\$2,071.42	\$2,071.42	1	Mountain-200 Silver, 42	Silver	\$2,071.42
13	767	01/02/2013	\$782.99	\$782.99	1	Road-650 Black, 62	Black	\$782.99
14	766	01/02/2013	\$782.99	\$782.99	1	Road-650 Black, 60	Black	\$782.99
15	759	01/02/2013	\$782.99	\$782.99	1	Road-650 Red, 58	Red	\$782.99

BAN130: Programming for Analytics

4. Data Analysis

- This phase requires you to analyze the SalesAnalysis for Adventure Works and answer the following 5 questions by generating reports using Proc Print for each of the 5 questions:

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

Code:

```
title "Question 1: How many Red color Helmets are sold in 2013 and 2014?";  
proc tabulate data=mylib.SalesAnalysis format=8.;  
class ModifiedDate Color;  
format ModifiedDate YEAR4.;  
var OrderQty;  
tables (ModifiedDate*Color),  
(OrderQty);  
where color='Red' and index(Name,"Helmet");  
keylabel n=' '  
Sum='Number of Red Helmets sold';  
run;
```

Output:

We have used proc tabulate to summarize the total number of red color helmets sold in 2013 and 2014 respectively. We see that the total number of red helmets sold in 2013 is **2940** and it is **1717** in 2014. The sale of red color helmets declined by 41.5% in one year.

BAN130: Programming for Analytics

Question 1: How many Red color Helmets are sold in 2013 and 2014?

		OrderQty
		Number of Red Helmets sold
ModifiedDate	Color	
2013	Red	2940
2014	Red	1717

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

Code:

```
title "Question 2: How many items sold in 2013 and 2014 have a Multi color?";
proc tabulate data=mylib.SalesAnalysis format=8.;
class ModifiedDate Color;
format ModifiedDate YEAR4.;
var OrderQty;
tables (ModifiedDate*Color),
(OrderQty);
where Color='Multi';
keylabel n=' '
Sum='Number of Multi color items sold';
run;
```

Output:

We see that the total number of multi color items sold in 2013 is **10721** and **4288** in 2014. The sale of multi color items declined by 60% in a span of 1 year.

BAN130: Programming for Analytics

Question 2: How many items sold in 2013 and 2014 have a Multi color?

		OrderQty
		Number of Multi color items sold
ModifiedDate	Color	
2013	Multi	10721
2014	Multi	4288

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

Code:

```
title "Question 3: What is the combined Sales total for all the helmets sold in 2013 and 2014";
proc tabulate data=mylib.SalesAnalysis format=dollar14.2;
class ModifiedDate;
var SubTotal;
tables (ModifiedDate ALL),
(SubTotal);
where index(Name,"Helmet");
keylabel n=' '
ALL='Total'
Sum='Sum of sales for all Helmets';
format ModifiedDate YEAR4.;
run;
```

Output:

The sale of helmets in 2013 was **\$227,512.41** and in 2014 it was **\$154,287.93**. The combined sales for these two years was **\$381,800.34**.

BAN130: Programming for Analytics

Question 3: What is the combined Sales total for all the helmets sold in 2013 and 2014

	SubTotal
	Sum of sales for all Helmets
ModifiedDate	
2013	\$227,512.41
2014	\$154,287.93
Total	\$381,800.34

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

Code:

```
title "How many Yellow Color Touring-1000 where sold in 2013 and 2014?";
proc tabulate data=mylib.SalesAnalysis format=8. ;
class ModifiedDate;
var OrderQty;
tables (ModifiedDate ALL),
(OrderQty);
where index(Name,"Touring-1000") and Color='Yellow';
keylabel n=' '
ALL='Total'
Sum='Number of Touring-1000 yellow sold';
format ModifiedDate YEAR4.;
run;
```

Output:

The Number of Touring-1000 of yellow color sold in 2013 is **1953** and that in 2014 is **1215**. The total items sold for these two year comes out to be **3168**. There was an approximately 38% decline in the number of items sold in one year.

BAN130: Programming for Analytics

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

	OrderQty
	Number of Touring-1000 yellow sold
ModifiedDate	
2013	1953
2014	1215
Total	3168

§ What was the total sales in 2013 and 2014?

Code:

```
title "Question 5: What was the total sales in 2013 and 2014?";
proc tabulate data=mylib.SalesAnalysis format=dollar14.2 ;
class ModifiedDate;
var SubTotal;
tables (ModifiedDate ALL),
(SubTotal);
keylabel n=' '
ALL='Total'
Sum='Sum of Total sales';
format ModifiedDate YEAR4.;
run;
```

Output:

We see the total combined sales for 2013 and 2014 was **\$63,680,405.18**. It was **\$43,622,476.81** and **\$20,057,928.37** in 2013 and 2014 respectively. We see the sales declined by 54% in 2014 in comparison to previous year.

BAN130: Programming for Analytics

Question 5: What was the total sales in 2013 and 2014?

	SubTotal
	Sum of Total sales
ModifiedDate	
2013	\$43,622,476.81
2014	\$20,057,928.37
Total	\$63,680,405.18

- Create at least one chart in SAS for any analysis of your choice from SalesAnalysis dataset (this analysis can be of your choice and not necessarily from above 5 questions.)

Code:

```
/* Distribution of OrderQty with respect to color */  
title "Distribution of Order Quantity with respect to color";  
proc tabulate data=mylib.SalesAnalysis format=8.;  
class Color;  
var OrderQty;  
tables (Color),  
(OrderQty);  
keylabel n=' '  
ALL='Total'  
Sum='Distribution of Order Quantity with respect to color';  
run;
```

Output:

We have tabulated the number of items sold for different colors based on our dataset. We see people prefer black color the most among all sorts of products purchased. Next position is occupied by Yellow being second. Although the dataset we used to summarize the distribution of order quantity with respect to color has a lot of missing values which were replaced by NA.

BAN130: Programming for Analytics

Distribution of Order Quantity with respect to color

	OrderQty
	Distribution of Order Quantity with respect to color
Color	
Black	46976
Blue	21780
Multi	15009
NA	44150
Red	9421
Silver	19250
Silver/Black	3931
White	4020
Yellow	28910

Conclusion:

We have completed all the four phases of this report using SAS which includes:

- **Data Import**
- **Data Cleaning**
- **Joining and Merging**
- **Data Analysis**

Also the answers to the Data Analysis Questions were answered to the best of our knowledge.