

# Comparative Tools in SAS: Analyzing Self-Reported versus Imputed Values

Neesha Nathwani & Sarah Woodruff

Department of Population Medicine, Harvard Pilgrim Health Care Institute, Boston, MA

## ABSTRACT

Imputation of data is often a critical step in compensating for missing values when analysis is due to be performed. The need for effective imputation has been demonstrated within the Sentinel Initiative, a program sponsored by the U.S. Food and Drug Administration (FDA) created to monitor the safety of FDA-regulated medical products via an active surveillance system. Since Sentinel functions as a distributed data network, data completeness can vary, emphasizing the importance of quality data imputation. The focus of this paper is on comparison of imputed values, in particular those reporting race, ethnicity, sex, and gender.

While current data partners are primarily using the Bayesian Improved Surname and Geocoding (BISG) method and a standardized imputation method developed by RTI International (RTI) for imputation, the tools and approaches available to compare reported versus imputed values vary much more widely. This paper describes and discusses the use of PROC COMPARE, implementation of crosstabulation particularly using PROC FREQ, and PROC MEANS, application of hash methods as well as regular expressions, and PROC SQL. Testing of these methods was done on fabricated descriptive reporting datasets which emulate those aggregated across the Sentinel population. This paper includes a comparative analysis of the methods to evaluate their effectiveness based on a range of priorities and relevant study outcomes.

## INTRODUCTION

Imputation of data is performed across disciplines as part of data management and analysis. Once imputed values are established, it makes sense to compare them to the originals to understand how often the process is introducing change and in what contexts. This paper explores the impact of imputation within the FDA's Sentinel Initiative, an active surveillance of regulated medical products once they are on the market. However, the goal of comparison extends far beyond the pharmaceutical or clinical realms, and the techniques explored here are applicable across all data types. Criteria for picking techniques is included with discussion of each of the procedures along with sample code that incorporates descriptive comments and sample output.

## BACKGROUND

Sentinel refers to an active surveillance system sponsored by the FDA to monitor the safety of FDA-regulated medical products. Administrative and clinical data are collected from various healthcare organizations, known as Data Partners (DPs), and maintained in a distributed data environment. While DPs must standardize their data to a Sentinel Common Data Model (SCDM) and store them as SAS datasets, they always maintain ownership of the usage and transfer of this data. In the distributed data environment, the Sentinel Operations Center (SOC) distributes analytic programs that the DPs run locally behind their firewalls, and aggregated results are sent back to the SOC. Patient privacy is always preserved.

A major area of concern in health studies is understanding how conditions, treatments, and outcomes impact different segments of the population. However, reliable sources of demographic data are needed to explore possible distinctions. Though efforts to collect this data in a consistent manner have improved over time, there is no one set way to do this. While the Sentinel Common Data Model (SCDM) standardizes demographic information like sex, race, and ethnicity within Sentinel, that does not compensate for values that are missing from the start. A supplied value of race and/or ethnicity may need to be compared to other factors related to the person like where they live or their surname.

This creates some of the many reasons reliable imputation may be necessary. The first consideration is likely to be the completion of missing values in order to increase the number of valid records which can be included in a study. Closely linked to this is the public health priority of working to ensure appropriate representation of any demographic groups that are present within a given population. In parallel with having complete data, there remains the question of how accurate the imputation of the demographic values is, and even when using the well-established algorithms discussed later, the comparison of reported vs. imputed values is an important part of understanding how the data can be used. While this discussion is based on race and ethnicity, the questions it raises could apply to any data where imputation is being used.

Examining health outcomes across different racial and ethnic groups can aid in identifying and addressing health disparities: preventable factors that disproportionately affect certain groups. Oftentimes, it is difficult to ascertain these health disparities due to limited or missing race and ethnicity data as most of this data is self-reported. To mitigate this, RAND corporation developed an algorithm that is widely used to impute these values, the Bayesian Improved Surname Geocoding (BISG) method.

RTI International, a nonprofit research institute, developed an imputation algorithm to aid in the identification of Hispanic and Asian/Pacific Islander (A/PI) individuals by drawing on additional information available from the Centers for Medicare and Medicaid Services (CMS) and the U.S. Census Bureau. Though the limitations still present within this imputation method were acknowledged, even at the time of its development it more than doubled the sensitivity of Hispanic identification and increased the sensitivity of A/PI identification by nearly 45%. Thus, the RTI imputation method remains a standard when imputing race and ethnicity.

More detailed information on these methods of imputation can be found in the links in the References section.

## INPUT DATASET FOR COMPARISON

To illustrate the methods, the same source data, dem\_l3\_catvars, is being used in each case. The dataset below represents the same format as the aggregated demographic datasets provided by various DPs for purposes of quality assurance and data characterization, but all values are fabricated for purposes of this demonstration; see the Appendix for the code used to create this data. Below find a representative sample of the raw data in the dataset, dem\_l3\_catvars, showing variable names, a variety of variable values across fabricated DPs, and description of variable values names and values.

|    | DP   | count | hispanic | imputedhispanic | race | imputedrace | sex |
|----|------|-------|----------|-----------------|------|-------------|-----|
| 1  | DP03 | 3219  | Y        | N               | 5    | 1           | M   |
| 2  | DP03 | 2647  | N        | N               | 6    | 4           | U   |
| 3  | DP01 | 6158  | N        | Y               | 3    | M           | M   |
| 4  | DP02 | 6758  | N        | U               | M    | 6           | F   |
| 5  | DP03 | 3204  | N        | N               | 0    | 5           | M   |
| 6  | DP03 | 4742  | U        | Y               | 4    | 0           | M   |
| 7  | DP01 | 1067  | Y        | Y               | M    | M           | M   |
| 8  | DP03 | 1179  | U        | Y               | 2    | 4           | U   |
| 9  | DP03 | 8609  | N        | N               | 5    |             | F   |
| 10 | DP02 | 6312  | Y        |                 | 1    | 6           | F   |
| 11 | DP02 | 7975  | N        | U               | 3    | 1           | M   |
| 12 | DP01 | 6506  | N        | N               | 4    | M           | M   |
| 13 | DP01 | 7175  | N        | N               | 4    | 6           | F   |
| 14 | DP02 | 4118  | Y        | N               | 4    | 0           | F   |
| 15 | DP02 | 9816  | N        | N               | 3    | 1           | M   |
| 16 | DP01 | 4171  | U        | N               | 4    | 0           | M   |
| 17 | DP03 | 847   | U        | U               | 1    | 5           | M   |
| 18 | DP01 | 5806  | Y        | N               | 5    | 4           | M   |
| 19 | DP02 | 7152  | N        | Y               | 5    | 4           | M   |
| 20 | DP01 | 4516  | Y        | U               | 3    | 5           | M   |
| 21 | DP03 | 2861  | U        | Y               | 5    | 6           | M   |
| 22 | DP03 | 7732  | Y        | N               | 5    | 5           | M   |
| 23 | DP03 | 7702  | N        | U               | 3    | 0           | F   |
| 24 | DP02 | 3877  | Y        | U               | 3    | M           | M   |
| 25 | DP03 | 1070  | Y        | N               | M    | 3           | F   |

DP: arbitrary Data Partner

Count: number of people with combinations of reported values, imputed values, and sex

hispanic: reported Hispanic; values include Y(yes), N(no), U(unknown)

imputedhispanic: imputed Hispanic; values same as above

race: reported race; values include 0 (unknown), 1 (American Indian or Alaska Native), 2(Asian), 3(Black or African American), 4(Native Hawaiian or other Pacific Islander), 5(White), M (Multi-racial)

imputedrace: imputed race; values same as above

sex: reported sex; values include M(male), F(female), U(unknown)

### Output 1. Sample of raw data

## METHODS OF COMPARISON

### PROC COMPARE

When comparing variables in SAS, whether for reported vs. imputed values or another scenario, the most obvious choice may be to use PROC COMPARE. Here the variable comparison is within the same dataset, so this is being done with the WITH statement:

```
title1 'Proc Compare of Reported vs. Imputed Variables' ;

/*SPECIFY NAME OF DATASET CONTAINING VARIABLES TO COMPARE*/
PROC COMPARE DATA = dem_l3_catvars;
  VAR race hispanic /*SPECIFY REPORTED VARIABLES*/ ;
  WITH imputedrace imputedhispanic /*SPECIFY IMPUTED VARIABLES*/ ;
RUN;
```

**Proc Compare of Reported vs. Imputed Variables**

| The COMPARE Procedure                           |            |                 |  |
|-------------------------------------------------|------------|-----------------|--|
| Comparisons of variables in WORK.DEM_L3_CATVARS |            |                 |  |
| (Method=EXACT)                                  |            |                 |  |
| Value Comparison Results for Variables          |            |                 |  |
| Obs                                             | Base Value | Compare Value   |  |
|                                                 | hispanic   | imputedhispanic |  |
|                                                 | —          | —               |  |
| 61                                              | Y          |                 |  |
| 62                                              | Y          | N               |  |
| 63                                              | U          |                 |  |
| 64                                              | N          |                 |  |
| 67                                              | N          | U               |  |
| 69                                              | U          | Y               |  |
| 72                                              | N          |                 |  |
| 74                                              | U          | Y               |  |
| 76                                              | U          | N               |  |

NOTE: The MAXPRINT=50 printing limit has been reached for the comparison of variable hispanic with imputedhispanic. No more values will be printed for this comparison.

### Output 2. Sample of PROC COMPARE default output

The output shown in Output 2 is not very useful to analyze, and by default, SAS only prints the first 50 differences for each comparison. PROC COMPARE has an option to produce an output dataset, and next, additional options are added to view a more thorough comparison. The code below produces a SAS dataset, in addition to the output in the SAS results viewer shown in Output 2.

```
/*SPECIFY DATASET CONTAINING VARIABLES TO COMPARE, AND OUTPUT DATASET*/
/*SPECIFY OPTIONS FOR OUTPUT*/
/*OUTBASE: INCLUDES VALUES FROM THE BASE (REPORTED) VARIABLE*/
/*OUTCOMP: INCLUDES VALUES FROM THE COMPARISON (IMPUTED) VARIABLE*/
/*OUTDIF: INCLUDES STRING SHOWING DIFFERENCE BETWEEN THE COMPARED VARIABLES*/
PROC COMPARE DATA = dem_l3_catvars OUT = comp OUTBASE OUTCOMP OUTDIF ;
  VAR race hispanic ;
  WITH imputedrace imputedhispanic ;
RUN ;
```

|    | Type of Observation | Observation Number | race | hispanic |
|----|---------------------|--------------------|------|----------|
| 1  | BASE                | 1 5                | Y    |          |
| 2  | COMPARE             | 1 1                | N    |          |
| 3  | DIF                 | 1 X                | X    |          |
| 4  | BASE                | 2 6                | N    |          |
| 5  | COMPARE             | 2 4                | N    |          |
| 6  | DIF                 | 2 X                | .    |          |
| 7  | BASE                | 3 3                | N    |          |
| 8  | COMPARE             | 3 M                | Y    |          |
| 9  | DIF                 | 3 X                | X    |          |
| 10 | BASE                | 4 M                | N    |          |
| 11 | COMPARE             | 4 6                | U    |          |
| 12 | DIF                 | 4 X                | X    |          |
| 13 | BASE                | 5 0                | N    |          |
| 14 | COMPARE             | 5 5                | N    |          |
| 15 | DIF                 | 5 X                | .    |          |
| 16 | BASE                | 6 4                | U    |          |
| 17 | COMPARE             | 6 0                | Y    |          |
| 18 | DIF                 | 6 X                | X    |          |
| 19 | BASE                | 7 M                | Y    |          |
| 20 | COMPARE             | 7 M                | Y    |          |
| 21 | DIF                 | 7 .                | .    |          |
| 22 | BASE                | 8 2                | U    |          |
| 23 | COMPARE             | 8 4                | Y    |          |
| 24 | DIF                 | 8 X                | X    |          |
| 25 | BASE                | 9 5                | N    |          |
| 26 | COMPARE             | 9 .                | N    |          |

### Output 3. Sample of PROC COMPARE dataset output with options OUTBASE OUTCOMP OUTDIF

The dataset output shows a vertical comparison of reported and imputed variables and displays an 'X' in the DIF rows signifying when there is a mismatch with the imputed value (see highlighted values).

PROC COMPARE also allows for stratification by categorical variables if the dataset is sorted by said categorical variables. See the code below:

```

/*SORT DATASET CONTAINING VARIABLES TO COMPARE BY THE CATEGORICAL VALUES*/
PROC SORT DATA = dem_l3_catvars ;
  BY dp sex ;
RUN ;

title1 'Proc Compare of Reported vs. Imputed Variables (Stratified by DP and
Sex)';

PROC COMPARE data = dem_l3_catvars OUT = comp_strat OUTBASE OUTCOMP OUTDIF ;
  /*ADD BY STATEMENT WITH CATEGORICAL VARIABLES FOR STRATIFICATION*/
  BY dp sex ;
  VAR race hispanic ;
  WITH imputedrace imputedhispanic ;
RUN ;

```

|    | Type of Observation | Observation Number | DP   | sex | race | hispanic |
|----|---------------------|--------------------|------|-----|------|----------|
| 1  | BASE                | 1                  | DP01 | F   | 4    | N        |
| 2  | COMPARE             | 1                  | DP01 | F   | 6    | N        |
| 3  | DIF                 | 1                  | DP01 | F   | X    | .        |
| 4  | BASE                | 2                  | DP01 | F   | 6    | N        |
| 5  | COMPARE             | 2                  | DP01 | F   | 2    | U        |
| 6  | DIF                 | 2                  | DP01 | F   | X    | X        |
| 7  | BASE                | 3                  | DP01 | F   | 5    | Y        |
| 8  | COMPARE             | 3                  | DP01 | F   | 2    | U        |
| 9  | DIF                 | 3                  | DP01 | F   | X    | X        |
| 10 | BASE                | 4                  | DP01 | F   | 2    | U        |
| 11 | COMPARE             | 4                  | DP01 | F   | 3    | U        |
| 12 | DIF                 | 4                  | DP01 | F   | X    | .        |
| 13 | BASE                | 5                  | DP01 | F   | 1    | U        |
| 14 | COMPARE             | 5                  | DP01 | F   | 1    | N        |
| 15 | DIF                 | 5                  | DP01 | F   | .    | X        |
| 16 | BASE                | 6                  | DP01 | F   | 6    | U        |
| 17 | COMPARE             | 6                  | DP01 | F   | 5    | U        |
| 18 | DIF                 | 6                  | DP01 | F   | X    | .        |
| 19 | BASE                | 7                  | DP01 | F   | M    | N        |
| 20 | COMPARE             | 7                  | DP01 | F   | M    | Y        |
| 21 | DIF                 | 7                  | DP01 | F   | .    | X        |
| 22 | BASE                | 8                  | DP01 | F   | 2    | N        |
| 23 | COMPARE             | 8                  | DP01 | F   | 0    | Y        |
| 24 | DIF                 | 8                  | DP01 | F   | X    | X        |
| 25 | BASE                | 9                  | DP01 | F   | 3    | N        |
| 26 | COMPARE             | 9                  | DP01 | F   | 5    | N        |

**Output 4. Sample of PROC COMPARE dataset output stratified by categorical variables with options OUTBASE OUTCOMP OUTDIF**

The output is lengthy in the results viewer window as up to 50 values of each stratification mismatch are displayed, but the dataset output as seen in Output 4 shows a clear vertical comparison of reported vs imputed variable value matches and mismatches stratified. Again, the highlighted 'X' values in the 'DIF' rows represent mismatches. For an additional analysis on this output dataset, PROC FREQ can be used to count the number of mismatches, or PROC COMPARE could be skipped entirely, going straight to using PROC FREQ for a comparison using crosstabulation.

## PROC FREQ: USING CROSTABULATION

A basic yet simple way to compare reported variables to their imputed counterpart is via PROC FREQ. The proc statement includes the specification of the dataset with the relevant variables and table statements to define the crosstabulations to be generated. Consider using the MISSING statement so that records without a value can be counted and compared.

```

title1 "PROC FREQ WITH ALL COUNTS AND PERCENTS" ;

/*SPECIFY DATASET THAT CONTAINS REPORTED AND IMPUTED VARIABLES*/
PROC FREQ DATA = dem_l3_catvars ;
  /*USE TABLE STATEMENT TO SPECIFY SPECIFIC CROSSTABULATION*/
  /*USE OF "MISSING" INCLUDES ANY MISSING IN THE TABULATION*/
  /*OTHERWISE MISSINGS ARE COUNTED SEPARATELY*/
  TABLE race * imputedrace / MISSING ;
  /*MULTIPLE TABLE STATEMENTS CAN BE INCLUDED IN ONE PROC*/
  TABLE hispanic * imputedhispanic / MISSING ;
RUN ;

```

Using this baseline implementation of PROC FREQ generates output in this format:

| The SAS System                             |                              |             |       |       |       |       |       |       |       |         |
|--------------------------------------------|------------------------------|-------------|-------|-------|-------|-------|-------|-------|-------|---------|
| The FREQ Procedure                         |                              |             |       |       |       |       |       |       |       |         |
| Frequency<br>Percent<br>Row Pct<br>Col Pct | Table of race by imputedrace |             |       |       |       |       |       |       |       |         |
|                                            | race                         | imputedrace |       |       |       |       |       |       |       |         |
|                                            |                              |             | 0     | 1     | 2     | 3     | 4     | 5     | 6     | M Total |
| 0                                          | 1                            | 4           | 2     | 1     | 0     | 0     | 1     | 1     | 0     | 10      |
|                                            | 1.00                         | 4.00        | 2.00  | 1.00  | 0.00  | 0.00  | 1.00  | 1.00  | 0.00  | 10.00   |
|                                            | 10.00                        | 40.00       | 20.00 | 10.00 | 0.00  | 0.00  | 10.00 | 10.00 | 0.00  |         |
|                                            | 14.29                        | 26.67       | 14.29 | 12.50 | 0.00  | 0.00  | 8.33  | 9.09  | 0.00  |         |
| 1                                          | 0                            | 1           | 2     | 1     | 0     | 0     | 2     | 2     | 3     | 11      |
|                                            | 0.00                         | 1.00        | 2.00  | 1.00  | 0.00  | 0.00  | 2.00  | 2.00  | 3.00  | 11.00   |
|                                            | 0.00                         | 9.09        | 18.18 | 9.09  | 0.00  | 0.00  | 18.18 | 18.18 | 27.27 |         |
|                                            | 0.00                         | 6.67        | 14.29 | 12.50 | 0.00  | 0.00  | 16.67 | 18.18 | 21.43 |         |
| 2                                          | 3                            | 2           | 0     | 2     | 4     | 1     | 1     | 0     | 3     | 16      |
|                                            | 3.00                         | 2.00        | 0.00  | 2.00  | 4.00  | 1.00  | 1.00  | 0.00  | 3.00  | 16.00   |
|                                            | 18.75                        | 12.50       | 0.00  | 12.50 | 25.00 | 6.25  | 6.25  | 0.00  | 18.75 |         |
|                                            | 42.86                        | 13.33       | 0.00  | 25.00 | 33.33 | 14.29 | 8.33  | 0.00  | 21.43 |         |
| 3                                          | 1                            | 2           | 2     | 1     | 3     | 0     | 3     | 1     | 2     | 15      |
|                                            | 1.00                         | 2.00        | 2.00  | 1.00  | 3.00  | 0.00  | 3.00  | 1.00  | 2.00  | 15.00   |
|                                            | 6.67                         | 13.33       | 13.33 | 6.67  | 20.00 | 0.00  | 20.00 | 6.67  | 13.33 |         |
|                                            | 14.29                        | 13.33       | 14.29 | 12.50 | 25.00 | 0.00  | 25.00 | 9.09  | 14.29 |         |
| 4                                          | 0                            | 5           | 2     | 0     | 0     | 2     | 0     | 2     | 1     | 12      |
|                                            | 0.00                         | 5.00        | 2.00  | 0.00  | 0.00  | 2.00  | 0.00  | 2.00  | 1.00  | 12.00   |
|                                            | 0.00                         | 41.67       | 16.67 | 0.00  | 0.00  | 16.67 | 0.00  | 16.67 | 8.33  |         |
|                                            | 0.00                         | 33.33       | 14.29 | 0.00  | 0.00  | 28.57 | 0.00  | 18.18 | 7.14  |         |
| 5                                          | 2                            | 0           | 3     | 2     | 2     | 2     | 4     | 2     | 1     | 18      |
|                                            | 2.00                         | 0.00        | 3.00  | 2.00  | 2.00  | 2.00  | 4.00  | 2.00  | 1.00  | 18.00   |
|                                            | 11.11                        | 0.00        | 16.67 | 11.11 | 11.11 | 11.11 | 22.22 | 11.11 | 5.56  |         |
|                                            | 28.57                        | 0.00        | 21.43 | 25.00 | 16.67 | 28.57 | 33.33 | 18.18 | 7.14  |         |
| 6                                          | 0                            | 1           | 1     | 1     | 1     | 2     | 1     | 2     | 2     | 11      |
|                                            | 0.00                         | 1.00        | 1.00  | 1.00  | 1.00  | 2.00  | 1.00  | 2.00  | 2.00  | 11.00   |
|                                            | 0.00                         | 9.09        | 9.09  | 9.09  | 9.09  | 18.18 | 9.09  | 18.18 | 18.18 |         |
|                                            | 0.00                         | 6.67        | 7.14  | 12.50 | 8.33  | 28.57 | 8.33  | 18.18 | 14.29 |         |
| M                                          | 0                            | 0           | 2     | 0     | 2     | 0     | 0     | 1     | 2     | 7       |
|                                            | 0.00                         | 0.00        | 2.00  | 0.00  | 2.00  | 0.00  | 0.00  | 1.00  | 2.00  | 7.00    |
|                                            | 0.00                         | 0.00        | 28.57 | 0.00  | 28.57 | 0.00  | 0.00  | 14.29 | 28.57 |         |
|                                            | 0.00                         | 0.00        | 14.29 | 0.00  | 16.67 | 0.00  | 0.00  | 9.09  | 14.29 |         |
| Total                                      | 7                            | 15          | 14    | 8     | 12    | 7     | 12    | 11    | 14    | 100     |
|                                            | 7.00                         | 15.00       | 14.00 | 8.00  | 12.00 | 7.00  | 12.00 | 11.00 | 14.00 | 100.00  |

| Frequency<br>Percent<br>Row Pct<br>Col Pct | Table of hispanic by imputedhispanic |                 |       |       |         |
|--------------------------------------------|--------------------------------------|-----------------|-------|-------|---------|
|                                            | hispanic                             | imputedhispanic |       |       |         |
|                                            |                                      |                 | N     | U     | Y Total |
| N                                          | 2                                    | 19              | 9     | 13    | 43      |
|                                            | 2.00                                 | 19.00           | 9.00  | 13.00 | 43.00   |
|                                            | 4.65                                 | 44.19           | 20.93 | 30.23 |         |
|                                            | 33.33                                | 45.24           | 37.50 | 46.43 |         |
| U                                          | 2                                    | 10              | 6     | 11    | 29      |
|                                            | 2.00                                 | 10.00           | 6.00  | 11.00 | 29.00   |
|                                            | 6.90                                 | 34.48           | 20.69 | 37.93 |         |
|                                            | 33.33                                | 23.81           | 25.00 | 39.29 |         |
| Y                                          | 2                                    | 13              | 9     | 4     | 28      |
|                                            | 2.00                                 | 13.00           | 9.00  | 4.00  | 28.00   |
|                                            | 7.14                                 | 46.43           | 32.14 | 14.29 |         |
|                                            | 33.33                                | 30.95           | 37.50 | 14.29 |         |
| Total                                      | 6                                    | 42              | 24    | 28    | 100     |
|                                            | 6.00                                 | 42.00           | 24.00 | 28.00 | 100.00  |

## Output 5. Sample of PROC FREQ default output with MISSING included

However, when making these comparisons the default output of PROC FREQ is often too cluttered, as the standard includes not only the counts (frequencies) of each crosstabulated category but also the overall percent, the percent based on the row, and the percent based on the column. To suppress the percents, use the desired mix of the NOCOL, NOROW, and NOPERCENT options on the table statement.

```

title1 "PROC FREQ WITH CUMULATIVE COUNTS AND PERCENTS REMOVED" ;

PROC FREQ DATA = dem_l3_catvars ;
    /*NOCOL SUPPRESSES COLUMN PERCENTAGE*/
    /*NOROW SUPPRESSES ROW PERCENTAGE*/
    /*NOPERCENT SUPPRESSES OVERALL PERCENTAGE*/
    TABLE race * imputedrace / MISSING NOCOL NOROW NOPERCENT ;
    TABLE hispanic * imputedhispanic / MISSING NOCOL NOROW NOPERCENT ;
RUN ;

```

Removing the percents generates a cleaner comparison for data where typically the frequencies are the focus.

PROC FREQ WITH CUMULATIVE COUNTS AND PERCENTS REMOVED

The FREQ Procedure

|           |                              |    |    |   |    |   |    |    |       |  |
|-----------|------------------------------|----|----|---|----|---|----|----|-------|--|
| Frequency | Table of race by imputedrace |    |    |   |    |   |    |    |       |  |
|           | imputedrace                  |    |    |   |    |   |    |    |       |  |
| race      | 0                            | 1  | 2  | 3 | 4  | 5 | 6  | M  | Total |  |
| 0         | 1                            | 4  | 2  | 1 | 0  | 0 | 1  | 1  | 10    |  |
| 1         | 0                            | 1  | 2  | 1 | 0  | 0 | 2  | 2  | 11    |  |
| 2         | 3                            | 2  | 0  | 2 | 4  | 1 | 1  | 0  | 16    |  |
| 3         | 1                            | 2  | 2  | 1 | 3  | 0 | 3  | 1  | 15    |  |
| 4         | 0                            | 5  | 2  | 0 | 0  | 2 | 0  | 2  | 12    |  |
| 5         | 2                            | 0  | 3  | 2 | 2  | 2 | 4  | 2  | 18    |  |
| 6         | 0                            | 1  | 1  | 1 | 1  | 2 | 1  | 2  | 11    |  |
| M         | 0                            | 0  | 2  | 0 | 2  | 0 | 0  | 1  | 7     |  |
| Total     | 7                            | 15 | 14 | 8 | 12 | 7 | 12 | 14 | 100   |  |

  

|           |                                      |    |    |       |  |
|-----------|--------------------------------------|----|----|-------|--|
| Frequency | Table of hispanic by imputedhispanic |    |    |       |  |
|           | imputedhispanic                      |    |    |       |  |
| hispanic  | N                                    | U  | Y  | Total |  |
| N         | 2                                    | 19 | 9  | 43    |  |
| U         | 2                                    | 10 | 6  | 29    |  |
| Y         | 2                                    | 13 | 9  | 28    |  |
| Total     | 6                                    | 42 | 24 | 100   |  |

## Output 6. PROC FREQ output with MISSING NOCOL NOROW NOPERCENT included

The TABLE statement can also be set up to run multiple crosstabulations of a variable against a list of other variables. This allows for comparisons between reported and imputed to be shown alongside comparisons to any other relevant variable or even to facilitate multiple comparisons if multiple methods of imputation were implemented and then need to be evaluated.

```

title1 "PROC FREQ WITH ONE VARIABLE CROSSTABULATED WITH MULTIPLE VARIABLES" ;

PROC FREQ DATA = dem_l3_catvars ;
    /*PUTTING MULTIPLE VARIABLES IN PARENTHESES ALLOWS ONE VARIABLE*/
    /* TO BE CROSSTABULATED WITH EACH ONE IN THE SPECIFIED LIST */
    TABLE race * ( imputedrace hispanic imputedhispanic ) / MISSING NOCOL NOROW
        NOPERCENT ;
RUN ;

```

This produces a series of crosstabulations from just one table statement. Particularly for large scale comparisons, this type of coding efficiency can be very helpful.

#### PROC FREQ WITH ONE VARIABLE CROSSTABULATED WITH MULTIPLE VARIABLES

The FREQ Procedure

| Frequency | Table of race by imputedrace |    |    |   |    |   |    |    |       |  |
|-----------|------------------------------|----|----|---|----|---|----|----|-------|--|
| race      | imputedrace                  |    |    |   |    |   |    |    |       |  |
|           | 0                            | 1  | 2  | 3 | 4  | 5 | 6  | M  | Total |  |
| 0         | 1                            | 4  | 2  | 1 | 0  | 0 | 1  | 1  | 10    |  |
| 1         | 0                            | 1  | 2  | 1 | 0  | 0 | 2  | 2  | 11    |  |
| 2         | 3                            | 2  | 0  | 2 | 4  | 1 | 1  | 0  | 16    |  |
| 3         | 1                            | 2  | 2  | 1 | 3  | 0 | 3  | 1  | 15    |  |
| 4         | 0                            | 5  | 2  | 0 | 0  | 2 | 0  | 2  | 12    |  |
| 5         | 2                            | 0  | 3  | 2 | 2  | 2 | 4  | 2  | 18    |  |
| 6         | 0                            | 1  | 1  | 1 | 1  | 2 | 1  | 2  | 11    |  |
| M         | 0                            | 0  | 2  | 0 | 2  | 0 | 0  | 1  | 7     |  |
| Total     | 7                            | 15 | 14 | 8 | 12 | 7 | 12 | 11 | 100   |  |

| Frequency | Table of race by hispanic |    |    |       |  |
|-----------|---------------------------|----|----|-------|--|
| race      | hispanic                  |    |    |       |  |
|           | N                         | U  | Y  | Total |  |
| 0         | 5                         | 2  | 3  | 10    |  |
| 1         | 3                         | 5  | 3  | 11    |  |
| 2         | 9                         | 4  | 3  | 16    |  |
| 3         | 7                         | 2  | 6  | 15    |  |
| 4         | 4                         | 6  | 2  | 12    |  |
| 5         | 5                         | 7  | 6  | 18    |  |
| 6         | 6                         | 2  | 3  | 11    |  |
| M         | 4                         | 1  | 2  | 7     |  |
| Total     | 43                        | 29 | 28 | 100   |  |

| Frequency | Table of race by imputedhispanic |    |    |    |       |
|-----------|----------------------------------|----|----|----|-------|
| race      | imputedhispanic                  |    |    |    | Total |
|           | N                                | U  | Y  |    |       |
| 0         | 0                                | 5  | 2  | 3  | 10    |
| 1         | 2                                | 1  | 3  | 5  | 11    |
| 2         | 1                                | 5  | 5  | 5  | 16    |
| 3         | 0                                | 7  | 5  | 3  | 15    |
| 4         | 0                                | 8  | 1  | 3  | 12    |
| 5         | 2                                | 9  | 2  | 5  | 18    |
| 6         | 1                                | 4  | 4  | 2  | 11    |
| M         | 0                                | 3  | 2  | 2  | 7     |
| Total     | 6                                | 42 | 24 | 28 | 100   |

#### Output 6. PROC FREQ output with one variable crosstabulated against multiple variables

This approach also allows a categorical variable to be applied to multiple variables of interest. For this sample data, there are multiple data partners, and a reported or imputed variable could be counted for comparison between those data partners.

```

title1 "PROC FREQ WITH CATEGORICAL VARIABLE USED TO CROSSTABULATE" ;

PROC FREQ DATA = dem_l3_catvars ;
  /*COUNTING BY ADDITIONAL CATEGORICAL VARIABLE*/
  TABLE dp * ( race imputedrace hispanic imputedhispanic ) /
    MISSING NOCOL NOROW NOPERCENT ;
RUN ;

```

Here is the output; note the similarities to the prior example:

PROC FREQ WITH ONE VARIABLE CROSSTABULATED WITH MULTIPLE VARIABLES

The FREQ Procedure

|           |                     |    |    |    |    |    |    |    |   |       |
|-----------|---------------------|----|----|----|----|----|----|----|---|-------|
| Frequency | Table of DP by race |    |    |    |    |    |    |    |   |       |
|           | race                |    |    |    |    |    |    |    |   |       |
|           | DP                  | 0  | 1  | 2  | 3  | 4  | 5  | 6  | M | Total |
|           | DP01                | 4  | 2  | 5  | 4  | 5  | 4  | 4  | 4 | 32    |
|           | DP02                | 4  | 4  | 5  | 6  | 2  | 6  | 3  | 2 | 32    |
|           | DP03                | 2  | 5  | 6  | 5  | 5  | 8  | 4  | 1 | 36    |
|           | Total               | 10 | 11 | 16 | 15 | 12 | 18 | 11 | 7 | 100   |

|           |                            |   |    |    |   |    |   |    |    |       |
|-----------|----------------------------|---|----|----|---|----|---|----|----|-------|
| Frequency | Table of DP by imputedrace |   |    |    |   |    |   |    |    |       |
|           | imputedrace                |   |    |    |   |    |   |    |    |       |
|           | DP                         | 0 | 1  | 2  | 3 | 4  | 5 | 6  | M  | Total |
|           | DP01                       | 1 | 6  | 4  | 4 | 3  | 2 | 6  | 2  | 32    |
|           | DP02                       | 1 | 4  | 6  | 2 | 4  | 2 | 1  | 7  | 32    |
|           | DP03                       | 5 | 5  | 4  | 2 | 5  | 3 | 5  | 2  | 36    |
|           | Total                      | 7 | 15 | 14 | 8 | 12 | 7 | 12 | 14 | 100   |

|           |                         |    |    |    |       |
|-----------|-------------------------|----|----|----|-------|
| Frequency | Table of DP by hispanic |    |    |    |       |
|           | hispanic                |    |    |    |       |
|           | DP                      | N  | U  | Y  | Total |
|           | DP01                    | 17 | 6  | 9  | 32    |
|           | DP02                    | 16 | 7  | 9  | 32    |
|           | DP03                    | 10 | 16 | 10 | 36    |
|           | Total                   | 43 | 29 | 28 | 100   |

|           |                                |   |    |    |       |
|-----------|--------------------------------|---|----|----|-------|
| Frequency | Table of DP by imputedhispanic |   |    |    |       |
|           | imputedhispanic                |   |    |    |       |
|           | DP                             | N | U  | Y  | Total |
|           | DP01                           | 1 | 13 | 9  | 32    |
|           | DP02                           | 3 | 14 | 7  | 32    |
|           | DP03                           | 2 | 15 | 8  | 36    |
|           | Total                          | 6 | 42 | 24 | 100   |

## Output 7. PROC FREQ output with crosstabulated by categorical variable

Fun note: To remove the default title of “The FREQ Procedure”, include the following ODS statement:

```
ODS NOPROCTITLE ;
```

In selecting a method of crosstabulation, PROC TABULATE is also an option if a numeric variable is also available for counting relevant records. Such a “counter” variable could always be added to the data, but this discussion focuses on methods that can be applied to the data as it is.

## PROC MEANS

Another method of comparison is possible through PROC MEANS. This proc is advantageous because it works well with character variables and provides counts that are easy to understand. Various forms of crosstabulation are also straightforward to build. Continuing to include the MISSING option allows comparisons where a value is not available to be included and to keep counts of observations the same.

The proc statements specify the dataset with additional statements to specify particular variables, crosstabulations, and output datasets with the results of the proc.

```
/* DATASET WITH REPORTED AND IMPUTED VARIABLES WITHOUT COUNT OR DP*/  
PROC MEANS data = imputed ;  
  /*USE CLASS STATEMENT TO SPECIFY VARIABLES OF INTEREST*/  
  CLASS race imputedrace / MISSING ;  
  /*USE TYPES STATEMENT TO SPECIFY CROSSTABULATIONS*/  
  TYPES () race*imputedrace ;  
  /*USE OUTPUT STATEMENT TO SPECIFY OUTPUT DATASET*/  
  OUTPUT OUT = race_imputedrace ;  
RUN ;
```

The basic output from the proc includes the following:

| The SAS System      |             |       |
|---------------------|-------------|-------|
| The MEANS Procedure |             |       |
|                     |             |       |
| N Obs               |             |       |
| 100                 |             |       |
|                     |             |       |
| race                | imputedrace | N Obs |
| 0                   |             | 1     |
|                     | 0           | 4     |
|                     | 1           | 2     |
|                     | 2           | 1     |
|                     | 5           | 1     |
|                     | 6           | 1     |
| 1                   | 0           | 1     |
|                     | 1           | 2     |
|                     | 2           | 1     |
|                     | 5           | 2     |
|                     | 6           | 2     |
|                     | M           | 3     |
| 2                   |             | 3     |
|                     | 0           | 2     |
|                     | 2           | 2     |
|                     | 3           | 4     |
|                     | 4           | 1     |
|                     | 5           | 1     |
| 3                   | M           | 3     |
|                     |             | 1     |
|                     | 0           | 2     |
|                     | 1           | 2     |

Output 8. PROC MEANS default output

The output starts with the overall count of records and then displays the requested crosstabulations. The value of the first variable – in this case RACE – will not be repeated in each row, but instead is left blank after the first occurrence until the value changes. These are displayed in sorted order as are the values of the variable being crossed with the first with counts given for each pairing. This output in Output 8 is partially truncated but would continue with all values of RACE crossed against IMPUTEDRACE.

Since other work might need to be done with these values, producing an output dataset is often useful. The output data set can be limited to observations with the highest values if the NWAY option is included, but the goal here is to see everything that is produced first; consider whether NWAY would be useful once see all the output.

|    | race | imputedrace | _TYPE_ | _FREQ_ |
|----|------|-------------|--------|--------|
| 1  |      |             | 0      | 100    |
| 2  |      |             | 3      | 1      |
| 3  | 0    | 0           | 3      | 4      |
| 4  | 0    | 1           | 3      | 2      |
| 5  | 0    | 2           | 3      | 1      |
| 6  | 0    | 5           | 3      | 1      |
| 7  | 0    | 6           | 3      | 1      |
| 8  | 1    | 0           | 3      | 1      |
| 9  | 1    | 1           | 3      | 2      |
| 10 | 1    | 2           | 3      | 1      |
| 11 | 1    | 5           | 3      | 2      |
| 12 | 1    | 6           | 3      | 2      |
| 13 | 1    | M           | 3      | 3      |
| 14 | 2    |             | 3      | 3      |
| 15 | 2    | 0           | 3      | 2      |
| 16 | 2    | 2           | 3      | 2      |
| 17 | 2    | 3           | 3      | 4      |
| 18 | 2    | 4           | 3      | 1      |
| 19 | 2    | 5           | 3      | 1      |
| 20 | 2    | M           | 3      | 3      |
| 21 | 3    |             | 3      | 1      |
| 22 | 3    | 0           | 3      | 2      |
| 23 | 3    | 1           | 3      | 2      |
| 24 | 3    | 2           | 3      | 1      |
| 25 | 3    | 3           | 3      | 3      |
| 26 | 3    | 5           | 3      | 3      |
| 27 | 3    | 6           | 3      | 1      |
| 28 | 3    | M           | 3      | 2      |

## Output 9. PROC MEANS output dataset

The first line of the table is once again the total overall records and then each combination of the variables being crossed are included in sorted order. The final column, \_FREQ\_, displays the counts of the combinations. The automatically generated variable, \_TYPE\_, may be of use and is worth understanding. PROC MEANS defaults \_TYPE\_ to a value of 0 for the overall, and then if variables are being counted individually, the first variable would have a \_TYPE\_ value of 1 while the second would have a value of 2, etc. In this example, two variables are being crosstabulated, so the proc assigns a value of 3 to \_TYPE\_, indicating a third level or step after the existence of the two variables involved in the crosstabulation. If you need to select certain categories of counts or are working with even more extensive crosstabulations, understanding what the value of \_TYPE\_ indicates can facilitate that subsetting.

Further breakdown of the counts is possible with the use of a BY statement. Variables should be sorted prior to PROC MEANS on the basis of the variables to be used in the BY statement.

```
PROC MEANS DATA = imputed_sortbysex ;
/* USE BY STATEMENT TO SUBSET COUNTS OR CROSSTABULATION */
BY sex ;
CLASS race imputedrace / MISSING ;
TYPES () race * imputedrace ;
OUTPUT OUT = race_imputedrace_sex ;
RUN ;
```

The SAS System

The MEANS Procedure

sex=F

|       |
|-------|
| N Obs |
| 38    |

sex=F

| race | imputedrace | N Obs |
|------|-------------|-------|
| 0    | 0           | 1     |
| 1    | 0           | 1     |
|      | 1           | 2     |
|      | 5           | 1     |
|      | 6           | 2     |
|      | M           | 2     |
| 2    |             | 2     |
|      | 0           | 1     |
|      | 3           | 2     |
|      | M           | 1     |
| 3    |             | 1     |
|      | 0           | 1     |
|      | 3           | 2     |
|      | 5           | 1     |
| 4    | 0           | 1     |
|      | 1           | 1     |
|      | 4           | 1     |
|      | 6           | 1     |
| 5    |             | 2     |
|      | 1           | 1     |

### Output 10. PROC MEANS output with BY statement

In this example, the data is split by values of SEX before being counted and crosstabulated.

As before, the overall count for the given category is provided first followed by the crosstabulation of the requested variables as they appear under that category.

The output dataset follows a similar structure, but now has the BY statement variable as its first column followed by the variables being crosstabulated in its second and third. Values of `_TYPE_` and `_FREQ_` follow the same format and have the same meaning as in the original version.

|    | sex | race | imputedrace | _TYPE_ | _FREQ_ |
|----|-----|------|-------------|--------|--------|
| 1  | F   |      |             | 0      | 38     |
| 2  | F   | 0    | 0           | 3      | 1      |
| 3  | F   | 1    | 0           | 3      | 1      |
| 4  | F   | 1    | 1           | 3      | 2      |
| 5  | F   | 1    | 5           | 3      | 1      |
| 6  | F   | 1    | 6           | 3      | 2      |
| 7  | F   | 1    | M           | 3      | 2      |
| 8  | F   | 2    |             | 3      | 2      |
| 9  | F   | 2    | 0           | 3      | 1      |
| 10 | F   | 2    | 3           | 3      | 2      |
| 11 | F   | 2    | M           | 3      | 1      |
| 12 | F   | 3    |             | 3      | 1      |
| 13 | F   | 3    | 0           | 3      | 1      |
| 14 | F   | 3    | 3           | 3      | 2      |
| 15 | F   | 3    | 5           | 3      | 1      |
| 16 | F   | 4    | 0           | 3      | 1      |
| 17 | F   | 4    | 1           | 3      | 1      |
| 18 | F   | 4    | 4           | 3      | 1      |
| 19 | F   | 4    | 6           | 3      | 1      |
| 20 | F   | 5    |             | 3      | 2      |
| 21 | F   | 5    | 1           | 3      | 1      |
| 22 | F   | 5    | 2           | 3      | 2      |
| 23 | F   | 5    | 3           | 3      | 1      |
| 24 | F   | 5    | 5           | 3      | 1      |
| 25 | F   | 6    | 2           | 3      | 1      |
| 26 | F   | 6    | 5           | 3      | 1      |
| 27 | F   | M    | 1           | 3      | 1      |
| 28 | F   | M    | 3           | 3      | 2      |
| 29 | F   | M    | 6           | 3      | 1      |
| 30 | F   | M    | M           | 3      | 1      |
| 31 | M   |      |             | 0      | 53     |
| 32 | M   | 0    | 0           | 3      | 3      |
| 33 | M   | 0    | 1           | 3      | 2      |

### Output 11. PROC MEANS with BY statement output dataset

The same code could be applied to HISPANIC and IMPUTEDHISPANIC and would work in the same way. Since the output would be of the same format, it is being omitted, and just the code is included here.

```
PROC MEANS DATA = imputed ;
  CLASS hispanic imputedhispanic / MISSING ;
  TYPES () hispanic * imputedhispanic ;
  OUTPUT OUT = hispanic_imputedhispanic ;
RUN ;
```

```
PROC MEANS DATA = imputed_sortbysex ;
  BY sex ;
  CLASS hispanic imputedhispanic / MISSING ;
  TYPES () hispanic * imputedhispanic ;
  OUTPUT OUT = hispanic_imputedhispanic_sex ;
RUN ;
```

### PROC SQL

Another method to consider reported vs. imputed variable disparities is PROC SQL, which can be used for a variety of analyses, but here the comparison is done using counts and with a mismatch analysis. Unlike other procedures explored here, PROC SQL automatically includes records with missing values without an additional option to do so being required.

See the code below for a basic crosstabulation:

```
title1 'Proc Sql: Counts of Reported vs. Imputed Variables';
```

```
PROC SQL;
  SELECT race
         , imputedrace
         , count(*) AS count
  FROM dem_l3_catvars
  GROUP BY race, imputedrace ;

  SELECT hispanic
         , imputedhispanic
         , count(*) AS count
  FROM dem_l3_catvars
  GROUP BY hispanic, imputedhispanic ;
QUIT;
```

**Proc Sql: Counts of Reported vs. Imputed Variables**

| race | imputedrace | count |
|------|-------------|-------|
| 0    |             | 1     |
| 0    | 0           | 4     |
| 0    | 1           | 2     |
| 0    | 2           | 1     |
| 0    | 5           | 1     |
| 0    | 6           | 1     |
| 1    | 0           | 1     |
| 1    | 1           | 2     |
| 1    | 2           | 1     |
| 1    | 5           | 2     |
| 1    | 6           | 2     |
| 1    | M           | 3     |
| 2    |             | 3     |
| 2    | 0           | 2     |
| 2    | 2           | 2     |

| hispanic | imputedhispanic | count |
|----------|-----------------|-------|
| N        |                 | 2     |
| N        | N               | 19    |
| N        | U               | 9     |
| N        | Y               | 13    |
| U        |                 | 2     |
| U        | N               | 10    |
| U        | U               | 6     |
| U        | Y               | 11    |
| Y        |                 | 2     |
| Y        | N               | 13    |
| Y        | U               | 9     |
| Y        | Y               | 4     |

#### Output 12. PROC SQL default output with counts

Further enhance this output with stratification by categorical variables. The beauty of PROC SQL is that it does not need to be sorted to produce sorted output, so this can be easily done by modifying the code as follows:

```
title1 'Proc Sql: Counts of Reported vs. Imputed Variables (Stratified by DP
and Sex)';
```

```
PROC SQL ;
  SELECT dp
    , sex
    , race
    , imputedrace
    , count(*) AS count
  FROM dem_l3_catvars
  GROUP BY dp, sex, race, imputedrace
  ORDER BY dp, sex ;

  SELECT dp
    , sex
    , hispanic
    , imputedhispanic
    , count(*) AS count
  FROM dem_l3_catvars
  GROUP BY dp, sex, hispanic, imputedhispanic
  ORDER BY dp, sex ;
QUIT ;
```

The output is as such:

**Proc Sql: Counts of Reported vs. Imputed Variables (Stratified by Dp and Sex)**

| DP   | sex | race | imputedrace | count |
|------|-----|------|-------------|-------|
| DP01 | F   | 1    | 1           | 1     |
| DP01 | F   | 2    |             | 1     |
| DP01 | F   | 2    | 0           | 1     |
| DP01 | F   | 2    | 3           | 1     |
| DP01 | F   | 3    | 5           | 1     |
| DP01 | F   | 4    | 1           | 1     |
| DP01 | F   | 4    | 6           | 1     |
| DP01 | F   | 5    | 2           | 2     |
| DP01 | F   | 5    | 5           | 1     |
| DP01 | F   | 6    | 2           | 1     |
| DP01 | F   | 6    | 5           | 1     |
| DP01 | F   | M    | 3           | 1     |
| DP01 | F   | M    | M           | 1     |
| DP01 | M   | 0    | 0           | 3     |
| DP01 | M   | 0    | 1           | 1     |

| DP   | sex | hispanic | imputedhispanic | count |
|------|-----|----------|-----------------|-------|
| DP01 | F   | N        | N               | 4     |
| DP01 | F   | N        | U               | 3     |
| DP01 | F   | N        | Y               | 2     |
| DP01 | F   | U        | N               | 1     |
| DP01 | F   | U        | U               | 2     |
| DP01 | F   | Y        | N               | 1     |
| DP01 | F   | Y        | U               | 1     |
| DP01 | M   | N        | N               | 2     |
| DP01 | M   | N        | Y               | 5     |
| DP01 | M   | U        | N               | 2     |
| DP01 | M   | Y        |                 | 1     |
| DP01 | M   | Y        | N               | 2     |
| DP01 | M   | Y        | U               | 2     |
| DP01 | M   | Y        | Y               | 2     |

**Output 13. PROC SQL output with stratifications**

If you only want to view counts of mismatched values, you can add a where statement to the PROC SQL code as such:

```
title1 'Proc Sql: Counts of Reported vs. Imputed Variable Mismatches
(Stratified by DP and Sex)';
```

```
PROC SQL ;
  SELECT dp
    , sex
    , race
    , imputedrace
    , count(*) AS count
  FROM dem_l3_catvars
  WHERE race ne imputedrace
  GROUP BY dp, sex, race, imputedrace
  ORDER BY dp, sex ;

  SELECT dp
    , sex
    , hispanic
    , imputedhispanic
    , count(*) AS count
  FROM dem_l3_catvars
  WHERE hispanic ne imputedhispanic
  GROUP BY dp, sex, hispanic, imputedhispanic
  ORDER BY dp, sex ;
QUIT ;
```

Output:

**Proc Sql: Counts of Reported vs. Imputed Variable Mismatches (Stratified by Dp and Sex)**

| DP   | sex | race | imputedrace | count |
|------|-----|------|-------------|-------|
| DP01 | F   | 2    |             | 1     |
| DP01 | F   | 2    | 0           | 1     |
| DP01 | F   | 2    | 3           | 1     |
| DP01 | F   | 3    | 5           | 1     |
| DP01 | F   | 4    | 1           | 1     |
| DP01 | F   | 4    | 6           | 1     |
| DP01 | F   | 5    | 2           | 2     |
| DP01 | F   | 6    | 2           | 1     |
| DP01 | F   | 6    | 5           | 1     |
| DP01 | F   | M    | 3           | 1     |
| DP01 | M   | 0    | 1           | 1     |
| DP01 | M   | 1    | 2           | 1     |
| DP01 | M   | 2    | 5           | 1     |
| DP01 | M   | 3    | 5           | 2     |
| DP01 | M   | 3    | M           | 1     |

| DP   | sex | hispanic | imputedhispanic | count |
|------|-----|----------|-----------------|-------|
| DP01 | F   | N        | U               | 3     |
| DP01 | F   | N        | Y               | 2     |
| DP01 | F   | U        | N               | 1     |
| DP01 | F   | Y        | N               | 1     |
| DP01 | F   | Y        | U               | 1     |
| DP01 | M   | N        | Y               | 5     |
| DP01 | M   | U        | N               | 2     |
| DP01 | M   | Y        |                 | 1     |
| DP01 | M   | Y        | N               | 2     |
| DP01 | M   | Y        | U               | 2     |
| DP02 | F   | N        | U               | 2     |
| DP02 | F   | N        | Y               | 1     |
| DP02 | F   | U        |                 | 1     |
| DP02 | F   | U        | N               | 2     |

#### Output 14. PROC SQL output with mismatch analysis

Take this mismatch analysis further by producing both counts and percentages across one table.

See the code below:

```

title1 'Proc Sql: Mismatch Analysis of Reported vs. Imputed Variables
(Stratified by DP and Sex)';
PROC SQL;
    SELECT dp
           , sex
           , SUM ( CASE WHEN race = imputedrace THEN 1 ELSE 0 END ) AS
race_match
           , sum( CASE WHEN race ne imputedrace THEN 1 ELSE 0 END ) AS
race_mismatch
           , sum( CASE WHEN hispanic = imputedhispanic THEN 1 ELSE 0 END ) AS
hisp_match
           , sum( CASE WHEN hispanic ne imputedhispanic THEN 1 ELSE 0 END ) AS
hisp_mismatch
           , count(*) AS total_count
           , ( calculated race_match / calculated total_count) * 100 AS
race_match_percent
           , ( calculated race_mismatch / calculated total_count) * 100 AS
race_mismatch_percent
           , ( calculated hisp_match / calculated total_count) * 100 AS
hisp_match_percent
           , ( calculated hisp_mismatch / calculated total_count) * 100 AS
hisp_mismatch_percent
    FROM dem_l3_catvars
    GROUP BY dp, sex ;
QUIT ;

```

Here values of 1 or 0 are assigned based on whether the variable comparison is a respective match or a mismatch, summing those values, and then dividing by the total count to obtain the percentages.

The output is as follows:

| Proc Sql: Mismatch Analysis of Reported vs. Imputed Variables (Stratified by Dp and Sex) |     |            |               |            |               |             |                    |                       |                    |                       |
|------------------------------------------------------------------------------------------|-----|------------|---------------|------------|---------------|-------------|--------------------|-----------------------|--------------------|-----------------------|
| DP                                                                                       | sex | race_match | race_mismatch | hisp_match | hisp_mismatch | total_count | race_match_percent | race_mismatch_percent | hisp_match_percent | hisp_mismatch_percent |
| DP01                                                                                     | F   | 3          | 11            | 6          | 8             | 14          | 21.42857           | 78.57143              | 42.85714           | 57.14286              |
| DP01                                                                                     | M   | 4          | 12            | 4          | 12            | 16          | 25                 | 75                    | 25                 | 75                    |
| DP01                                                                                     | U   | 0          | 2             | 2          | 0             | 2           | 0                  | 100                   | 100                | 0                     |
| DP02                                                                                     | F   | 4          | 10            | 3          | 11            | 14          | 28.57143           | 71.42857              | 21.42857           | 78.57143              |
| DP02                                                                                     | M   | 5          | 10            | 4          | 11            | 15          | 33.33333           | 66.66667              | 26.66667           | 73.33333              |
| DP02                                                                                     | U   | 0          | 3             | 1          | 2             | 3           | 0                  | 100                   | 33.33333           | 66.66667              |
| DP03                                                                                     | F   | 1          | 9             | 3          | 7             | 10          | 10                 | 90                    | 30                 | 70                    |
| DP03                                                                                     | M   | 4          | 18            | 5          | 17            | 22          | 18.18182           | 81.81818              | 22.72727           | 77.27273              |
| DP03                                                                                     | U   | 0          | 4             | 1          | 3             | 4           | 0                  | 100                   | 25                 | 75                    |

**Output 15. PROC SQL output with mismatch analysis**

## HASH TABLES

When working with a large dataset, it is often important to consider efficiency, and using a hash table to compare variable disparities could make the difference. The examples here use hash tables to create summary counts of matching and mismatched values. Please note that because there are several lines of code required for the hash table comparisons, only race vs. imputedrace is displayed below. The code will be the same for the hispanic vs imputedhispanic variables so this code can easily be put in a macro to run comparisons for any reported vs imputed variables.

```

DATA _NULL_ ( DROP = rc ) ;
/*INITIALIZE HASH OBJECT*/
IF _n_ = 1 THEN DO ;
  /*DEFINE HASH TABLE FOR RACE VS IMPUTEDRACE COMPARISON*/
  DECLARE HASH race_hash() ;
  /*DEFINE KEY VARIABLES FOR COMPARISON*/
  race_hash.definekey ( 'race' , 'imputedrace' ) ;
  /*DEFINE VARIABLES TO BE STORED IN HASH TABLE, INCLUDING KEY AND COUNT
  VARIABLES*/
  race_hash.definedata( 'race' , 'imputedrace' , 'race_match_count' ,
  'race_mismatch_count' ) ;
  race_hash.definedone() ;
END ;

/*READ OBSERVATIONS FROM INPUT DATASET AND DEFINE END OF DATASET AS EOF*/
SET dem_l3_catvars END = eof;

IF race = imputedrace THEN DO ; /*CHECK FOR RACE MATCH AND COUNT MATCHES*/
  rc = race_hash.find() ;
  IF rc ne 0 THEN DO ;
    race_match_count = 0 ;
    race_mismatch_count = 0 ;
  END ;
  race_match_count + 1;
  rc = race_hash.replace();
END ;
ELSE DO ; /*CHECK FOR RACE AND COUNT MISMATCHES*/
  rc = race_hash.find() ;
  IF rc ne 0 THEN DO ;
    race_match_count = 0 ;
    race_mismatch_count = 0 ;
  END ;
  race_mismatch_count + 1 ;
  rc = race_hash.replace() ;
END;

/*OUTPUT HASH SUMMARY TABLE AT END OF DATASET*/
IF eof THEN DO ;
  race_hash.output(dataset: 'race_summary' ) ;
END ;
RUN ;

/*SORT OUTPUT DATASET FOR READABILITY*/
PROC SORT DATA = race_summary ;
  BY race imputedrace ;
RUN ;

```

The hash table code produces the dataset `race_summary`, as shown below:

|    | race | imputedrace | race_match_count | race_mismatch_count |
|----|------|-------------|------------------|---------------------|
| 1  | 0    |             | 0                | 1                   |
| 2  | 0    | 0           | 4                | 0                   |
| 3  | 0    | 1           | 0                | 2                   |
| 4  | 0    | 2           | 0                | 1                   |
| 5  | 0    | 5           | 0                | 1                   |
| 6  | 0    | 6           | 0                | 1                   |
| 7  | 1    | 0           | 0                | 1                   |
| 8  | 1    | 1           | 2                | 0                   |
| 9  | 1    | 2           | 0                | 1                   |
| 10 | 1    | 5           | 0                | 2                   |
| 11 | 1    | 6           | 0                | 2                   |
| 12 | 1    | M           | 0                | 3                   |
| 13 | 2    |             | 0                | 3                   |
| 14 | 2    | 0           | 0                | 2                   |
| 15 | 2    | 2           | 2                | 0                   |
| 16 | 2    | 3           | 0                | 4                   |
| 17 | 2    | 4           | 0                | 1                   |
| 18 | 2    | 5           | 0                | 1                   |
| 19 | 2    | M           | 0                | 3                   |
| 20 | 3    |             | 0                | 1                   |
| 21 | 3    | 0           | 0                | 2                   |
| 22 | 3    | 1           | 0                | 2                   |
| 23 | 3    | 2           | 0                | 1                   |
| 24 | 3    | 3           | 3                | 0                   |
| 25 | 3    | 5           | 0                | 3                   |
| 26 | 3    | 6           | 0                | 1                   |

#### Output 16. HASH output dataset

Take this a step further and consider stratifying these counts by the categorical variables `DP` and `sex`. Easily add these categorical variables when declaring the hash object as shown in the code below.

```

DATA _NULL_ ( DROP = rc ) ;
/*INITIALIZE HASH OBJECT*/
IF _n_ = 1 THEN DO ;
  /*DEFINE HASH TABLE FOR RACE VS IMPUTEDRACE COMPARISON*/
  DECLARE HASH race_hash() ;
  /*DEFINE KEY VARIABLES FOR STRATIFICATION AND COMPARISON*/
  race_hash.definekey ( 'dp' , 'sex' , 'race' , 'imputedrace' ) ;
  /*DEFINE VARIABLES TO BE STORED IN HASH TABLE, INCLUDING KEY AND COUNT
  VARIABLES*/
  race_hash.definedata( 'dp' , 'sex' , 'race' , 'imputedrace' ,
  'race_match_count' , 'race_mismatch_count' ) ;
  race_hash.definedone() ;
END ;

/*READ OBSERVATIONS FROM YOUR INPUT DATASET AND DEFINE END OF DATASET AS
EOF*/
SET dem_l3_catvars END = eof;

IF race = imputedrace THEN DO ; /*CHECK FOR RACE MATCH AND COUTN MATCHES*/
  rc = race_hash.find() ;
  IF rc ne 0 THEN DO ;
    race_match_count = 0 ;
    race_mismatch_count = 0 ;
  END ;
  race_match_count + 1;
  rc = race_hash.replace();
END ;
ELSE DO ; /*CHECK FOR RACE AND COUNT MISMATCHES*/
  rc = race_hash.find() ;
  IF rc ne 0 THEN DO ;
    race_match_count = 0 ;
    race_mismatch_count = 0 ;
  END ;
  race_mismatch_count + 1 ;
  rc = race_hash.replace() ;
END;

/*OUTPUT HASH SUMMARY TABLE AT END OF DATASET*/
IF eof THEN DO ;
  race_hash.output(dataset: 'race_summary_stratified' ) ;
END ;
RUN ;

/*SORT OUTPUT DATASET FOR READABILITY*/
PROC SORT DATA = race_summary_stratified ;
  BY dp sex race imputedrace ;
RUN ;

```

The only lines of code that changed were the definekey and definedata steps when declaring the hash table to include the categorical variables for stratification.

The output is shown below:

|    | DP   | sex | race | imputedrace | race_match_count | race_mismatch_count |
|----|------|-----|------|-------------|------------------|---------------------|
| 1  | DP01 | F   | 1    | 1           | 1                | 0                   |
| 2  | DP01 | F   | 2    |             | 0                | 1                   |
| 3  | DP01 | F   | 2    | 0           | 0                | 1                   |
| 4  | DP01 | F   | 2    | 3           | 0                | 1                   |
| 5  | DP01 | F   | 3    | 5           | 0                | 1                   |
| 6  | DP01 | F   | 4    | 1           | 0                | 1                   |
| 7  | DP01 | F   | 4    | 6           | 0                | 1                   |
| 8  | DP01 | F   | 5    | 2           | 0                | 2                   |
| 9  | DP01 | F   | 5    | 5           | 1                | 0                   |
| 10 | DP01 | F   | 6    | 2           | 0                | 1                   |
| 11 | DP01 | F   | 6    | 5           | 0                | 1                   |
| 12 | DP01 | F   | M    | 3           | 0                | 1                   |
| 13 | DP01 | F   | M    | M           | 1                | 0                   |
| 14 | DP01 | M   | 0    | 0           | 3                | 0                   |
| 15 | DP01 | M   | 0    | 1           | 0                | 1                   |
| 16 | DP01 | M   | 1    | 2           | 0                | 1                   |
| 17 | DP01 | M   | 2    | 5           | 0                | 1                   |
| 18 | DP01 | M   | 3    | 5           | 0                | 2                   |
| 19 | DP01 | M   | 3    | M           | 0                | 1                   |
| 20 | DP01 | M   | 4    | 0           | 0                | 1                   |
| 21 | DP01 | M   | 4    | M           | 0                | 1                   |
| 22 | DP01 | M   | 5    | 4           | 0                | 1                   |
| 23 | DP01 | M   | 6    | 0           | 0                | 1                   |
| 24 | DP01 | M   | 6    | 4           | 0                | 1                   |
| 25 | DP01 | M   | M    | 1           | 0                | 1                   |
| 26 | DP01 | M   | M    | M           | 1                | 0                   |
| 27 | DP01 | M   | 2    | 2           | 0                | 1                   |

#### Output 17. HASH output dataset with stratifications

### PERL REGULAR EXPRESSIONS

Regular expressions are typically considered when code needs to evaluate, deconstruct, or analyze text strings, but some basic regular expression functions can also be used to generate comparisons between variables. Due to how the comparisons are generated, this can be a preferred method if later steps will also need a counter variable, as a numeric variable is generated in this process that could be used for other procedures.

In this initial example, PRXPARSE and PRXMATCH are used in conjunction. Combining these two provides variables that can be used in both further selection steps as well as in summing.

PRXPARSE establishes the values that are being searched in the variable and the condition(s) for those values. PRXMATCH completes the search within the variable of interest.

```

DATA REGEX_01 ;
  SET dem_l3_catvars ( KEEP = DP HISPANIC IMPUTEDHISPANIC ) ;
/* STARTING WITH RECORDS WHERE THE VALUE OF HISPANIC IS N = NO */
  IF upcase ( HISPANIC ) in ('N') THEN DO ;
    /* PRXPARSE COMPILES THE PERL REGULAR EXPRESSION FOR USE IN SEARCH */
    /* SETTING UP non-NO VALUES - Y = YES, U = UNKNOWN, AND BLANK */
    /* THE i OPTION MAKES SEARCH CASE INSENSITIVE */
    patternID = prxparse ('/Y|U| /i') ;
    /* PRXMATCH FINDS POSITION OF PATTERN MATCH IN COMPARED VARIABLE */
    mismatch = prxmatch ( patternID , IMPUTEDHISPANIC ) ;
  END ;
/* REPEAT TO CREATE COMPARISON FOR OTHER VALUES OF VARIABLES */
  ELSE IF upcase ( HISPANIC ) in ('Y') THEN DO ;
    patternID = prxparse ('/N|U| /i') ;
    mismatch = prxmatch ( patternID , IMPUTEDHISPANIC ) ;
  END ;
  ELSE IF upcase ( HISPANIC ) in ('U') THEN DO ;
    patternID = prxparse ('/Y|N| /i') ;
    mismatch = prxmatch ( patternID , IMPUTEDHISPANIC ) ;
  END ;
RUN ;

```

Here is a sample of the output dataset REGEX\_01:

|    | DP   | hispanic | imputedhispanic | patternID | mismatch |
|----|------|----------|-----------------|-----------|----------|
| 1  | DP03 | Y        | N               | 1         | 1        |
| 2  | DP03 | N        | N               | 2         | 0        |
| 3  | DP01 | N        | Y               | 2         | 1        |
| 4  | DP02 | N        | U               | 2         | 1        |
| 5  | DP03 | N        | N               | 2         | 0        |
| 6  | DP03 | U        | Y               | 3         | 1        |
| 7  | DP01 | Y        | Y               | 1         | 0        |
| 8  | DP03 | U        | Y               | 3         | 1        |
| 9  | DP03 | N        | N               | 2         | 0        |
| 10 | DP02 | Y        |                 | 1         | 1        |
| 11 | DP02 | N        | U               | 2         | 1        |
| 12 | DP01 | N        | N               | 2         | 0        |
| 13 | DP01 | N        | N               | 2         | 0        |
| 14 | DP02 | Y        | N               | 1         | 1        |
| 15 | DP02 | N        | N               | 2         | 0        |
| 16 | DP01 | U        | N               | 3         | 1        |
| 17 | DP03 | U        | U               | 3         | 0        |
| 18 | DP01 | Y        | N               | 1         | 1        |
| 19 | DP02 | N        | Y               | 2         | 1        |
| 20 | DP01 | Y        | U               | 1         | 1        |
| 21 | DP03 | U        | Y               | 3         | 1        |

**Output 18. PERL regular expression output dataset**

The PRXPARSE function yields the values in patternID. This enumerates the combinations or “patterns” of search that were in the data. The PRXMATCH function yields the values in mismatch. This provides a numeric value indicating the position in the string of the parsed values with the number of that position or a non-match to the parsed values with a 0. For this comparison between a variable and its imputed version, the parsed search was based on looking for mismatches, so for this example the records where mismatch = 1 indicate the values are not the same; if the strings were longer, the numbers could have varied more.

A sorted, deduplicated version of this dataset can make it easier to determine if the results of PRXPARSE could aid in further subsetting or crosstabulating the data based on the combinations of values in patternID. Here is one method to create that dataset using PROC SORT followed by its output:

```
/* SHOW FINAL COMBINATIONS OF VALUES TO DETERMINE HOW TO BEST */
/* STRUCTURE SELECTION OF RECORDS FOR COUNTS */
PROC SORT DATA = REGEX_01 OUT = REGEX_01_DISTINCT NODUPKEY ;
  BY _ALL_ ;
RUN ;
```

|    | DP   | hispanic | imputedhispanic | patternID | mismatch |
|----|------|----------|-----------------|-----------|----------|
| 1  | DP01 | N        | N               | 2         | 0        |
| 2  | DP01 | N        | U               | 2         | 1        |
| 3  | DP01 | N        | Y               | 2         | 1        |
| 4  | DP01 | U        | N               | 3         | 1        |
| 5  | DP01 | U        | U               | 3         | 0        |
| 6  | DP01 | Y        |                 | 1         | 1        |
| 7  | DP01 | Y        | N               | 1         | 1        |
| 8  | DP01 | Y        | U               | 1         | 1        |
| 9  | DP01 | Y        | Y               | 1         | 0        |
| 10 | DP02 | N        | N               | 2         | 0        |
| 11 | DP02 | N        | U               | 2         | 1        |
| 12 | DP02 | N        | Y               | 2         | 1        |
| 13 | DP02 | U        |                 | 3         | 1        |
| 14 | DP02 | U        | N               | 3         | 1        |
| 15 | DP02 | U        | Y               | 3         | 1        |
| 16 | DP02 | Y        |                 | 1         | 1        |
| 17 | DP02 | Y        | N               | 1         | 1        |
| 18 | DP02 | Y        | U               | 1         | 1        |
| 19 | DP02 | Y        | Y               | 1         | 0        |
| 20 | DP03 | N        |                 | 2         | 1        |
| 21 | DP03 | N        | N               | 2         | 0        |
| 22 | DP03 | N        | U               | 2         | 1        |
| 23 | DP03 | N        | Y               | 2         | 1        |
| 24 | DP03 | U        |                 | 3         | 1        |

#### Output 19. De-duplicated PERL regular expression output dataset

In this case, HISPANIC = N links to patternID = 1, HISPANIC = U links to patternID = 2, and HISPANIC = Y links to patternID = 3. Depending on what type of analysis is being done after the comparison, these values could be used in conjunction with the match or non-match (indicated in this case by mismatch), but it is important to see how the patterns were assigned, and a de-duplicated dataset makes that easy to visualize.

If the only relevant information is the question of a match or non-match, this can be accomplished using only PRXMATCH.

```
DATA REGEX_02 ;
  SET dem_l3_catvars ( KEEP = DP HISPANIC IMPUTEDHISPANIC ) ;
  /* IF DISTINCT IDENTIFICATION OF THE PATTERN IS NOT NEEDED */
  /* THE SEARCH CAN BE DONE USING ONLY PRXMATCH */
  IF upcase ( HISPANIC ) in ('N') THEN DO ;
    mismatch = prxmatch ( '/Y|U| /i' , IMPUTEDHISPANIC ) ;
  END ;
  /* REPEAT TO CREATE COMPARISON FOR OTHER VALUES OF VARIABLE */
  ELSE IF upcase ( HISPANIC ) in ('Y') THEN DO ;
    mismatch = prxmatch ( '/N|U| /i' , IMPUTEDHISPANIC ) ;
  END ;
RUN ;
```

Note that this output has the same values as was generated in REGEX\_01 but without the inclusion of position.

|    | DP   | hispanic | imputedhispanic | mismatch |
|----|------|----------|-----------------|----------|
| 1  | DP03 | Y        | N               | 1        |
| 2  | DP03 | N        | N               | 0        |
| 3  | DP01 | N        | Y               | 1        |
| 4  | DP02 | N        | U               | 1        |
| 5  | DP03 | N        | N               | 0        |
| 6  | DP03 | U        | Y               | .        |
| 7  | DP01 | Y        | Y               | 0        |
| 8  | DP03 | U        | Y               | .        |
| 9  | DP03 | N        | N               | 0        |
| 10 | DP02 | Y        |                 | 1        |
| 11 | DP02 | N        | U               | 1        |
| 12 | DP01 | N        | N               | 0        |
| 13 | DP01 | N        | N               | 0        |
| 14 | DP02 | Y        | N               | 1        |
| 15 | DP02 | N        | N               | 0        |
| 16 | DP01 | U        | N               | .        |
| 17 | DP03 | U        | U               | .        |
| 18 | DP01 | Y        | N               | 1        |
| 19 | DP02 | N        | Y               | 1        |
| 20 | DP01 | Y        | U               | 1        |
| 21 | DP03 | U        | Y               | .        |
| 22 | DP03 | Y        | N               | 1        |
| 23 | DP03 | N        | U               | 1        |

#### Output 20. PERL regular expression output dataset using only PRXMATCH

Particularly if dealing with large data or restrictions on available memory, try to be conservative in the construction of regular expressions as complex ones can become resource intensive. However, the use of regular expressions is often more efficient than longer or repeated procedures.

## PROC FREQ: ADDITIONAL APPLICATIONS

To further analyze the PROC COMPARE output, we can perform use PROC FREQ for counts of matches and mismatches. See the following code and output below for the race variable.

This evaluates the PROC COMPARE output (shown in Output 4) for records where `_type_ = 'DIF': '.'` values match, and 'X' values are mismatches. The code can substitute the hispanic variable for race to produce a mismatch analysis for Hispanic/Imputed Hispanic.

```
title1 'Mismatch Analysis of Proc Compare Output for Race/Imputed Race
(stratified by DP and Sex)';
```

```
PROC FREQ DATA=comp_strat ( where = ( _type_ = 'DIF' ) ) ;
  TABLES _type_ * dp * sex * race / NOROW NOCOL NOPERCENT ;
RUN ;
```

### Mismatch Analysis of Proc Compare Output for Race/Imputed Race (stratified by DP and Sex)

The FREQ Procedure

| Frequency | Table 1 of sex by race             |    |       |
|-----------|------------------------------------|----|-------|
|           | Controlling for _TYPE_=DIF DP=DP01 |    |       |
| sex       | race                               |    |       |
|           | .                                  | X  | Total |
| F         | 3                                  | 11 | 14    |
| M         | 4                                  | 12 | 16    |
| U         | 0                                  | 2  | 2     |
| Total     | 7                                  | 25 | 32    |

| Frequency | Table 2 of sex by race             |    |       |
|-----------|------------------------------------|----|-------|
|           | Controlling for _TYPE_=DIF DP=DP02 |    |       |
| sex       | race                               |    |       |
|           | .                                  | X  | Total |
| F         | 4                                  | 10 | 14    |
| M         | 5                                  | 10 | 15    |
| U         | 0                                  | 3  | 3     |
| Total     | 9                                  | 23 | 32    |

| Frequency | Table 3 of sex by race             |    |       |
|-----------|------------------------------------|----|-------|
|           | Controlling for _TYPE_=DIF DP=DP03 |    |       |
| sex       | race                               |    |       |
|           | .                                  | X  | Total |
| F         | 1                                  | 9  | 10    |
| M         | 4                                  | 18 | 22    |
| U         | 0                                  | 4  | 4     |
| Total     | 5                                  | 31 | 36    |

### Output 21. PROC COMPARE output analyzed with PROC FREQ

Hash table output can be further analyzed with PROC FREQ to produce frequencies of the matches vs mismatches:

```

title1 'Frequencies of Race Matches';

PROC FREQ DATA = race_summary_stratified ;
  TABLES race_match_count / NOROW NOCOL NOPERCENT ;
  BY dp sex;
RUN ;

title1 'Frequencies of Race Mismatches';

PROC FREQ DATA = race_summary_stratified ;
  TABLES race_mismatch_count / NOROW NOCOL NOPERCENT ;
  BY dp sex;
RUN ;

```

### Frequencies of Race Matches

The FREQ Procedure

DP=DP01 sex=F

| race_match_count | Frequency | Cumulative Frequency |
|------------------|-----------|----------------------|
| 0                | 10        | 10                   |
| 1                | 3         | 13                   |

### Frequencies of Race Mismatches

The FREQ Procedure

DP=DP01 sex=F

| race_mismatch_count | Frequency | Cumulative Frequency |
|---------------------|-----------|----------------------|
| 0                   | 3         | 3                    |
| 1                   | 9         | 12                   |
| 2                   | 1         | 13                   |

DP=DP01 sex=M

| race_match_count | Frequency | Cumulative Frequency |
|------------------|-----------|----------------------|
| 0                | 11        | 11                   |
| 1                | 1         | 12                   |
| 3                | 1         | 13                   |

DP=DP01 sex=M

| race_mismatch_count | Frequency | Cumulative Frequency |
|---------------------|-----------|----------------------|
| 0                   | 2         | 2                    |
| 1                   | 10        | 12                   |
| 2                   | 1         | 13                   |

DP=DP01 sex=U

| race_match_count | Frequency | Cumulative Frequency |
|------------------|-----------|----------------------|
| 0                | 2         | 2                    |

DP=DP01 sex=U

| race_mismatch_count | Frequency | Cumulative Frequency |
|---------------------|-----------|----------------------|
| 1                   | 2         | 2                    |

## Output 22. HASH output analyzed with PROC FREQ

Additionally, PROC FREQ can produce crosstabulations of matches vs mismatches using the hash table output dataset (shown in Output 17):

```

title1 'Crosstabulation of Race Matches';

PROC FREQ DATA = race_summary_stratified ;
  TABLES dp*sex*race_match_count / NOROW NOCOL NOPERCENT ;
RUN ;

title1 'Frequencies of Race Mismatches';

PROC FREQ DATA = race_summary_stratified ;
  TABLES dp*sex*race_mismatch_count / NOROW NOCOL NOPERCENT ;
RUN ;

```

| Crosstabulation of Race Matches |                                    |   |   |   |       |
|---------------------------------|------------------------------------|---|---|---|-------|
| The FREQ Procedure              |                                    |   |   |   |       |
| Frequency                       | Table 1 of sex by race_match_count |   |   |   |       |
|                                 | Controlling for DP=DP01            |   |   |   |       |
|                                 | race_match_count                   |   |   |   |       |
| sex                             | 0                                  | 1 | 2 | 3 | Total |
| F                               | 10                                 | 3 | 0 | 0 | 13    |
| M                               | 11                                 | 1 | 0 | 1 | 13    |
| U                               | 2                                  | 0 | 0 | 0 | 2     |
| Total                           | 23                                 | 4 | 0 | 1 | 28    |

| Crosstabulation of Race Mismatches |                                       |    |   |   |       |
|------------------------------------|---------------------------------------|----|---|---|-------|
| The FREQ Procedure                 |                                       |    |   |   |       |
| Frequency                          | Table 1 of sex by race_mismatch_count |    |   |   |       |
|                                    | Controlling for DP=DP01               |    |   |   |       |
|                                    | race_mismatch_count                   |    |   |   |       |
| sex                                | 0                                     | 1  | 2 | 3 | Total |
| F                                  | 3                                     | 9  | 1 | 0 | 13    |
| M                                  | 2                                     | 10 | 1 | 0 | 13    |
| U                                  | 0                                     | 2  | 0 | 0 | 2     |
| Total                              | 5                                     | 21 | 2 | 0 | 28    |

Output 23. HASH output analyzed with PROC FREQ on matches and mismatches

## RESULTS

The focus of these methods is on data that is not inherently numeric, even if it presents as numbers. By utilizing data in a character format, more procedures can be considered, and more flexibility is possible in the comparative analysis. These tools can help you to understand the quality of your data before research with it begins, can be a crucial part of analytical work, and can even provide post-analytical summaries. At any of these stages, decisions can be made to select or subset records based on the patterns or trends that are found.

An initial exploration of the data is often helpful, and starting with PROC FREQ or PROC SQL would be productive. Both offer an initial code block that is short to get started but can also be extensively modified or customized with available options. This provides easy manipulation of the data to help examine initial trends and decide what details merit further exploration.

Both PROC COMPARE and PROC MEANS can add additional levels of details and create more nuanced comparisons. Consider starting with them if the data is already familiar or if specifications already tightly define the types of comparisons that are needed.

If the data requires reworking, then it is worth stepping directly into HASH tables or implementing the use of regular expressions to derive needed components and make comparisons. Since either of these methods can then easily be combined with others, look at what the end result needs to be and build the series of procedures to meet that result.

Should all else fail, default to PROC FREQ. It is a true workhorse in this context.

## CONCLUSION

This paper explores various methods that can compare reported versus imputed values using SAS. As demonstrated, there is no one preferred method, no “one proc to rule them all”, as it depends on what is most important to consider in the analysis, whether that is calculating counts, mismatches, percentages, or considering efficiency while performing these analyses. The choice is driven by the nature of the data and the specific research questions to answer, so this paper can serve as a guide to help make those choices clearer.

## REFERENCES

- Correll, R. 2024. “Health Disparities: What They Mean and Examples.” Access May 2024. <https://www.verywellhealth.com/health-disparities-4173220>.
- Coyle, D. T. “Sentinel System Overview.” Food and Drug Administration. <https://www.fda.gov/files/about%20fda/published/Sentinel-System-Overview---Presentation.pdf>.
- Eicheldinger, C. and Bonito, A. 2008. “More Accurate Racial and Ethnic Codes for Medicare Administrative Data.” *Health Care Financial Review*, 29(3):27-42. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4195038/>.
- Elliott, M. N., Fremont, A., Morrison, P. A., Pantoja, P., and Lurie, N. 2008. “A New Method for Estimating Race/Ethnicity and Associated Disparities Where Administrative Records Lack Self-Reported Race/Ethnicity.” *Health Services Research*, 43(5 Pt 1): 1722-1736. doi: 10.1111/j.1475-6773.2008.00854.x
- Food and Drug Administration. 2024. “FDA’s Sentinel Initiative.” Accessed May 2024. <https://www.fda.gov/safety/fdas-sentinel-initiative>.
- Fremont, A., Weissman, J. S., Hoch, E., and Elliott, M. N. 2016. “When Race/Ethnicity Data Are Lacking: Using Advanced Indirect Estimation Methods to Measure Disparities.” RAND. [https://www.rand.org/pubs/research\\_reports/RR1162.html](https://www.rand.org/pubs/research_reports/RR1162.html).
- Imai, K., Olivella, S., and Rosenman, E. T. R. 2022. “Addressing census data problems in race imputation via fully Bayesian Improved Surname Geocoding and name supplements.” *Science Advances*, 8:49. doi: 10.1126/sciadv.adc9824.
- Katam, J. 2019. “Understanding regular expression and its application in SAS using PRX (PERL regEx) functions.” *Proceedings of PhUSE*. <https://www.lexjansen.com/phuse/2019/ct/CT12.pdf>. PhUSE.
- Oreshko, V. 2018. “Increase Defensiveness of Your Code: Regular Expressions.” *Proceedings of PhUSE*. <https://www.lexjansen.com/phuse/2018/ct/CT12.pdf>. PhUSE.
- Popovic, J. R. 2015. “Distributed data networks: A paradigm shift in data sharing and healthcare analytics” *Proceedings of PharmaSUG*. <https://www.pharmasug.org/proceedings/2015/HA/PharmaSUG-2015-HA07.pdf>. PharmaSUG.
- Popovic, J. R. 2017. “Healthcare data sharing and innovative analytic development: Lessons learned from distributed data networks.” *Proceedings of SGF*. <https://support.sas.com/resources/papers/proceedings17/0840-2017.pdf>. SGF.
- Popovic, J. R. 2019. “Real-world data as real-world evidence: Establishing the meaning of data as a prerequisite to determining secondary-use value.” *Proceedings of PharmaSUG*. <https://www.lexjansen.com/pharmasug/2019/RW/PharmaSUG-2019-RW-310.pdf>. PharmaSUG.
- RAND. 2024. “RAND Bayesian Improved Surname Geocoding” Accessed May 2024. <https://www.rand.org/health-care/tools-methods/bisg.html#:~:text=Bayesian%20Improved%20Surname%20and%20Geocoding,and%20illuminate%20areas%20for%20improvement>.

SAS. “Perl Regular Expressions Tip Sheet.” Accessed May 2024.

<https://support.sas.com/content/dam/SAS/support/en/products-solutions/base-sas/tip-sheets/regexp-tip-sheet.pdf>.

SAS. PRXMATCH Function.” Accessed May 2024.

[https://documentation.sas.com/doc/en/pgmsascdc/9.4\\_3.5/lefunctionsref/n0bj9p4401w3n9n1gmv6tfshit9m.htm](https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/lefunctionsref/n0bj9p4401w3n9n1gmv6tfshit9m.htm).

SAS. “PRXPARE Function.” Accessed May 2024.

[https://documentation.sas.com/doc/en/pgmsascdc/9.4\\_3.5/lefunctionsref/p06i7305izsnvcn1ru9147suzyv5.htm](https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/lefunctionsref/p06i7305izsnvcn1ru9147suzyv5.htm).

Slaughter, M. T., Nyongesa, D. B., Dickerson, J. F., and Kuntz, J. L. 2023. “Real World Evidence in Distributed Data Networks: Lessons from a Post-Marketing Safety Study.” *Proceedings of PharmaSUG*. <https://www.lexjansen.com/pharmasug/2023/RW/PharmaSUG-2023-RW-050.pdf>. PharmaSUG.

Sorbero, M. E., Euler, R., Kofner, A., and Elliott, M. N. “Imputation of Race and Ethnicity in Health Insurance Marketplace Enrollment Data, 2015-2022 Open Enrollment Periods.” HHS. <https://aspe.hhs.gov/sites/default/files/documents/931e1e3edec745ea7fdd364f9e28d6c6/aspe-rand-imputation-race-ethnicity-mktplc-rpt.pdf>.

## ACKNOWLEDGMENTS

We would like to thank Daniel Scarnecchia for the original inspiration behind this paper, Lauren Zichittella for her support and feedback, David Cole for his feedback and input, and the Harvard Pilgrim Health Care Institute for supporting our writing and conference attendance.

## CONTACT INFORMATION

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

Neesha Nathwani  
Sarah Woodruff  
Harvard Pilgrim Health Care Institute, Department of Population Medicine  
401 Park Drive, Suite 401, Boston, MA 02215  
neesha\_nathwani@hphci.harvard.edu  
sarah\_woodruff@hphci.harvard.edu

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.

## APPENDIX

Below find the code which generated the fabricated source data set used in this paper:

```
DATA dem_l3_catvars ;
  /* SET THE NUMBER OF OBSERVATIONS DESIRED */
  DO i = 1 to 100; /* ADJUST NUMBER OF ROWS AS NEEDED */
    /* ASSIGN VALUES FOR DP */
    SELECT (floor(ranuni(123) * 3) + 1);
      WHEN (1) DP = 'DP01' ;
      WHEN (2) DP = 'DP02' ;
      WHEN (3) DP = 'DP03' ;
    END ;

    /* GENERATE RANDOM COUNT VALUE; RANDOM NUMBERS 10 AND GREATER */
    count = floor(10 + ranuni(123) * 10000) ;
    /* ASSIGN VALUES FOR HISPANIC */
    IF ranuni(123) < 0.33 THEN hispanic = 'Y' ;
    ELSE IF ranuni(123) < 0.66 THEN hispanic = 'N' ;
    ELSE hispanic = 'U' ;
    /* ASSIGN VALUES FOR IMPUTEDHISPANIC; 10% CHANCE OF BEING NULL */
    IF ranuni(123) < 0.1 THEN imputedhispanic = '' ;
    ELSE IF ranuni(123) < 0.33 THEN imputedhispanic = 'Y' ;
    ELSE IF ranuni(123) < 0.66 THEN imputedhispanic = 'N' ;
    ELSE imputedhispanic = 'U' ;
    /* ASSIGN VALUES FOR RACE; GENERATES VALUES 0 TO 6 */
    race = put(floor(ranuni(123) * 7), 1.) ;
    IF ranuni(123) < 0.1 THEN race = 'M' ; /* 10% CHANCE OF BEING 'M' */
    /* ASSIGN VALUES FOR IMPUTEDRACE; 10% CHANCE OF BEING NULL */
    IF ranuni(123) < 0.1 THEN imputedrace = '' ;
    ELSE imputedrace = put(floor(ranuni(123) * 7), 1.) ;
    /* GENERATES VALUES 0 TO 6; 10% CHANCE OF BEING 'M' */
    if ranuni(123) < 0.1 then imputedrace = 'M' ;
    /* ASSIGN VALUES FOR SEX */
    IF ranuni(123) < 0.5 THEN sex = 'M' ;
    ELSE if ranuni(123) < 0.8 THEN sex = 'F' ;
    ELSE sex = 'U' ;
    OUTPUT ; /* WRITE RECORD TO DATASET */
  END ;
  DROP i ; /* DROP LOOP INDEX VARIABLE */
RUN ;
```