Seneca

Programming For Analytics

Adventure Works - Territory Sales Project

By Tareq Haboukh

Contents

Phase 1: Data Import

First, I assigned the library "project' to the directory "~" and created a file reference using the filename statement. Then to import the AdventureWorks.xlsx file I used the proc import step for each sheet in the file and created a corresponding dataset.

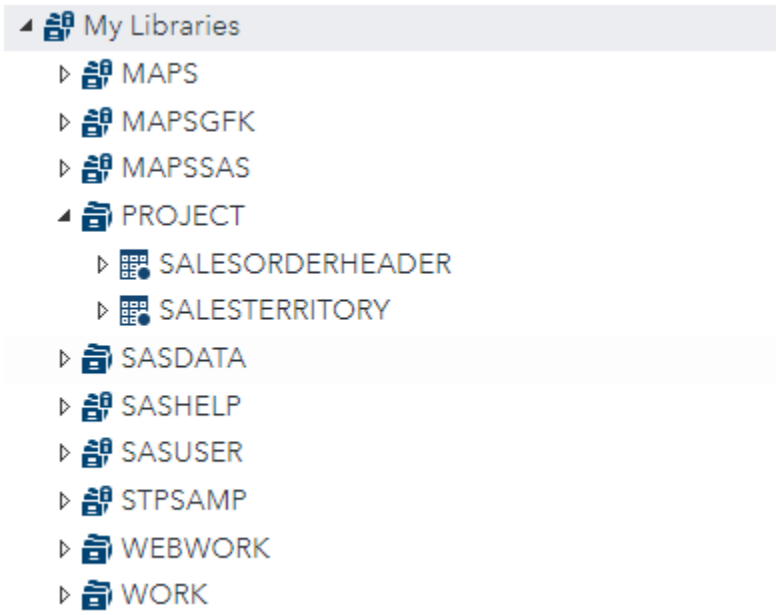
Both Product and SalesOrderDetails sheets were not loaded because they are not being used.

```
* Assign library and reference for the AdventureWorks.xlsx file;
libname project '~';
filename reffile '/home/u59396653/BAN 130/Project/AdventureWorks.xlsx';

* 1. Data Import;

* Import SalesOrderHeader sheet;
proc import
    datafile=reffile
    dbms=xlsx
    out=project.SalesOrderHeader;
    sheet='SalesOrderHeader';
    getnames=yes;
run;

* Import SalesTerritory sheet;
proc import
    datafile=reffile
    dbms=xlsx
    out=project.SalesTerritory;
    sheet='SalesTerritory';
    getnames=yes;
run;
```



Phase 2: Data Cleaning

Creating the SalesOrderHeader_Clean

First, I started with cleaning the SalesOrderHeader dataset and created a new one called SalesOrderHeader_Clean by using a data step.

In this step, only SalesOrderID TotalDue OnlineOrderFlag OrderDate TerritoryID variables were added by using the "keep" statement within the "set" statement, then I created new variables for the ones I wanted to change from character to numeric and used "input" with the format needed and named them by adding Num or Date to the original variable name. Dropped the original variables and then renamed the new ones by removing the added text.

```
* 2. Data Cleaning;

* SalesOrderHeader_Clean;
data project.SalesOrderHeader_Clean;
    set     project.SalesOrderHeader(keep=SalesOrderID TotalDue OnlineOrderFlag OrderDate TerritoryID);
            NumTotalDue=input(TotalDue,dollar16.2);
            NumOnlineOrderFlag=input(OnlineOrderFlag,8.);
            DateOrderDate=input(OrderDate,anydtdte10.);
            NumTerritoryID=input(TerritoryID,8.);
    format  NumTotalDue dollar16.2
            DateOrderDate mmddyy10.;
    drop    TotalDue
            OnlineOrderFlag
            OrderDate
            TerritoryID;
    rename  NumTotalDue=TotalDue
            NumOnlineOrderFlag=OnlineOrderFlag
            DateOrderDate=OrderDate
            NumTerritoryID=TerritoryID;
run;
```

Table: PROJECT.SALESORDERHEADER_CLEAN | View: Column names | Filter: (none)

Columns

- Select all
- SalesOrderID
- TotalDue
- OnlineOrderFlag
- OrderDate
- TerritoryID

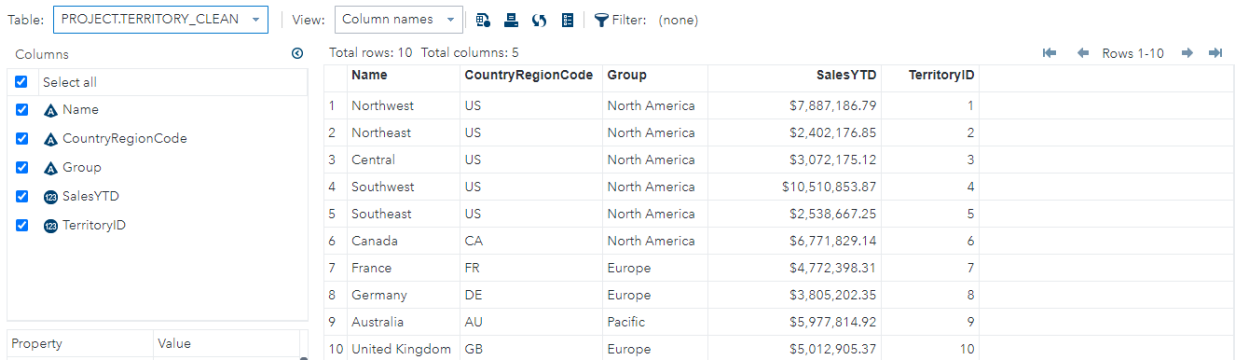
Total rows: 31465 Total columns: 5 Rows 1-100

	SalesOrderID	TotalDue	OnlineOrderFlag	OrderDate	TerritoryID
1	43664	$27,510.41	0	05/31/2011	1
2	43665	$16,158.70	0	05/31/2011	1
3	43669	$807.26	0	05/31/2011	1
4	43671	$9,153.61	0	05/31/2011	1
5	43683	$48,204.07	0	05/31/2011	1
6	43686	$3,899.68	0	05/31/2011	1
7	43699	$3,756.99	-1	05/31/2011	1
8	43711	$3,953.99	-1	06/03/2011	1

Creating the Territory_Clean dataset

The same was done to the Territory dataset and the Territory_Clean dataset was created. Kept only TerritoryID Name CountryRegionCode Group SalesYTD variables and changed SalesYTD and TerritoryID from character to numeric and applied the format needed.

```
* Territory_Clean;
data project.Territory_Clean;
    set     project.SalesTerritory(keep=TerritoryID Name CountryRegionCode Group SalesYTD);
            NumSalesYTD=input(SalesYTD,dollar16.2);
            NumTerritoryID=input(TerritoryID,8.);
    format  NumSalesYTD dollar16.2
            NumTerritoryID 8.;
    drop    SalesYTD TerritoryID;
    rename  NumSalesYTD=SalesYTD NumTerritoryID=TerritoryID;
run;
```



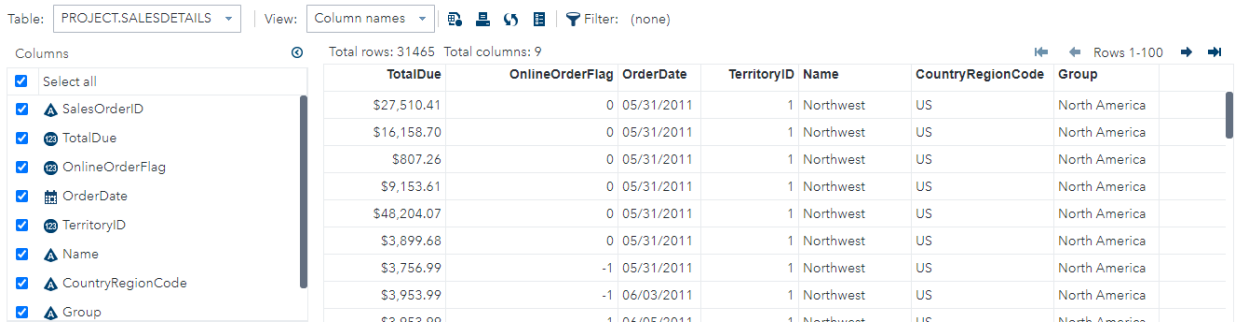
	Name	CountryRegionCode	Group	SalesYTD	TerritoryID
1	Northwest	US	North America	$7,887,186.79	1
2	Northeast	US	North America	$2,402,176.85	2
3	Central	US	North America	$3,072,175.12	3
4	Southwest	US	North America	$10,510,853.87	4
5	Southeast	US	North America	$2,538,667.25	5
6	Canada	CA	North America	$6,771,829.14	6
7	France	FR	Europe	$4,772,398.31	7
8	Germany	DE	Europe	$3,805,202.35	8
9	Australia	AU	Pacific	$5,977,814.92	9
10	United Kingdom	GB	Europe	$5,012,905.37	10

Phase 3: Joining and Merging

Creating the SalesDetails dataset

To join the two new datasets, we first need to sort them by the TerritoryID variable using the proc sort step and then a data merge step that creates a new dataset called SalesDetails by matching records from SalesOrderHeader_Clean with values from Territory_Clean if they have the same TerritoryID in a one-to-many join.

```
* Joining & Merging;
proc sort
    data=project.SalesOrderHeader_Clean;
    by TerritoryID;
run;
proc sort
    data=project.territory_clean;
    by TerritoryID;
run;
data project.SalesDetails;
    merge   project.SalesOrderHeader_Clean(in=Q1)
            project.territory_clean(in=Q2);
    by      TerritoryID;
    if      Q1 = 1 and Q2 =1;
run;
```



TotalDue	OnlineOrderFlag	OrderDate	TerritoryID	Name	CountryRegionCode	Group
$27,510.41	0	05/31/2011	1	Northwest	US	North America
$16,158.70	0	05/31/2011	1	Northwest	US	North America
$807.26	0	05/31/2011	1	Northwest	US	North America
$9,153.61	0	05/31/2011	1	Northwest	US	North America
$48,204.07	0	05/31/2011	1	Northwest	US	North America
$3,899.68	0	05/31/2011	1	Northwest	US	North America
$3,756.99	-1	05/31/2011	1	Northwest	US	North America
$3,953.99	-1	06/03/2011	1	Northwest	US	North America

Creating the SalesAnalysis dataset

To create the SalesAnalysis dataset I used the "Obtaining a total for each by group in SAS" method using the following code, this method consolidates all rows with the same TerritoryID and sums the TotalDue into a new variable SubtTotal.and creates the SalesAnalysis dataset

```
* Sales Analysis;
proc sort
    data=project.SalesDetails
    out=project.SortedSalesDetails;
    by TerritoryID;
run;

data project.SalesAnalysis(drop=TotalDue TerritoryID SalesOrderID OnlineOrderFlag OrderDate);
    retain  Name CountryRegionCode Group SalesYTD SubTotal;
    format  SubTotal dollar16.2;
    set     project.SortedSalesDetails;
    by      TerritoryID;
    if      First.TerritoryID then SubTotal=0;
            SubTotal + TotalDue;
    if      Last.TerritoryID;
run;
```

Table: PROJECT.SALESANALYSIS | View: Column names | Filter: (none)

Columns

- Select all
- Name
- CountryRegionCode
- Group
- SalesYTD
- SubTotal

Property | Value

Total rows: 10 Total columns: 5 Rows 1-10

	Name	CountryRegionCode	Group	SalesYTD	SubTotal
1	Northwest	US	North America	$7,887,186.79	$18,061,660.37
2	Northeast	US	North America	$2,402,176.85	$7,820,209.63
3	Central	US	North America	$3,072,175.12	$8,913,299.25
4	Southwest	US	North America	$10,510,853.87	$27,125,275.34
5	Southeast	US	North America	$2,538,667.25	$8,884,099.37
6	Canada	CA	North America	$6,771,829.14	$18,398,929.19
7	France	FR	Europe	$4,772,398.31	$8,119,749.35
8	Germany	DE	Europe	$3,805,202.35	$5,479,819.58
9	Australia	AU	Pacific	$5,977,814.92	$11,814,376.10
10	United Kingdom	GB	Europe	$5,012,905.37	$8,574,048.71

Phase 4: Data Analysis

Answering the 5 questions

to answer the 5 questions, I used the proc tabulate step, I used a variety of labels and formats statements to modify the result of the procedure

Q1: What is the Total Due for all the North American Regions?

$89,203,473.14

```
proc tabulate
    data=project.SalesAnalysis;
    class Group;
    var Subtotal;
    table Subtotal=' '*f=dollar16.2, Group='All Regions' / rts=25 row=float;
    keylabel sum='Total Due';
run;
proc tabulate
    data=project.SalesAnalysis;
    class Group;
    var Subtotal;
    table Subtotal=' '*f=dollar16.2, Group='Only North America Regions' / rts=25 row=float;
    keylabel sum='Total Due';
    where Group='North America';
run;
```

	All Regions		
	Europe	North America	Pacific
Total Due	$22,173,617.63	$89,203,473.14	$11,814,376.10

	Only North America Regions
	North America
Total Due	$89,203,473.14

Q2: What is the total Sales YTD for U.S.?

$26,411,059.88

```
proc tabulate
    data=project.SalesAnalysis;
    class CountryRegionCode;
    var SalesYTD;
    table SalesYTD=' '*f=dollar16.2, CountryRegionCode='By Country Region Code' / rts=25 row=float;
    keylabel sum='Sales YTD';
run;
proc tabulate
    data=project.SalesAnalysis;
    class CountryRegionCode;
    var SalesYTD;
    table SalesYTD=' '*f=dollar16.2, CountryRegionCode='By Country Region Code' / rts=25 row=float;
    keylabel sum='Sales YTD';
    where CountryRegionCode='US';
run;
```

	By Country Region Code					
	AU	CA	DE	FR	GB	US
Sales YTD	$5,977,814.92	$6,771,829.14	$3,805,202.35	$4,772,398.31	$5,012,905.37	$26,411,059.88

	By Country Region Code
	US
Sales YTD	$26,411,059.88

Q3: How much is due from France and Germany?

$8,119,749.35 and $5,479,819.58

```
proc tabulate
    data=project.SalesAnalysis;
    class Name;
    var Subtotal;
    table Subtotal=' '*f=dollar16.2, Name='Total Amount Due' / rts=25 row=float;
    keylabel sum='Total Due';
    where Name in('France','Germany');
run;
```

	Total Amount Due	
	France	Germany
Total Due	$8,119,749.35	$5,479,819.58

Q4: What is the total Sales YTD for Europe?

$13,590,506.02

```
proc tabulate
    data=project.SalesAnalysis;
    class Group;
    var SalesYTD;
    table SalesYTD=' '*f=dollar16.2, Group='Total Sales YTD' / rts=25 row=float;
    keylabel sum='Total Due';
run;
proc tabulate
    data=project.SalesAnalysis;
    class Group;
    var SalesYTD;
    table SalesYTD=' '*f=dollar16.2, Group='Total Sales YTD In Europe' / rts=25 row=float;
    keylabel sum='Total Due';
    where Group='Europe';
run;
```

	Total Sales YTD		
	Europe	North America	Pacific
Total Due	$13,590,506.02	$33,182,889.02	$5,977,814.92

	Total Sales YTD In Europe
	Europe
Total Due	$13,590,506.02

Q5: How many total territories in U.S?

12,041

In this question, I had to go back to the SalesDetails dataset and then formatted all territories into one value so the tabulate step would count the same value for each country.

```
proc format;
    value Territoryfmt
    low-high='Territories';
run;
proc tabulate
    data=project.SalesDetails;
    format TerritoryID Territoryfmt.;
    class CountryRegionCode TerritoryID;
    table TerritoryID=' ', CountryRegionCode='Country Code' / rts=25 row=float;
    keylabel n=' ';
run;
proc tabulate
    data=project.SalesDetails;
    format TerritoryID Territoryfmt.;
    class CountryRegionCode TerritoryID;
    table TerritoryID=' ', CountryRegionCode='Country Code' / rts=25 row=float;
    keylabel n=' ';
    where CountryRegionCode='US';
run;
```

	Country Code					
	AU	CA	DE	FR	GB	US
Territories	6843	4067	2623	2672	3219	12041

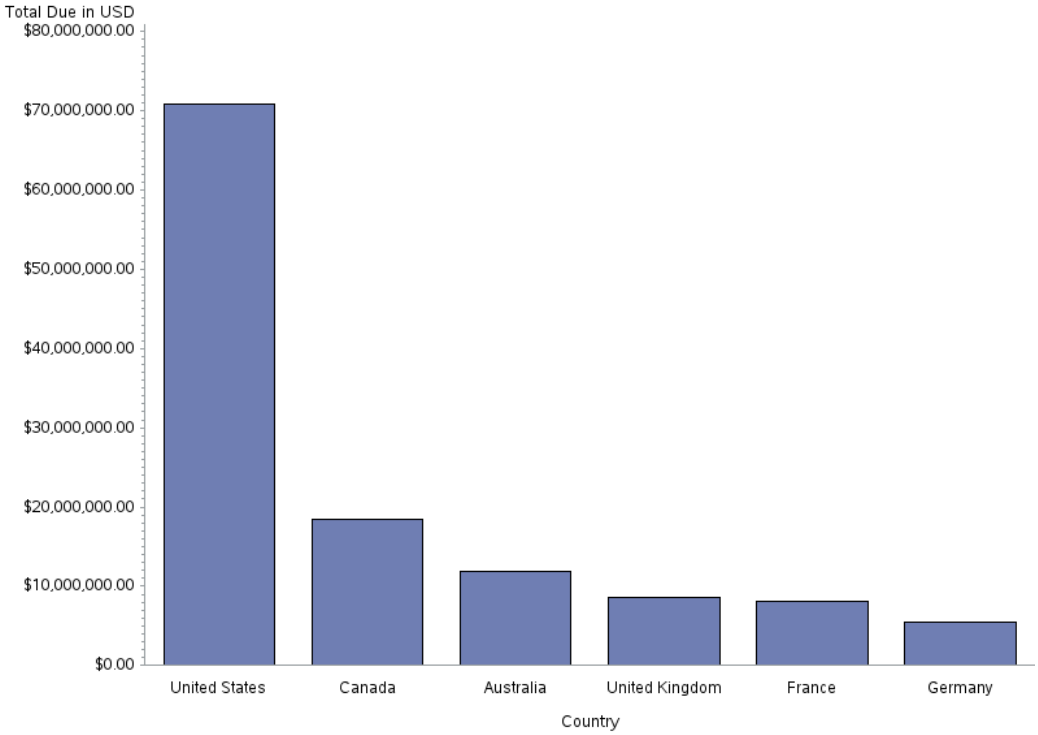
	Country Code
	US
Territories	12041

Create at least one chart

I used the proc gchart to create a bar chart and a pie chart, I wanted to show the total amount due for each country side by side so it was easy to compare but after making the first bar graph and adding all US regions together I felt the need to show the amount due for each region of the US and thus I used the pie chart.

```
proc format;
    value   $CountryRegionName
            'US'='United States'
            'CA'='Canada'
            'AU'='Australia'
            'GB'='United Kingdom'
            'FR'='France'
            'DE'='Germany';
run;
title'Total Due BY Country';
proc gchart
    data=project.SalesDetails;
    format CountryRegionCode $CountryRegionName.;
    label  CountryRegionCode='Country'
           TotalDue="Total Due in USD";
    vbar CountryRegionCode/ sumvar=TotalDue descending noframe;
run;
title'Total Due & Percentages in The US by Region';
proc gchart
    data=project.SalesDetails;
    pie Name/ sumvar=TotalDue descending clockwise angle=90 percent=inside noheading;
    where CountryRegionCode='US';
run;
quit;
```

Total Due BY Country



Total Due & Percentages in The US by Region