

## Paper 076-2012

## Using SAS® and Zip Codes to Create a Nationwide First Responders Directory

Andy Hummel, Delta Air Lines, Atlanta, GA

### ABSTRACT

SAS can calculate the distance in miles between two points by using ZIP codes and longitudinal and latitudinal data. This can be used to match potential disaster areas with the location of first responders based on their distance from the potential crisis area. Delta Air Lines serves more than 250 airports in the United States and has more than 10,000 pilots residing in all 50 states who could act as first responders. Using each airport's longitudinal and latitudinal coordinates along with each pilot's home ZIP code, a first responder's directory can be created listing each pilot living within a specified distance from each airport. This directory can be used to quickly match first responders to any emergency location in the United States.

### INTRODUCTION

Emergency situations frequently occur during inconvenient hours and in remote locations. Corporate emergency response teams are often housed in centralized locations and can be hindered when responding to rural areas or during non-business hours. In such examples the ability to quickly deploy first responders that are already positioned close to the crisis can be critical to mitigating the impact of the emergency.

### LONGITUDINAL AND LATITUDINAL COORDINATES FOR THE FIRST RESPONDERS

To calculate the distance that each first responder (Delta pilot) lives from an airport a zip code or longitudinal and latitudinal coordinate must be found for each pilot and each airport. At Delta each pilot's home address with zip code is stored in a Teradata data warehouse and is easily retrievable.

The following code retrieves the pilot's employee number, zip code, primary phone number and first and last name. By using a WHERE statement of IS NOT NULL only pilots with zip code data in the data base will be included.

```
PROC SQL;
CONNECT to teradata (database=secmv tdpid=dwprod user=XXX password=XXX mode=teradata);
CREATE TABLE per_1 AS
SELECT * FROM connection to teradata
(SELECT DISTINCT
  substr(empl_nbr,2,6) AS empl_nbr,
  substr(postl_area_cd,1,5) AS tera_zip,
  primary_ph,
  plt_frst_nm AS Plt_First_Name,
  plt_lst_nm AS Plt_Last_Name
FROM SECMV.PLT_PRSNL AS a
WHERE mail_postl_area_cd IS NOT NULL
ORDER by empl_nbr
);DISCONNECT FROM teradata;
QUIT;
```

Once the zip codes are found for the pilots a corresponding longitudinal and latitudinal coordinate needs to be determined for each zip code. SAS provides a data set named SASUSER.ZIPCODE that contains a list of United States zip codes and centroid x y coordinates for each code. The data set is updated quarterly and can be downloaded from the support.sas.com website.

The following code opens the SASUSER.ZIPCODE data set and renames the variables.

```
libname out '/home/ahummel/zip_code/output';

DATA all_zip_1 (keep=plt_long plt_lat city2 match_key all_zip_code);
  SET out.zip_code_source ;
  RENAME x=plt_long y=plt_lat city=city2 zip=all_zip_code;
RUN;
```

The pilot zip code data needs to be merged to the data set that contains all zip codes so that the pilot zip codes are matched to the corresponding longitudinal and latitudinal coordinates. An INNER JOIN is used to keep records where there is a longitudinal and latitudinal coordinate for the pilot. This ensures that incomplete data is not populated in the directory. A variable named "merge\_on" is created for merging the pilot data to the station data later in the program.

```
PROC SQL;
CREATE TABLE per_and_zip_1 AS
SELECT DISTINCT
    a.*, b.*,
    1 as merge_on
FROM per_1 as a
INNER JOIN all_zip_1 AS b
ON a.plt_zip=b.all_zip_code
ORDER BY empl_nbr
;QUIT;
```

## LONGITUDINAL AND LATITUDINAL COORDINATES FOR DELTA STATIONS

The longitudinal and latitudinal data for all Delta stations needs to be determined. Delta has a data set stored in Teradata that contains each station's zip code and corresponding longitudinal and latitudinal coordinates.

The following code retrieves the station code and coordinates. Again, a variable named "merge\_on" is created for merging the pilot data to the station data later in the program.

```
PROC SQL;
CONNECT TO teradata (database=secmv tdpid=dwprod user=XXX password=XXX mode=teradata);
CREATE TABLE station_1 AS
SELECT DISTINCT * FROM connection to teradata
(SELECT DISTINCT
    arpt_cd,
    station_lat,
    station_long,
    1 as merge_on
FROM STDMV.STATION AS a
);DISCONNECT FROM teradata;
quit;
```

## CALCULATING THE DISTANCE BETWEEN TWO COORDINATES

The following steps will build the code needed to create the emergency response directory step-by-step with the complete PROC SQL code shown at the end.

The first step in calculating the distance is to merge the pilot data with the station data. This is done with the JOIN statement in the below code. During the earlier steps of creating the pilot data set and the station data set a variable named "merge\_on" was created and given a value of 1 in each data set.

```
FROM per_and_zip_1 AS a
JOIN station_1 AS b
ON a.merge_on=b.merge_on
```

Since the "merge\_on" variable is 1 in both data sets each pilot will be merged with each Delta station which creates a data set with over 2 million rows (10,000 pilots each matched to 250 Delta stations). In order to create a data set that is useful and manageable a value of 50 miles or less is required for the record to be kept in the output data set. This will be achieved by using a HAVING statement in the below code. The HAVING statement evaluates the "miles\_away" variable and only outputs records where "miles\_away" is less than or equal to 50. The HAVING statement is very valuable in that it allows evaluation go be conducted after a calculation has taken place.

```
HAVING miles_away <= 50
```

To calculate the distance between a pilot's home and a Delta station the longitude and latitude for each is plugged into a formula which computes the distance in miles between the two centroids. Four values are needed to perform the calculation. The values are the longitude and latitude of the pilot's home address ('plt\_long', 'plt\_lat') and the longitude and latitude of the Delta station ('station\_long', 'station\_lat'). The formula below does the calculation and is rather ugly, the SAS user does not have to understand the math behind the calculation, only where to correctly input the two longitude and two latitude coordinates.

```
(7921.6623*arcsin(sqrt((sin((0.0174532925199433*station_lat
-0.0174532925199433*plt_lat)/2))**2
+cos(0.0174532925199433*plt_lat)
*cos(0.0174532925199433*station_lat)
*(sin((0.0174532925199433*station_long
-0.0174532925199433*plt_long)/2))**2)))
```

Finally, the data needs to be sorted in a logical and useful manner. In the below code the ORDER statement will sort the data by station code and then by the number of miles a pilots lives from each station.

```
ORDER BY arpt_cd, miles_away
```

Below is the complete code for creating the final data set for the emergency response directory. The code merges the pilot data and station data, calculates the distance each pilot lives for all Delta stations, keeps only records where the pilot is 50 miles or less away and sorts the data by Delta station and the number of miles away each pilot lives.

```
PROC SQL;
CREATE TABLE eval_1 AS
SELECT DISTINCT
    arpt_cd,
    empl_nbr,
    plt_first_name,
    plt_last_name,
    primary_ph,

    (7921.6623*arcsin(sqrt((sin((0.0174532925199433*station_lat
-0.0174532925199433*plt_lat)/2))**2
+cos(0.0174532925199433*plt_lat)
*cos(0.0174532925199433*station_lat)
*(sin((0.0174532925199433*station_long
-0.0174532925199433*plt_long)/2))**2)))
    FORMAT comma10.0 AS miles_away

FROM per_and_zip_1 AS a
INNER JOIN station_1 AS b
ON a.merge_on=b.merge_on
HAVING miles_away <= 50
ORDER BY arpt_cd, miles_away
;quit;
```

## CREATING A REPORT

The following code produces a report using PROC REPORT from the data set "final\_1" which contains the emergency response data.

```
PROC REPORT DATA=final_1 nowd SPLIT='*'

STYLE(HEADER)= {background=#DDDDDD foreground=#800517 just=right vjust=top
font_size=8PT font_weight=bold}
STYLE(COLUMN)= {background=#FFFFFF foreground=#000000 just=right vjust=top
font_size=8PT};

COLUMNS arpt_cd empl_nbr plt_first_name plt_last_name cat miles_away phone;

DEFINE arpt_cd / 'Station' STYLE={just=c};
DEFINE empl_nbr / 'Empl Nbr' STYLE={just=c};;
DEFINE plt_first_name / 'Pilot First Name' STYLE={just=c};
DEFINE plt_last_name / 'Pilot Last Name' STYLE={just=c};
DEFINE cat / 'Category' STYLE={just=c};
DEFINE miles_away / 'Plt Miles Away' STYLE={just=c};

RUN; QUIT;
```

Below is an example of a report created by PROC REPORT.

Station	Empl Nbr	Pilot First Name	Pilot Last Name	Category	Plt Miles Away
BWI	3054	ROBERT	KALBERT	DTW-DC9-B	2
BWI	9835	FRED	COOPER	NYC-73N-B	4
BWI	6589	TIM	FRANKLIN	NYC-777-A	4
BWI	7319	JULIE	DAVID	NYC-M88-A	7
BWI	8628	JANE	BOND	NYC-7ER-B	10
BWI	5502	DEWEY	SPENCER	NYC-73N-B	13
BWI	9430	JOHN	USHER	NYC-7ER-A	28
BWI	293	ROBERT	BLACK	NYC-73N-B	29
BWI	1161	JOHN	FISHER	DTW-7ER-B	34
BWI	6593	TODD	ROBBINS	NYC-73N-A	45
BWI	4677	WAYNE	HOPE	MSP-M88-A	49

## CONCLUSION

The above example has shown how SAS and longitudinal and latitudinal data along with zip codes can be used to create a first responders directory. SAS is a very useful tool to quickly calculate and evaluate the distance between two points. Additionally, SAS can create a very useful report to display the data.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Andy Hummel  
Delta Air Lines  
General Manager – Crew Resources  
Andrew.Hummel@Delta.com  
404-715-1270

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.