BI/Data Advisor Assignment 02

Grant T. Aguinaldo

November 3, 2020

1 Assignment 02: Phone Numbers (SAS Solution)

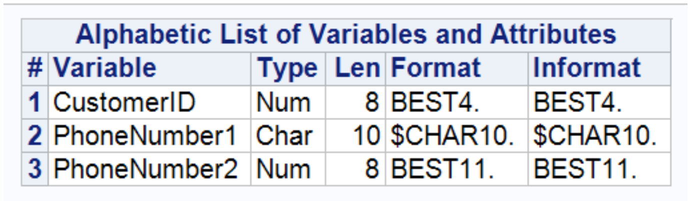
For this assignment, we were asked to developed a `SQL/SAS` script to return the correct phone number from two entries. Our approach to solving this problem is outlined below. The code below returns a table of all of the data types that is contained in the data set provided. From this query, we see that we have a mixture of `char` and `num` that we need to work with (see Figure 3). The link to the git repository that contains the code for this assignment is noted below:
https://github.com/grantaguinaldo/bi-work/blob/main/bi-work-assignment-02-sas-solution.sas

1. Understand the data types that we have in the dataset.
2. Look at the data that we have to get an understanding of what was provided.
3. Convert all of the datatypes to numeric so that we have a consistent format
4. Determine the length of each phone numbers
5. Write out case statements to select the proper phone number based on the correctness of the numbers provided. We like to note that we assume that a phone number is "correct" if it contains exactly 10 digits. The correct phone number is provided as the column named `CallNumber`.
6. If we were not able to find a correct phone number, we returned the number of `9999999999` since we cannot mix both `num` and `char` formats in the same column.

As previously mentioned, we first needed to convert all of the phone numbers to a numeric data type so we could calculate the length of each number. Again, we are calling a number "correct" if it contains exactly 10 digits. Once we converted all of the numbers to a numeric datatype, we then proceed to determine the length of each number provided. We stored the number lengths in the variables `PhoneNumber1Len` and `PhoneNumber2Len`. Once we had the columns `PhoneNumber1Len` and `PhoneNumber2Len`, we wrote wrote a `CASE` statement to pick out the correct number, and place that number in the column named `CallNumber` based on the values in the length columns. In the case statement shown below, if the length of `PhoneNumber1` was 10, we returned `PhoneNumber1`

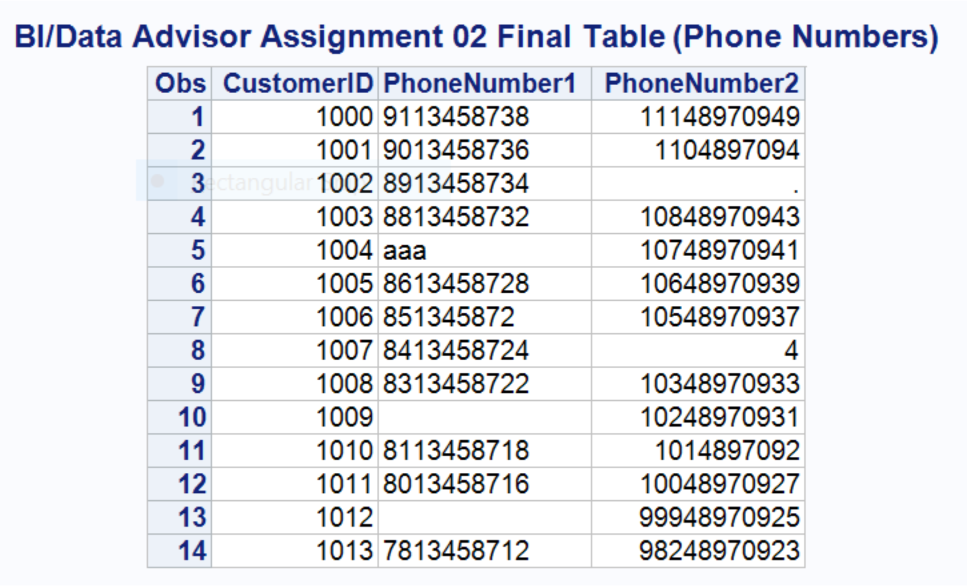
```
proc contents data=work.grantaguinaldo;
run;
```

Figure 1: PROC Contents SAS Code



Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Informat
1	CustomerID	Num	8	BEST4.	BEST4.
2	PhoneNumber1	Char	10	$CHAR10.	$CHAR10.
3	PhoneNumber2	Num	8	BEST11.	BEST11.

Figure 2: Resutls from PROC CONTENTS



BI/Data Advisor Assignment 02 Final Table (Phone Numbers)

Obs	CustomerID	PhoneNumber1	PhoneNumber2
1	1000	9113458738	11148970949
2	1001	9013458736	1104897094
3	1002	8913458734	.
4	1003	8813458732	10848970943
5	1004	aaa	10748970941
6	1005	8613458728	10648970939
7	1006	851345872	10548970937
8	1007	8413458724	4
9	1008	8313458722	10348970933
10	1009		10248970931
11	1010	8113458718	1014897092
12	1011	8013458716	10048970927
13	1012		99948970925
14	1013	7813458712	98248970923

Figure 3: Data Provided for Assignment

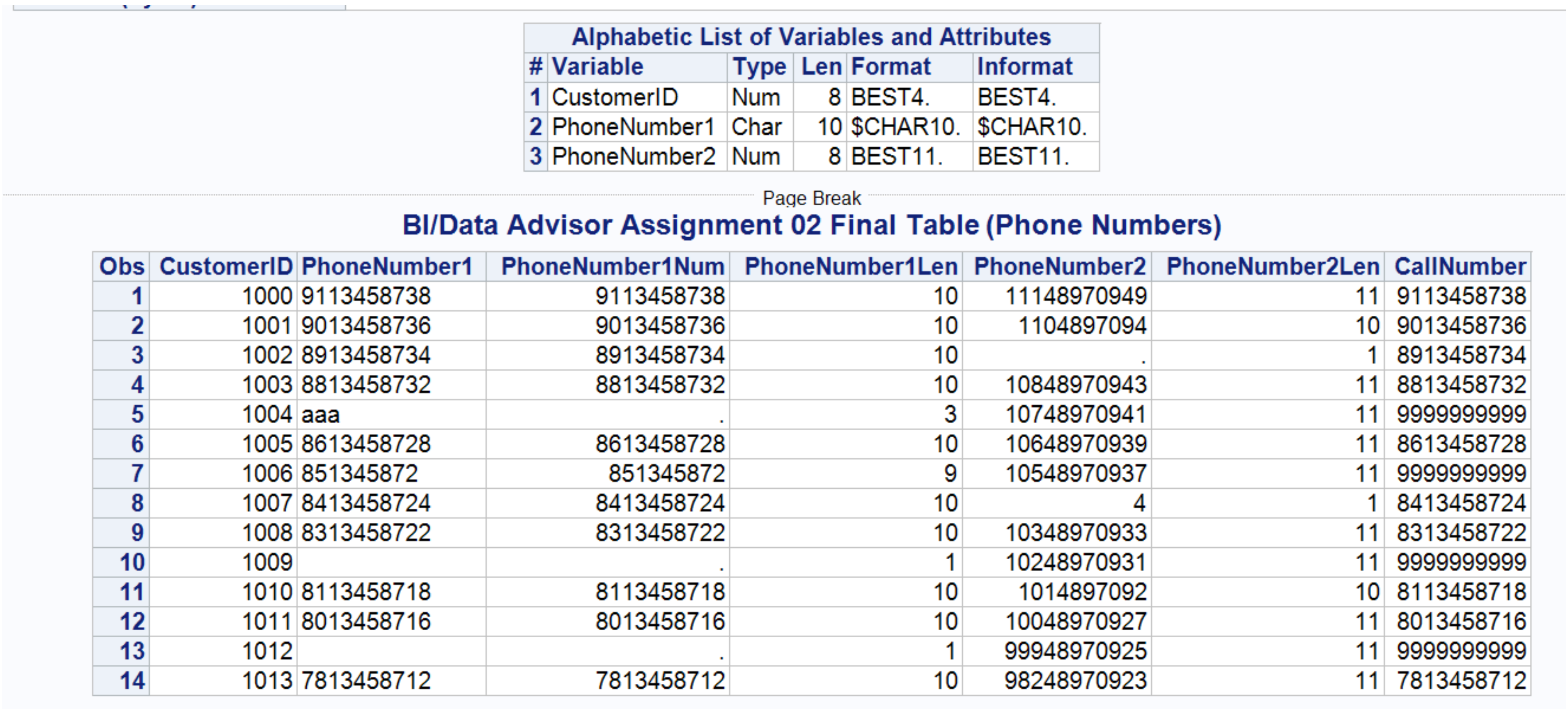
```
proc sql;
Create table assignment02 as
select

CustomerID,
PhoneNumber1,
input(PhoneNumber1, best12.) as PhoneNumber1Num,
length(compress(put(PhoneNumber1,$32.))) as PhoneNumber1Len,
PhoneNumber2,
length(compress(put(PhoneNumber2, 32.))) as PhoneNumber2Len,
(CASE

WHEN calculated PhoneNumber1Len = 10 THEN calculated PhoneNumber1Num
WHEN calculated PhoneNumber2Len = 10 THEN PhoneNumber2
ELSE 9999999999

/*9999999999 Means
that there is no
valid phone number
in the data since
neither Number 1 or
Number 2 contains 10 digits*/

END) AS CallNumber

from work.grantaguinaldo;
quit;
run;
```

Figure 4: PROC SQL SAS Code

as the correct number. If length of `PhoneNumber1` was not 10, we then looked at the length of `PhoneNumber2`. If the length of `PhoneNumber2` was 10, we returned that number as the correct number. Lastly, if neither the length of `PhoneNumber1` or `PhoneNumber2` was 10, we returned the number of `9999999999` to mean that neither of the numbers provided is a correct phone number. Using this methodology, we identified four accounts that do not have correct phone numbers (`CustomerID`: 1004, 1006, 1009 and 1012). The final table containing the “correct” phone numbers is shown in Figure 6.

```
/*Print Final Table*/
title "BI/Data Advisor Assignment 02 Final Table (Phone Numbers)";
proc print data=work.assignment02;
run;
```

Figure 5: PROC PRINT SAS Code



Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Informat
1	CustomerID	Num	8	BEST4.	BEST4.
2	PhoneNumber1	Char	10	$CHAR10.	$CHAR10.
3	PhoneNumber2	Num	8	BEST11.	BEST11.

BI/Data Advisor Assignment 02 Final Table (Phone Numbers)

Obs	CustomerID	PhoneNumber1	PhoneNumber1Num	PhoneNumber1Len	PhoneNumber2	PhoneNumber2Len	CallNumber
1	1000	9113458738	9113458738	10	11148970949	11	9113458738
2	1001	9013458736	9013458736	10	1104897094	10	9013458736
3	1002	8913458734	8913458734	10	.	1	8913458734
4	1003	8813458732	8813458732	10	10848970943	11	8813458732
5	1004	aaa	.	3	10748970941	11	9999999999
6	1005	8613458728	8613458728	10	10648970939	11	8613458728
7	1006	851345872	851345872	9	10548970937	11	9999999999
8	1007	8413458724	8413458724	10	4	1	8413458724
9	1008	8313458722	8313458722	10	10348970933	11	8313458722
10	1009		.	1	10248970931	11	9999999999
11	1010	8113458718	8113458718	10	1014897092	10	8113458718
12	1011	8013458716	8013458716	10	10048970927	11	8013458716
13	1012		.	1	99948970925	11	9999999999
14	1013	7813458712	7813458712	10	98248970923	11	7813458712

Figure 6: Final Table For Assignment 02