SAS

THE POWER TO KNOW®

SAS® Programming 2: Data Manipulation Techniques Case Study

Course Notes

SAS® Programming 2: Data Manipulation Techniques Case Study Course Notes was developed by Peter Styliadis. Additional contributions were made by Brittany Coleman, Mark Jordan, Allison Saito, Prem Shah and Gina Repole. Instructional design, editing, and production support was provided by the Learning Design and Development team.

SAS® Programming 2: Data Manipulation Techniques Case Study Course Notes

Book code E712711, course code PG294CS, prepared date 27Aug2018. PG294CS_001

Table of Contents

To learn more…

For information about other courses in the curriculum, contact the SAS Education Division at 1-800-333-7660, or send e-mail to training@sas.com. You can also find this information on the web at http://support.sas.com/training/ as well as in the Training Course Catalog.

For a list of SAS books (including e-books) that relate to the topics covered in this course notes, visit https://www.sas.com/sas/books.html or call 1-800-727-0025. US customers receive free shipping to US addresses.

Lesson 1 Prepare World Tourism Data

1.1 Case Study Introduction

In this case study, you solve a real-world business problem by applying concepts that you learned in the SAS Programming 2: Data Manipulation Techniques course. Be aware that there are numerous solutions to this problem, and some can include concepts that are outside the scope of this course.

Background Information

In your current role, it is your responsibility to use your SAS programming skills to acquire, organize, and prepare data for your company's business analysts to use. The business analysts will use the data that you provide to create reports, visualizations, and statistical models designed to grow your company's market share.

Business Problem

Your company is interested in analyzing 2014 global tourism data. Your supervisor has sent you two SAS data sets with an email that contains a list of data requirements and an example of the desired outcome. It is your job to follow her requirements and prepare the data for use.

Here are the three tables that you will deliver upon successful completion of the project:

- **cleaned_tourism**
- **final_tourism**
- **nocountryfound**

Specifications for these tables can be found in Section 1.3 of this document.

Data Information

In this case study, two source SAS data sets are provided:

- **tourism**
  contains information about the arrivals of nonresident tourists/visitors, departures, and tourism expenditure in the country and in other countries. The raw data was downloaded from UNdata.
- **country_info**
  contains country names with the associated continent IDs.

Resources

To create the tables for the Tourism case study, please visit the Extended Learning page (ELP).

1. Click the **Data for Tourism Case Study** link.
2. Copy and paste the program in your editor.
3. Run the program to create the **tourism** and **country_info** tables.
4. Confirm that both tables have been created in your **Work** library.

The **Log Scanner** SAS program (bonus) is available on the ELP.

1.2 Data Layout

Here is the column information for the two tables that you will use.

tourism

Column	Type	Description
A	Num	Numeric ID when a country appears in the **Country** column.
Country	Char	This column contains country names, tourism type, tourism category, and the conversion type.
Series	Char	.. = Data not available **Inbound tourism:** TF = Arrivals of non-resident tourists at national borders VF = Arrivals of non-resident visitors at national borders THS = Arrivals of non-resident tourists in hotels and similar establishments TCE = Arrivals of non-resident tourists in all types of accommodation establishments **Outbound tourism:** TF = Departures - trips abroad by resident tourists VF = Departures - trips abroad by resident visitors **Monetary amounts:** IMF = International Monetary Fund CB = Central Bank
_1995 through **_2014**	Char	Scaled numeric data stored as text. The **Country** column contains the information necessary to properly convert this data to a numeric value. Values are dollar amounts (in millions) for rows containing expenditure data or passenger count in thousands for rows containing arrival or departure data. .. = Data not available

country_info

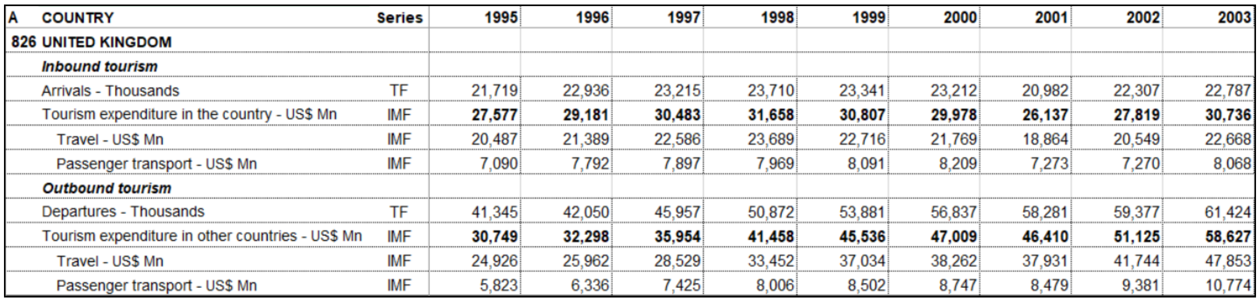
Column	Type	Description
Continent	Num	Continent ID associated with the country. 1 = North America 2 = South America 3 = Europe 4 = Africa 5 = Asia 6 = Oceania 7 = Antarctica
Country	Char	A character country name.

1.3 Requirements

Your supervisor has emailed the requirements and desired outcome for this project. Be sure to read through all the requirements thoroughly.

Raw Data

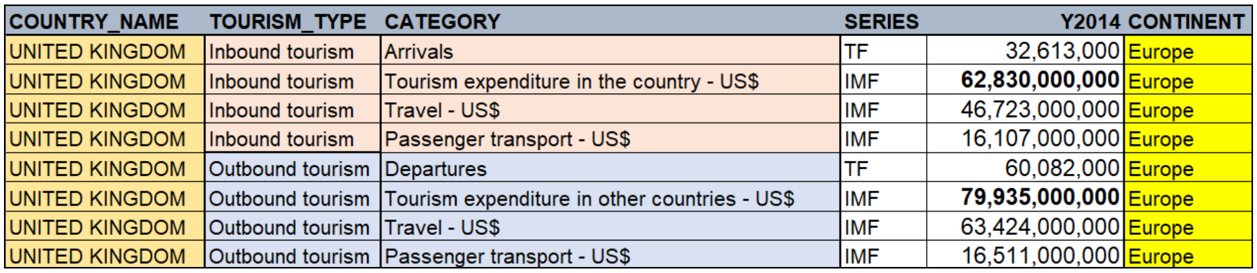
Here is a partial image of the raw data that was imported into SAS for you to use.



A	COUNTRY	Series	1995	1996	1997	1998	1999	2000	2001	2002	2003
826	**UNITED KINGDOM**										
	Inbound tourism										
	Arrivals - Thousands	TF	21,719	22,936	23,215	23,710	23,341	23,212	20,982	22,307	22,787
	Tourism expenditure in the country - US$ Mn	IMF	**27,577**	**29,181**	**30,483**	**31,658**	**30,807**	**29,978**	**26,137**	**27,819**	**30,736**
	Travel - US$ Mn	IMF	20,487	21,389	22,586	23,689	22,716	21,769	18,864	20,549	22,668
	Passenger transport - US$ Mn	IMF	7,090	7,792	7,897	7,969	8,091	8,209	7,273	7,270	8,068
	Outbound tourism										
	Departures - Thousands	TF	41,345	42,050	45,957	50,872	53,881	56,837	58,281	59,377	61,424
	Tourism expenditure in other countries - US$ Mn	IMF	**30,749**	**32,298**	**35,954**	**41,458**	**45,536**	**47,009**	**46,410**	**51,125**	**58,627**
	Travel - US$ Mn	IMF	24,926	25,962	28,529	33,452	37,034	38,262	37,931	41,744	47,853
	Passenger transport - US$ Mn	IMF	5,823	6,336	7,425	8,006	8,502	8,747	8,479	9,381	10,774

Desired Outcome

The final table should be a combination of the **cleaned_tourism** table and the **country_info** table. See my example below:



COUNTRY_NAME	TOURISM_TYPE	CATEGORY	SERIES	Y2014	CONTINENT
UNITED KINGDOM	Inbound tourism	Arrivals	TF	32,613,000	Europe
UNITED KINGDOM	Inbound tourism	Tourism expenditure in the country - US$	IMF	**62,830,000,000**	Europe
UNITED KINGDOM	Inbound tourism	Travel - US$	IMF	46,723,000,000	Europe
UNITED KINGDOM	Inbound tourism	Passenger transport - US$	IMF	16,107,000,000	Europe
UNITED KINGDOM	Outbound tourism	Departures	TF	60,082,000	Europe
UNITED KINGDOM	Outbound tourism	Tourism expenditure in other countries - US$	IMF	**79,935,000,000**	Europe
UNITED KINGDOM	Outbound tourism	Travel - US$	IMF	63,424,000,000	Europe
UNITED KINGDOM	Outbound tourism	Passenger transport - US$	IMF	16,511,000,000	Europe

Data Requirements

I have broken up the requirements into two parts for you.

Here is what I am looking for when you create the **cleaned_tourism** table:

- **Tourism_Type** – Create a new column that contains the type of tourism by reorganizing the original **Country** column. Valid values are *Inbound tourism* or *Outbound tourism*.
- **Category** – Create a new column that contains the category names in the data by extracting and modifying values from the original **Country** column. Original categories values need to be updated to the new values shown below. In total, there should be six distinct values.

Original Category Distinct Values	New Distinct Values
Arrivals - Thousands	Arrivals
Departures - Thousands	Departures
Passenger transport - US$ Mn	Passenger transport - US$

Original Category Distinct Values	New Distinct Values
Tourism expenditure in other countries - US$ Mn	Tourism expenditure in other countries - US$
Tourism expenditure in the country - US$ Mn	Tourism expenditure in the country - US$
Travel - US$ Mn	Travel - US$

- **Series** – Convert all values to uppercase and change data that is not available ("..") to a missing character value.
- **Y2014** – Create a new column that changes the scaled character values in the original **_2014** year column to the full numeric value. To create the numeric values, determine the conversion type and multiply with the scaled original values. The row category determines whether the value for **Y2014** should be in the millions (abbreviated Mn) or thousands. The new values should be formatted with the COMMA format.
  - For example, if the category is Travel - US$ MN and the value for **_2014** is 4.26, **Y2014** is equal to 4.26 * 1000000, or 4,260,000.
- Include only **Country_Name**, **Tourism_Type**, **Category**, **Series**, and **Y2014** in the output table.
- ***Bonus task***: After creating the **cleaned_tourism** table, run the **Log Checker.sas** program on your log to ensure that your code meets our regulatory requirements.

Lastly, I need you to merge the **cleaned_tourism** table with the **country_info** table and do the following:

- Create two new tables:
  - One named **final_tourism** that contains ***only*** merged data.
  - A second table that contains a list of distinct countries from the **cleaned_tourism** table that do not have a match in the **country_info** table. Name this table **nocountryfound**.
- Create a format for the **Continent** column that assigns continent IDs to the corresponding continent names. Permanently apply the format in the **final_tourism** table.
  - 1 = North America
  - 2 = South America
  - 3 = Europe
  - 4 = Africa
  - 5 = Asia
  - 6 = Oceania
  - 7 = Antarctica

1.4 Assignment Guide

Below is a suggested guide to help you solve the business problem. Be aware that there are multiple solutions to this problem and that you do not need to follow the steps below.

For this case study, the steps will be broken down into two sections for you to follow. The first section is preparing and cleaning the **tourism** data, and the second section is merging **cleaned_tourism** table with the **country_info** table. Each section contains steps to help guide you through the problem. If you are stuck, you can refer to the **Hints** section in the document or post a question in the discussion forums.

Prepare Tourism Data

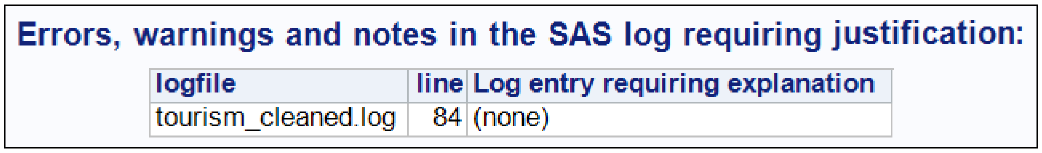
1. Remove years **_1994** through **_2013**.
2. Create the **Country_Name** and **Tourism_Type** columns.
3. Convert values to uppercase for **Series** and also convert its missing values.
4. Determine the conversion type.
5. Change the data not available in **_2014** to a single ".".
6. Create the **Y2014** column by explicitly converting character values in **_2014** and multiplying by the conversion type.
7. Create the new **Category** column and change the original values to the required values.
8. Permanently format **Y2014**.
9. Remove unnecessary variables.

Merge Data

1. Create the format for the **Continent** column.
2. Merge the **cleaned_tourism** table with **country_info**, create the **final_tourism** and **nocountryfound** tables, and permanently apply your format.

Bonus

In regulated industries, there are frequently automated processes that scan logs for “forbidden” messages and create reports of items requiring justification to the regulators. Rerun the program that you wrote to prepare the **tourism** table and save the log. Open the **Log Scanner** SAS program and run the log-checking program to test your code. Your results should look like this:



Errors, warnings and notes in the SAS log requiring justification:

logfile	line	Log entry requiring explanation
tourism_cleaned.log	84	(none)

Note: Depending on the length of your code, the value in the **line** column might differ.

1.5 Data Validation

To validate your results, run SAS procedures to compare your results with the results below.

1. Use the PRINT procedure to view the list of countries from the **cleaned_tourism** table that did not have a match in the **country_info** table.

Country_Name
BONAIRE
CABO VERDE
CONGO, DEMOCRATIC REPUBLIC OF
CURAÇAO
HONG KONG, CHINA
MACAO, CHINA
SABA
SERBIA AND MONTENEGRO
SINT EUSTATIUS

2. Use the FREQ procedure to view the distinct levels and frequency counts of the variables **Category**, **Series**, **Tourism_Type**, and **Continent** in the **final_tourism** table. Suppress the percent and cumulative frequencies.

Number of Variable Levels

Variable	Label	Levels	Missing Levels	Nonmissing Levels
Category		6	0	6
Series	Series	8	1	7
Tourism_Type		2	0	2
Continent	Numeric number for continent	6	0	6

Category	Frequency
Arrivals	249
Departures	221
Passenger transport - US$	418
Tourism expenditure in other countries - US$	209
Tourism expenditure in the country - US$	209
Travel - US$	418

Series

Series	Frequency
CB	12
COUNTRY	29
IMF	1016
TCE	23
TF	291
THS	19
VF	49

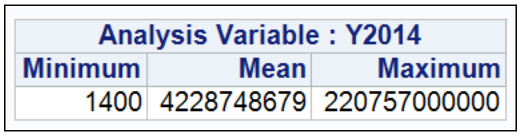
Frequency Missing = 285

Tourism_Type	Frequency
Inbound tourism	876
Outbound tourism	848

Numeric number for continent

Continent	Frequency
North America	284
South America	105
Europe	369
Africa	422
Asia	382
Oceania	162

3. Use the MEANS procedure to view the mean, minimum, and maximum of every year in the **final_tourism** table. Specify zero as the maximum number of decimal places.



Analysis Variable : Y2014		
Minimum	Mean	Maximum
1400	4228748679	220757000000

1.6 Hints

The following hints will help you complete the case study. You can also use SAS Documentation for additional information.

Prepare Tourism Data

1. Remove years **_1994** through **_2013**.

   Consider using the DROP statement or the DROP= data set option.

2. Create the **Country_Name** and **Tourism_Type** columns.

```
length Country_Name $300 Tourism_Type $20;
retain Country_Name "" Tourism_Type "";
if A ne . then Country_Name=Country;
if lowcase(Country)="inbound tourism" then
   Tourism_Type="Inbound tourism";
else if lowcase(Country)='outbound tourism' then
   Tourism_Type="Outbound tourism";
if Country_Name ne Country and Country ne Tourism_Type;
```

3. Convert values to uppercase for **Series** and also convert its missing values.

```
Series=upcase(series);
if Series=".." then Series="";
```

4. Determine the conversion type.

```
ConversionType=strip(scan(country,-1,' '));
```

5. Change the data not available in **_2014** to a single ".".

```
if _2014='..' then _2014='.';
```

6. Create the **Y2014** column by explicitly converting character values in **_2014** and multiplying by the conversion type.

```
if ConversionType = 'Mn' then
   do;
      if input(_2014,16.) ne . then
       Y2014=input(_2014,16.)*1000000;
      else Y2014=.;
   end;
else if ConversionType = 'Thousands' then
   do;
      if input(_2014,16.) ne . then
       Y2014=input(_2014,16.)*1000;
      else Y2014=.;
   end;
```

7. Create the new **Category** column by changing the original values to the required values.

```
if ConversionType = 'Mn' then
   Category=cat(scan(country,1,'-','r')," - US$");
else if ConversionType = 'Thousands' then
   Category=scan(country,1,'-','r');
```

8. Permanently format **Y2014**.

```
format Y2014 comma25.;
```

9. Remove unnecessary variables.

```
drop A ConversionType Country _2014;
```

Merge Data

1. Create the format for the **Continent** column.

```
proc format;
   value continents
      1 = "North America"
      2 = "South America"
      3 = "Europe"
      4 = "Africa"
      5 = "Asia"
      6 = "Oceania"
      7 = "Antarctica";
run;
```

2. Merge the **cleaned_tourism** table with **country_info**, create the **final_tourism** and **nocountryfound** tables, and permanently apply your format.

```
proc sort data=libref.country_info(rename=(Country=Country_Name))
          out=Country_Sorted;
    by Country_Name;
run;

data    Final_Tourism NoCountryFound(keep=Country_Name);
    merge cleaned_tourism(in=t)
          country_sorted(in=c);
    by country_name;
    if t=1 and c=1 then output Final_Tourism;
    if (t=1 and c=0) and first.country_name then
       output NoCountryFound;
    format Continent continents.;
run;
```

Bonus

In regulated industries, there are frequently automated processes that scan logs for “forbidden” messages and create reports of items requiring justification to the regulators. Rerun the program that you wrote to prepare the tourism table and save the log. Open the Log Scanner SAS program and run the log-checking program to test your code.

Open the **Log Scanner** SAS program and change the macro variables that specify the location of your log and your log’s file name. Here is an example:

```
%let logpath = s:\saswork\logs;
%let filename = test.log;
```