SAS® Programming 1: Essentials Case Study

Course Notes

SAS® Programming 1: Essentials Case Study Course Notes was developed by Peter Styliadis. Additional contributions were made by Brittany Coleman, Mark Jordan, Gina Repole and Kristin Snyder. Instructional design, editing, and production support was provided by the Learning Design and Development team.

SAS® Programming 1: Essentials Case Study Course Notes

Book code E712361, course code PG194CS, prepared date 23Aug2018. PG194CS_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 United States Airport Claims

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 1: Essentials course. Be aware that there are numerous solutions to this problem, and some can include concepts that are outside the scope of the SAS Programming 1 course.

Background Information

You are a SAS programmer with six months of experience who is in charge of creating basic reports and maintaining SAS programs. Recently, you completed a SAS Programming course, and your supervisor gives you your first SAS programming project.

Business Problem

Your first project is to prepare and analyze Transportation Security Administration (TSA) Airport Claims data from 2002 through 2017. The TSA is an agency of the United States Department of Homeland Security that has authority over the security of the traveling public. A claim is filed if you are injured or your property is lost or damaged during the screening process at an airport.

To complete your project, you follow your supervisor's requirements, which are in Section 1.3 of this document. Here is what you need to do:

- Prepare the data.
- Create one PDF report that analyzes the overall data as well as the data for a dynamically specified state.

Data Information

The data that you use is **TSAClaims2002_2017.csv**, which was created from the following:

- TSA Airport Claims data from https://www.dhs.gov/tsa-claims-data.
- FAA Airport Facilities data from https://www.faa.gov/airports/airport_safety/airportdata_5010/.

The **TSAClaims2002_2017.csv** file was created by concatenating each individual TSA Airport Claims table. After the concatenation, the data was joined with the FAA Airport Facilities data. Here are a few notes regarding the data:

- All data is public data, and accuracy is not guaranteed.
- The column **Airport_Codes** from the TSA Airport Claims data has been joined with **Location_ID** from the FAA Airports Facilities data. Some **Airport_Codes** values do not correspond to **Location_ID** values.
- Columns in the TSA Airport Claims data have changed over the years. Because of this, some of the original columns were removed from the data for this case study.
- The column **Item_Category** does not have consistent input values over the years. For this reason, you do not clean this column in this case study.

Resources

To attempt this case study, you need to download the **TSAClaims2002_2017.csv** file.

1.2 Data Layout

Here is the column information for the **TSAClaims2002_2017.csv** table.

Column	Description
Claim_Number	Number for each claim. Some claims can have duplicate claim numbers but different information for each claim. Those claims are considered valid for this case study. Any duplicate rows should be removed from the data.
Date_Received	Date the claim was received. **Date_Received** must occur after **Incident_Date**. Range: From 2002 through 2017
Incident_Date	Date the incident occurred. **Incident_Date** must occur before **Date_Received**. Range: From 2002 through 2017
Airport_Code	Airport code three-letter abbreviation.
Airport_Name	Full name of the airport.
Claim_Type	Category of the claim. If the claim is separated into two types by a slash, **Claim_Type** is the first type. For example: Personal Property Loss/Injury is considered Personal Property Loss. Possible values (14): • Bus Terminal • Complaint • Compliment • Employee Loss (MPCECA) • Missed Flight • Motor Vehicle • Not Provided • Passenger Property Loss • Passenger Theft • Personal Injury • Property Damage • Property Loss • Unknown • Wrongful Death

Column	Description
Claim_Site	Airport location of the claim. Possible values (8): • Bus Station • Checked Baggage • Checkpoint • Motor Vehicle • Not Provided • Other • Pre-Check • Unknown
Item_Category	Type of items that have been filed in the claim. Depending on the year of the data, the **Item_Category** values are input differently. Because of varying consistency, you do not clean this column for the case study.
Close_Amount	The dollar amount a claim was closed for.
Disposition	The final settlement of the claim. Possible values (10): • *Insufficient • Approve in Full • Closed:Canceled • Closed:Contractor Claim • Deny • In Review • Pending Payment • Received • Settle • Unknown *Insufficient is the value from the raw data.
StateName	Associated airport state name (for example, NEW YORK). Requirements: Values should be in all proper case. (Original data is in all uppercase.)
State	Associated airport state code. This is the standard two-letter abbreviation used by the post office for US states and territories (for example, IL, PR, CQ). Requirements: Values should be in all uppercase.
County	Airport associated county (or parish) name (for example, Cook).
City	Associated airport city name (for example, CHICAGO).

1.3 Requirements

To prepare and analyze the data, you follow the requirements given to you by your supervisor. Be aware that these requirements are only assumed for this case study. They are not an accurate representation of TSA requirements.

Data Requirements

- Import the raw data file **TSAClaims2002_2017.csv**.
- The final data should be in the permanent library **tsa**, and the data set should be named **claims_cleaned**.
- Entirely duplicated records need to be removed from the data set.
- All missing and "-" values in the columns **Claim_Type**, **Claim_Site**, and **Disposition** must be changed to *Unknown*.
- Values in the columns **Claim_Type**, **Claim_Site**, and **Disposition** must follow the requirements in the data layout.
- All **StateName** values should be in proper case.
- All **State** values should be in uppercase.
- You create a new column named **Date_Issues** with a value of *Needs Review* to indicate that a row has a date issue. Date issues consist of the following:
  - a missing value for **Incident_Date** or **Date_Received**
  - an **Incident_Date** or **Date_Received** value out of the predefined year range of 2002 through 2017
  - an **Incident_Date** value that occurs after the **Date_Received** value
- Remove the **County** and **City** columns.
- Currency should be permanently formatted with a dollar sign and include two decimal points.
- All dates should be permanently formatted in the style 01JAN2000.
- Permanent labels should be assigned columns by replacing the underscores with a space.
- Final data should be sorted in ascending order by **Incident_Date**.

Report Requirements

The final single PDF report needs to ***exclude all rows with date issues*** in the analysis and answer the following questions:

1. How many date issues are in the overall data?
2. How many claims per year of **Incident_Date** are in the overall data? Be sure to include a plot.
3. Lastly, a user should be able to dynamically input a specific state value and answer the following:
   a. What are the frequency values for **Claim_Type** for the selected state?
   b. What are the frequency values for **Claim_Site** for the selected state?
   c. What are the frequency values for **Disposition** for the selected state?
   d. What is the mean, minimum, maximum, and sum of **Close_Amount** for the selected state? Round to the nearest integer.

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.

You can follow the SAS programming process steps below to solve the business problem. How you solve each requirement is your choice, but if you are stuck, you can refer to the **Hints** section in this document or post a question in the discussion forums.

Access Data

1. Import the **TSAClaims2002_2017.csv** file.

Explore Data

1. Preview the data.
2. Explore the following columns and make note of any adjustments needed using the information from the **Data Layout** and **Requirements** sections above.
   a. **Claim_Site**
   b. **Disposition**
   c. **Claim_Type**
   d. **Date_Received**
   e. **Incident_Date**

Prepare Data

1. Remove duplicate rows.
2. Sort the data by ascending **Incident_Date**.
3. Clean the **Claim_Site** column.
4. Clean the **Disposition** column.
5. Clean the **Claim_Type** column.
6. Convert all **State** values to uppercase and all **StateName** values to proper case.
7. Create a new column to indicate date issues.
8. Add permanent labels and formats.
9. Drop **County** and **City**.

Analyze Data

1. Analyze the overall data to answer the business questions. Be sure to add appropriate titles.
2. Analyze the state-level data to answer the business questions. Be sure to add appropriate titles.

Export Data

1. Export the end results into a single PDF named **ClaimReports** that has a style of your choice.

2. Customize the procedure labels in your report.

1.5 Results

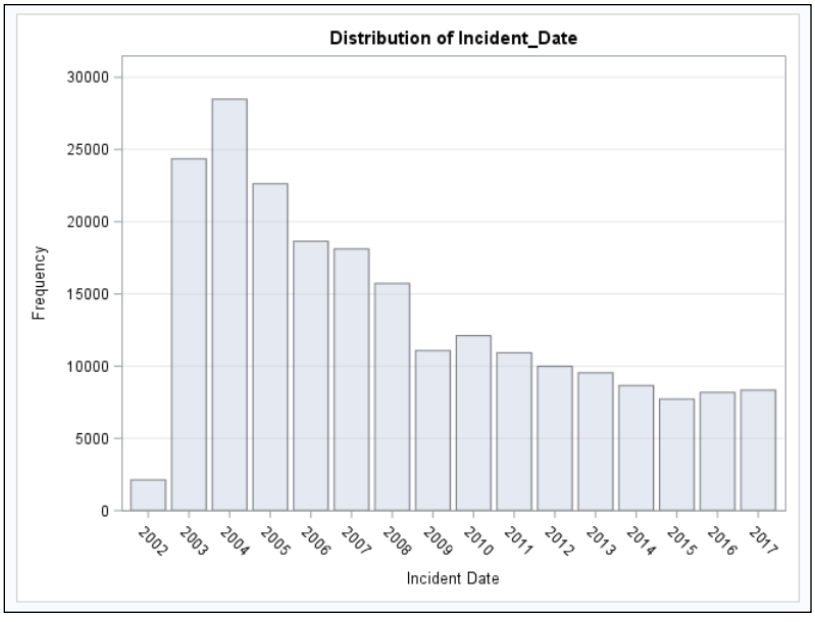
Here are the results for the overall analysis and a report with the selected state of California.

1. How many date issues are in the overall data?

Date Issues	
Date_Issues	Frequency
Needs Review	4241
Frequency Missing = 216609	

2. How many claims per year of **Incident_Date** are in the overall data? Be sure to include a plot.

Incident Date	
Incident_Date	Frequency
2002	2123
2003	24359
2004	28484
2005	22631
2006	18643
2007	18116
2008	15727
2009	11075
2010	12108
2011	10921
2012	9984
2013	9536
2014	8659
2015	7721
2016	8182
2017	8340



3. Dynamically input a specific state value and answer the following:
   a. What are the frequency values for **Claim_Type** for the selected state?

Claim Type	
Claim_Type	Frequency
Passenger Property Loss	14892
Property Damage	8996
Unknown	756
Personal Injury	194
Passenger Theft	55
Employee Loss (MPCECA)	51
Motor Vehicle	17
Complaint	7
Compliment	2
Missed Flight	1
Property Loss	1

   b. What are the frequency values for **Claim_Site** for the selected state?

Claim Site	
Claim_Site	Frequency
Checked Baggage	19553
Checkpoint	5142
Other	187
Unknown	68
Motor Vehicle	19
Pre-Check	2
Bus Station	1

   c. What are the frequency values for **Disposition** for the selected state?

Disposition	Frequency
Deny	10918
Approve in Full	5266
Settle	4192
Unknown	3188
In Review	1086
*Insufficient	187
Closed:Contractor Claim	83
Closed:Canceled	49
Received	3

   d. What is the mean, minimum, maximum, and sum of **Close_Amount** for the selected state? Round to the nearest integer.

Analysis Variable : Close_Amount Close Amount			
Mean	Minimum	Maximum	Sum
98	0	14519	2096386

1.6 Hints

The following hints will help you in completing the case study. You can also use SAS Documentation for additional information or post a question in the discussion forums.

Access Data

1. Import the **TSAClaims2002_2017.csv** file.

```
%let path=<Enter file location of the input data>;
libname tsa "<Enter the file location of the output data>";

options validvarname=v7;
proc import datafile="&path/TSAClaims2002_2017.csv"
     dbms=csv
     out=tsa.ClaimsImport
     replace;
    guessingrows=max;
run;
```

Partial Display of the Log

```
NOTE: TSA.CLAIMSIMPORT data set was successfully created.
NOTE: The data set TSA.CLAIMSIMPORT has 220855 observations and 14 variables.
NOTE: PROCEDURE IMPORT used (Total process time):
```

Explore Data

1. Preview the data.

```
proc print data=tsa.ClaimsImport (obs=20);
run;
proc contents data=tsa.ClaimsImport varnum;
run;
```

2. Explore the following columns and make note of any adjustments needed using the information from the **Data Layout** and **Requirements** sections above.

```
proc freq data=tsa.Claims_NoDups;
    tables Claim_Site Disposition Claim_Type / nocum nopercent;
    tables Date_Received Incident_Date / nocum nopercent;
    format Date_Received Incident_Date year4.;
run;
```

   a. **Claim_Site**

      1) Change missing values and "-" to *Unknown*.

   b. **Claim_Type**

      1) Change missing values and "-" to *Unknown*.
      2) Change *Passenger Property Loss/Injury* values to *Passenger Property Loss*.
      3) Change *Property Damage/Personal Injury* values to *Property Damage*.

c. **Disposition**

1) Change missing values and "-" to *Unknown*.

2) Remove the leading space in *Closed: Canceled*.

3) Fix the missing character *C* and the leading space in *losed: Contractor Claim*.

d. **Date_Received**

1) Notice that many dates are after the year 2017 or missing.

e. **Incident_Date**

1) Notice that many dates are before 2002, after 2017, missing, or after **Date_Received**.

Prepare Data

1. Remove duplicate rows.

```
proc sort data=tsa.ClaimsImport
     out=tsa.Claims_NoDups
     nodupkey;
    by _all_;
run;
```

Partial Display of the Log

```
NOTE: There were 220855 observations read from the data set TSA.CLAIMSIMPORT.
NOTE: 5 observations with duplicate key values were deleted.
NOTE: The data set TSA.CLAIMS_NODUPS has 220850 observations and 14 variables.
NOTE: PROCEDURE SORT used (Total process time):
      real time           0.46 seconds
      cpu time            0.73 seconds
```

2. Sort the data by ascending **Incident_Date**.

```
proc sort data=tsa.Claims_NoDups;
    by Incident_Date;
run;
```

3. Clean the **Claim_Site** column.

```
if Claim_Site in ('-',"") then Claim_Site="Unknown";
```

4. Clean the **Disposition** column.

```
if Disposition in ("-","") then
   Disposition='Unknown';
else if Disposition='Closed: Canceled' then
     Disposition='Closed:Canceled';
else if Disposition='losed: Contractor Claim' then
     Disposition='Closed:Contractor Claim';
```

5. Clean the **Claim_Type** column.

```
if Claim_Type in ("-","") then Claim_Type="Unknown";
else if Claim_Type = 'Passenger Property Loss/Personal Injur' then
     Claim_Type='Passenger Property Loss';
else if Claim_Type = 'Passenger Property Loss/Personal Injury' then
     Claim_Type='Passenger Property Loss';
else if Claim_Type = 'Property Damage/Personal Injury' then
     Claim_Type='Property Damage';
```

6. Convert all **State** values to uppercase and all **StateName** values to proper case.

```
State=upcase(state);
StateName=propcase(StateName);
```

7. Create a new column that indicates date issues.

```
if (Incident_Date > Date_Received or
   Incident_Date = . or
   Date_Received = . or
   year(Incident_Date) < 2002 or
   year(Incident_Date) > 2017 or
   year(Date_Received) < 2002 or
   year(Date_Received) > 2017) then Date_Issues="Needs Review";
```

8. Add permanent labels and formats.

```
format Incident_Date Date_Received date9. Close_Amount Dollar20.2;
label Airport_Code="Airport Code"
      Airport_Name="Airport Name"
      Claim_Number="Claim Number"
      Claim_Site="Claim Site"
      Claim_Type="Claim Type"
      Close_Amount="Close Amount"
      Date_Issues="Date Issues"
      Date_Received="Date Received"
      Incident_Date="Incident Date"
      Item_Category="Item Category";
```

9. Drop **County** and **City**.

```
drop County City;
```

Analyze

1. Analyze the overall data to answer the business questions. Be sure to add appropriate titles.

```
title "Overall Date Issues in the Data";
proc freq data=TSA.Claims_Cleaned;
    table Date_Issues / nocum nopercent;
run;
title;

ods graphics on;
title "Overall Claims by Year";
proc freq data=TSA.Claims_Cleaned;
    table Incident_Date / nocum nopercent plots=freqplot;
    format Incident_Date year4.;
    where Date_Issues is null;
run;
title;
```

2. Analyze the state-level data to answer the business questions. Be sure to add appropriate titles.

```
%let StateName=California;

title "&StateName Claim Types, Claim Sites and Disposition
     Frequencies";
proc freq data=TSA.Claims_Cleaned order=freq;
    table Claim_Type Claim_Site Disposition / nocum nopercent;
    where StateName="&StateName" and Date_Issues is null;
run;

title "Close_Amount Statistics for &StateName";
proc means data=TSA.Claims_Cleaned mean min max sum maxdec=0;
    var Close_Amount;
    where StateName="&StateName" and Date_Issues is null;
run;
title;
```

Export

1. Export the end results into a single PDF named **ClaimReports** that has a style of your choice.

```
%let outpath=<Enter file location of your output data>;
ods pdf file="&outpath\ClaimsReports.pdf" style=Meadow;
```

2. Customize the procedure labels in your report.

```
ods proclabel "Enter new procedure title";
```