Analysis Specifications

Version Number:	Version 1.0
Analysis Name:	Demographic Analysis Panel
Engine Type / Engine Version:	SAS 9.2 Excel
Analysis Overview:	The Demographic Analysis Panel provides the user with an overview of the number and percent of subjects that correspond to many of the variables in the Demographics dataset. It looks at the frequency of variables such as Race, Ethnicity, Sex, Country, Site and a derived variable for Age Groups, as well as descriptive statistics of the variable Age. This is done by total study population, treatment arm and disposition categories.
Analysis Description:	The Demographics Analysis Panel contains the following 15 analyses that are outlined below. For the demographic variables, the denominator for percent calculations is automatically defined by the DM dataset using the actual arm (ACTARM) or planned treatment arm variable (ARM), excluding cases labeled as Screen Failures or Not Assigned. For the demographic variables by disposition categories, the denominator for percent calculations is the demographic variable total by arm and overall study population. **Analysis 1: Demographics Overview** 1. Provides a table that combines the counts and percents for Race, Ethnicity, Sex and the derived variable Age Groups and the Mean, Median, Standard Deviation, Minimum and Maximum for Age. **Analysis 2&3: Age Group Counts and Percents** 1. Provides a table of the counts and percents for the derived variable Age Groups for overall treatment population and by treatment arm. 2. Provides a chart of the counts and percents for the derived variable Age Groups for overall treatment population and by treatment arm. 3. Provides a table of the counts and percents for the derived variable Age Groups by disposition categories for overall treatment population and by treatment arm. **Analysis 4: Age Descriptive Statistics** 1. Provides a table of the descriptive statistics Mean, Median, Mode, Standard Deviation, Quartile 1, Quartile 3, Minimum and Maximum for the Age variable for overall treatment population and by treatment arm. 2. Provides a chart of the descriptive statistics Mean, Median, Mode, Standard Deviation, Quartile 1, Quartile 3, Minimum and Maximum for the Age variable for overall treatment population and by treatment arm. **Analysis 5&6: Sex Counts and Percents** 1. Provides a table of the counts and percents for the Sex variable categories submitted by the sponsor for overall treatment population and by treatment arm. 2. Provides a chart of the counts and percents for the Sex variable categories

	submitted by the sponsor for overall treatment population and by treatment arm. 3. Provides a table of the counts and percents for the Sex variable categories submitted by the sponsor by disposition categories for overall treatment population and by treatment arm. **Analysis 7&8: Race Counts and Percents** 1. Provides a table of the counts and percents for the Race variable categories submitted by the sponsor for overall treatment population and by treatment arm. 2. Provides a chart of the counts and percents for the Race variable categories submitted by the sponsor for overall treatment population and by treatment arm. 3. Provides a table of the counts and percents for the Race variable categories submitted by the sponsor by disposition categories for overall treatment population and by treatment arm. **Analysis 9&10: Ethnicity Counts and Percents** 1. Provides a table of the counts and percents for the Ethnicity variable categories submitted by the sponsor for overall treatment population and by treatment arm. 2. Provides a chart of the counts and percents for the Ethnicity variable categories submitted by the sponsor for overall treatment population and by treatment arm. 3. Provides a table of the counts and percents for the Ethnicity variable categories submitted by the sponsor by disposition categories for overall treatment population and by treatment arm. **Analysis 11&12: Country Counts and Percents** 1. Provides a table of the counts and percents for the Country variable categories submitted by the sponsor for overall treatment population and by treatment arm. 2. Provides a table of the counts and percents for the Country variable categories submitted by the sponsor by disposition categories for overall treatment population and by treatment arm. **Analysis 13&14: Site ID Counts and Percents** 1. Provides a table of the counts and percents for the SiteID variable categories submitted by the sponsor for overall treatment population and by treatment arm. 2. Provides a table of the counts and percents for the SiteID variable categories submitted by the sponsor by disposition categories for overall treatment population and by treatment arm. **Analysis 15: Country and Site ID Counts and Percents** 1. Provides a table of the counts and percents for the Country and SiteID variable categories submitted by the sponsor for overall treatment population and by treatment arm.
Precondition(s):	The Demographic Analysis Panel has the following preconditions: 1. A Demographic dataset (DM) has to exist according to CDISC SDTM format specifications. 1.1. The Demographic variables ACTARM or ARM and USUBJID must be

<table>
<tr><td></td><td>present in the DM dataset
1.2. ARMCD, AGEU, AGE, RACE, ETHNIC, SEX, COUNTRY, and SITEID will be used if present but are not necessary

2. A Disposition dataset has to exist according to CDISC format specifications

2.1 The Disposition variables USUBJID and DSDECOD must be present in the DS dataset.</td></tr>
<tr><td>Panel Options:</td><td>The following is a list of topics and values that the user is required to enter from the panel options page of the Script Launcher. The value chosen will be used in the analysis:

<table>
<tr><th>Topic</th><th>Possible Values</th><th>Default</th></tr>
<tr><td>Age Units</td><td>Hours
Days
Weeks
Months
Years</td><td>Years</td></tr>
<tr><td>Age Group 1-5</td><td>Ability to group ages into buckets</td><td>Group1: 0 <= Age < 65
Group2: Age >= 65</td></tr>
</table>
</td></tr>
<tr><td>Input(s):</td><td>The Demographic Analysis Panel uses the following data and variables:

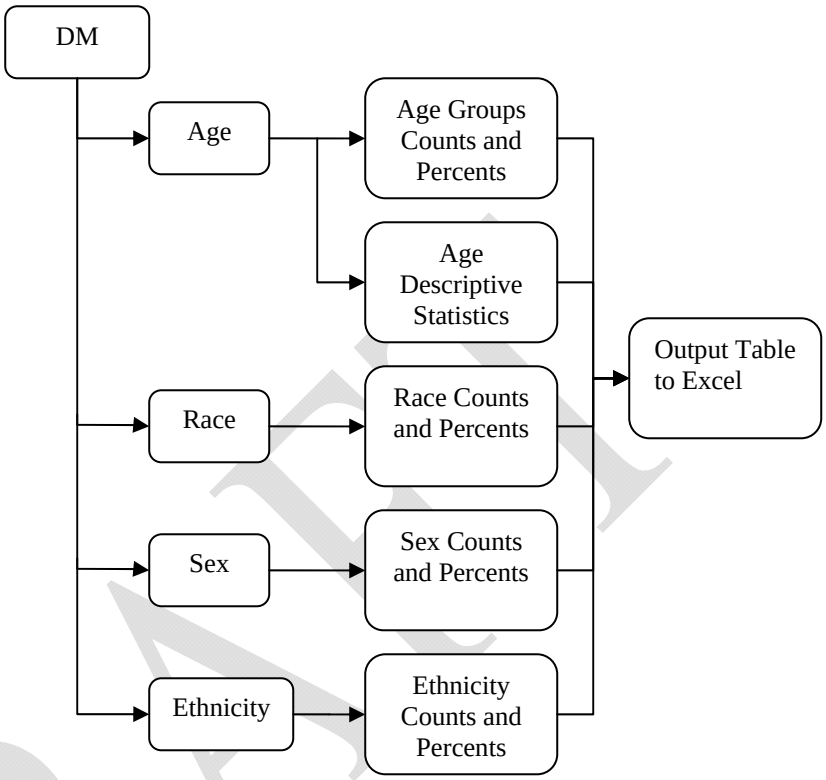
1. Dataset DM –
1.1. ACTARM or ARM
1.1.1. ACTARM to be used in preference to ARM if available
1.2. ARMCD (optional)
1.3. AGE (optional)
1.4. AGEU (optional)
1.5. RACE (optional)
1.6. ETHNIC (optional)
1.7. SEX (optional)
1.8. COUNTRY (optional)
1.9. SITEID (optional)
1.10. USUBJID

2. Dataset DS –
2.1. DSDECOD
2.2. USUBJID</td></tr>
<tr><td>Input from:</td><td>There are two different types of inputs for this analysis panel:

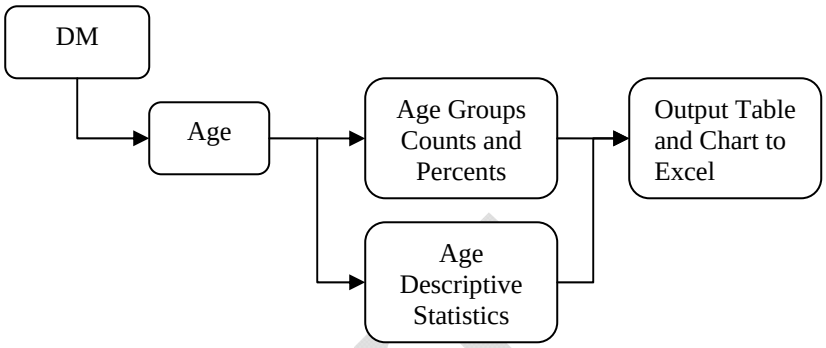
1) The CDISC SDTM datasets required to perform the analysis are chosen by the user and provided by Script Launcher
2) The panel options discussed above</td></tr>
</table>

Process Flow:

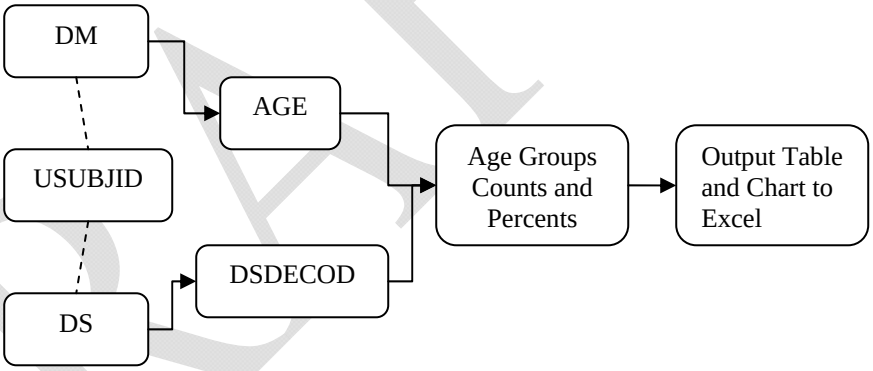
Analysis 1: Demographic Overview



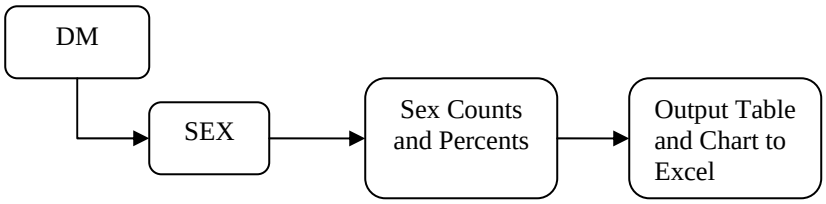
Analysis 2 and 3: Age Groups Counts and Percents and Age Descriptive Statistics



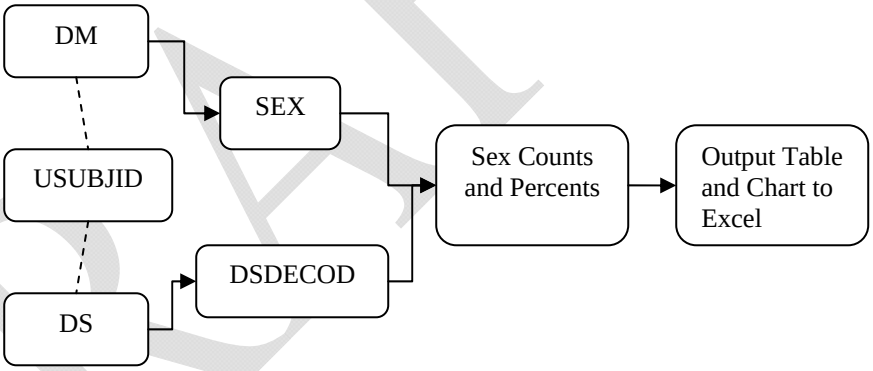
Analysis 4: Age Groups Counts and Percents by Disposition



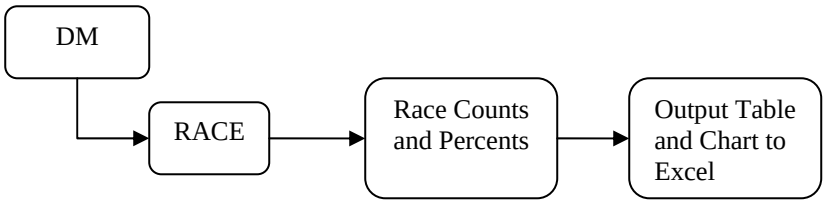
Analysis 5: Sex Counts and Percents



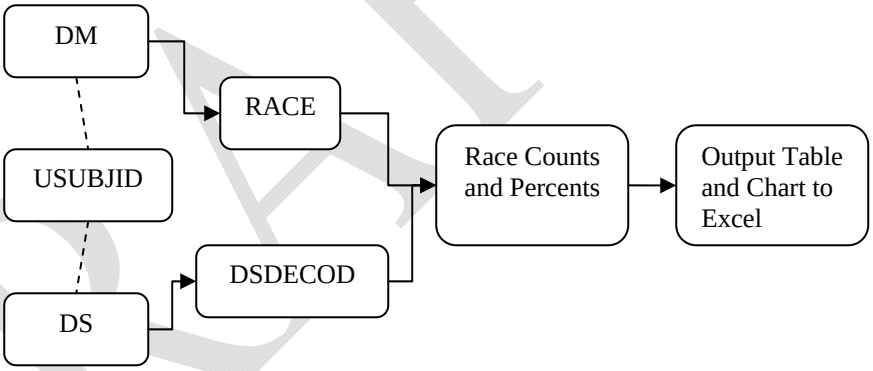
Analysis 6: Sex Counts and Percents by Disposition



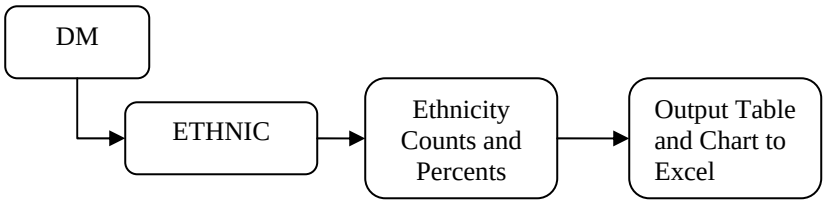
Analysis 7: Race Counts and Percents



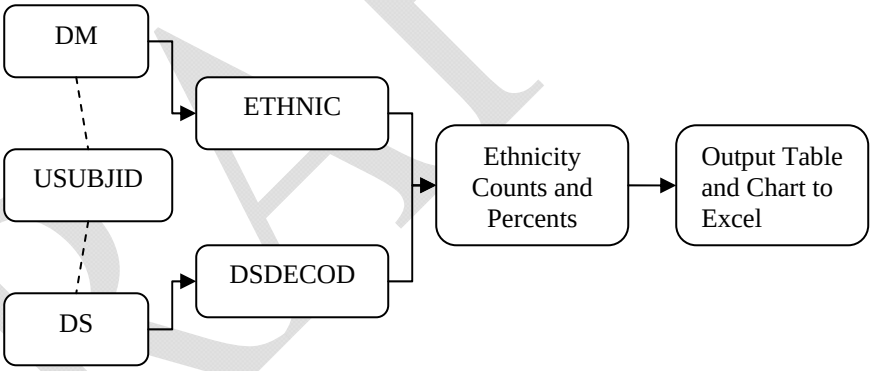
Analysis 8: Race Counts and Percents by Disposition



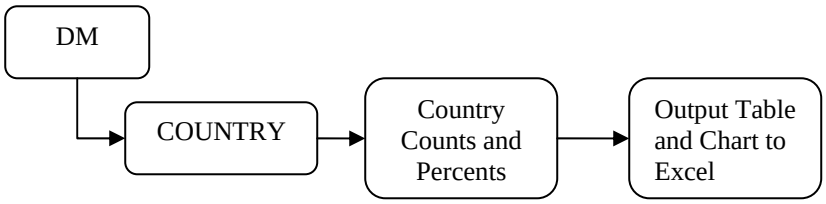
Analysis 9: Ethnicity Counts and Percents



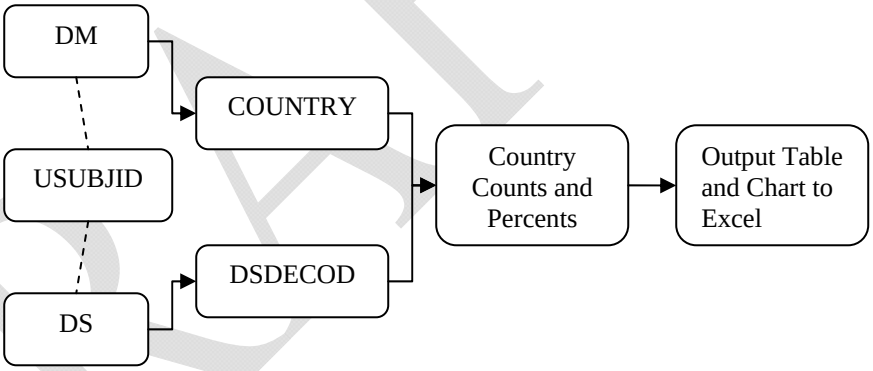
Analysis 10: Ethnicity Counts and Percents by Disposition



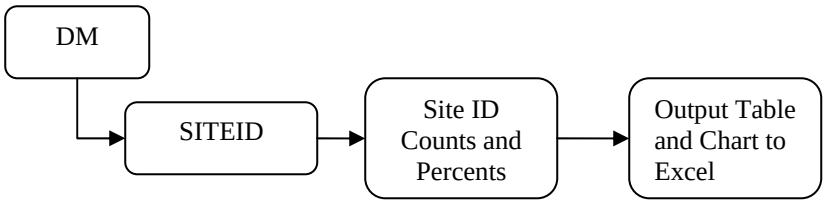
Analysis 11: Country Counts and Percents



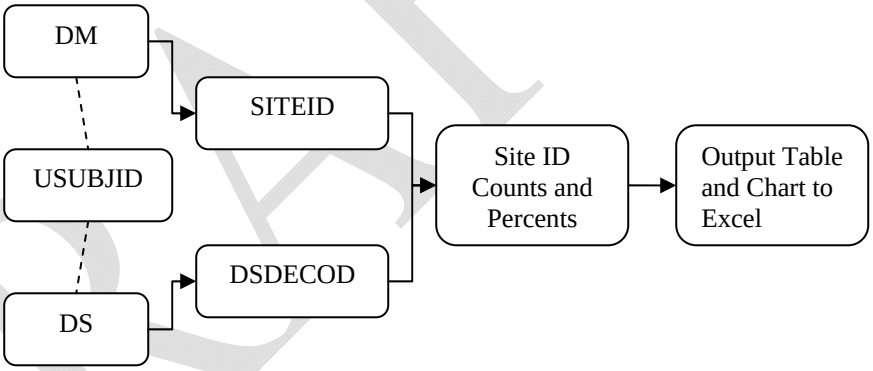
Analysis 12: Country Counts and Percents by Disposition



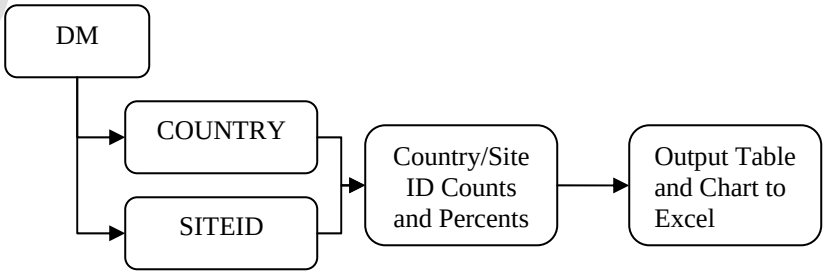
Analysis 13: Site ID Counts and Percents



Analysis 14: Site ID Counts and Percents by Disposition



Analysis 15: Country and Site ID Counts and Percents



Study Design:	The Demographic Analysis Panel will run on a study with up to six Treatment Arms in the study design. This analysis can be run on studies that are parallel study designs.
Requirements:	The Demographic Analysis Panel shall perform and execute the following steps within the prescribed order: 1. Identify the Demographic and Disposition datasets. 1.1 Determine whether the Demographic and Disposition datasets are in the prescribed CDISC SDTM format and whether the variables within the dataset comply with CDISC SDTM standards. 1.1.1. Demographics (DM) 1.1.2. Disposition (DS) 1.2 Determine whether the needed Demographic variables are present within the Demographics dataset. 1.2.1. ACTARM or ARM 1.2.2. ARMCD 1.2.3. AGE 1.2.4. AGEU 1.2.5. RACE 1.2.6. ETHNIC 1.2.7. SEX 1.2.8. COUNTRY 1.2.9. SITEID 1.2.10. USUBJID 1.3 Determine whether the needed Disposition variables are present in the Disposition dataset. 1.3.1. DSDECOD 1.3.2. USUBJID 1.4 If the Demographic and Disposition datasets and variables are CDISC SDTM compliant and the variables ACTARM or ARM and USUBJID from DM and DSDECOD and USUBJID from DS are all present then continue 1.4.1. If any of the AGE, RACE, ETHNIC, SEX, COUNTRY, or SITEID variables are not included in the datasets, the program will continue to run but those variables' results will be made entirely from missing values 1.5 Reduce the Disposition dataset to the last disposition event (DSDECOD) for each subject (USUBJID), disposition category (DSCAT), and disposition subcategory (DSSCAT) where the category is not 'Protocol Milestone' and the disposition term is not 'Informed Consent Obtained' or 'Randomized'. Further reduce these records to one disposition term (DSDECOD) per subject in the case that a term is used in more than category. This is done to avoid percents greater than 100 in the demographics by disposition analyses. The resulting percents in those analyses are thus the percent of each arm whose last disposition event was the reported term. **Analysis 1: Demographics Overview** 1. Collect the per arm and overall counts and percents found by analyses 2, 5, 7, and 9, and the descriptive statistics per arm and overall found by analysis 4 and place them

in a table, giving an overview of the study's demographics on a single tab

2. Each arm in the study will get a column, and at the top of each column will be a row containing the arm name (ACTARM or ARM, henceforth referred to as arm) and the subject count for the arm. After all arms, there will be an overall column, headed by 'Overall' and followed by the total study subject count
3. Under the column header, the table is divided into two sections
    - 3.1 The first section contains the descriptive statistics for age by arm and overall
    - 3.2 The second section contains the counts and percents for AGE_GROUP (where AGE_GROUP is specified by the user in the panel options), SEX, RACE, and ETHNIC.

Analyses 2&3: Age Group Counts and Percents

1. Create a variable called Age Group
    - 1.1 The user will be asked to specify Age Groups in the Script Launcher interface. The Age Groups variable will contain the following 2 default age groups and ranges that will populate for the automatic run
        - 1.1.1 0 years to < 65 years
        - 1.1.2 $\geq$ 65 years
    - 1.2 Instead of using the default Age Groups, this variable can also be customized. The user can to choose between 2 and 6 age groups and the unit of time the ages are defined in from years, months, weeks, and days.

2. Provide a frequency count and percentage of the Age Groups variable created in Analyses 2&3 sections 1.1.1 – 1.1.2 or 1.2 by arm. Percentages are calculated by using the frequency count of the age groups defined in Analyses 2&3 sections 1.1.1 – 1.1.2 or 1.2 as the numerator and the arm subject count as the denominator.
    - 2.1 The denominator is based on the total number of subjects in the DM dataset that have been assigned to a treatment group in the arm variable. Formula depicted in Figure 1.

Figure 1:

$$f(ARM) = \sum_{\forall S_i} X_i = \left(X_1 + X_2 + X_3 + \cdots + X_M\right)$$

where

S_i is the ith subject

$X_i = g(S_i, ARM)$

$$X_i = \begin{cases} 1 & \text{if } i\text{th subject } S_i \text{ is in ARM} \\ 0 & \text{otherwise} \end{cases}$$

3. Provide a frequency count and percentage of the Age Groups variable created in Analyses 2&3 sections 1.1.1 – 1.1.2 or 1.2 by DSDECOD and arm. Percentages are calculated by using the frequency count of each age group by DSDECOD and arm as the numerator and the Age Group variable count for each category by arm as the denominator. Formula depicted in Figure 1.

4. The outputs produced in Analyses 2&3 shall be exported into an Excel workbook and defined as:

    4.1. For analysis 2, a table using the data output by section 2 consisting of the values of age group down the rows and the arms (and overall) across the

columns. The cells at the intersection of the two will contain the counts and percents for each age group in the respective arm.

4.2. For analysis 2, a bar graph by arm and for overall with percentage of subjects on the y-axis and age group on the x-axis.

4.3. For analysis 3, a table from the data output by section 3 consisting of the disposition event terms and the age groups in each disposition event term down the rows, and the arms (and overall) across the columns. The cells at the intersection of the two will contain the counts of subjects of a given age group in a given arm that had the disposition event and what percent of the age group had the disposition event.

Analysis 4: Age Descriptive Statistics

1. Produce descriptive statistics of the AGE variable

1.1 Mean (Formula depicted in Figure 2)

Figure 2

$$\frac{1}{n}\sum_{i=1}^{n} x_i$$

i represents the index summation
n is the upper bound of the summation
x is the variable of interest

1.2 Median
The middle value of the given numbers or distribution in their ascending order

1.3 Mode
The most frequently occurring value in a frequency distribution

1.4 Standard Deviation

Figure 3

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2}$$

i represents the index summation
x is variable of interest
N is the upper bound of the summation, i.e the number of subjects exposed to at least one dose of treatment

1.5 1st Quartile (25th Percentile)

Figure 4

$$\frac{P}{100}\times\left(N+1\right)$$

P is the desired percentile
N is the number of values

1.6 3rd Quartile (75th Percentile, Formula depicted in Figure 4)

1.7 Minimum
Is the smallest value

1.8 Maximum
Is the largest value

2. Produce descriptive statistics of the AGE variable by arm
   - 2.1 Mean (Formula depicted in Figure 2)
   - 2.2 Median (Analysis 4 section 1.2)
   - 2.3 Mode (Analysis 4 section 1.3)
   - 2.4 Standard Deviation (Formula depicted in Figure 3)
   - 2.5 1st Quartile (25th Percentile, Formula depicted in Figure 4)
   - 2.6 3rd Quartile (75th Percentile, Formula depicted in Figure 4)
   - 2.7 Minimum (Analysis 4 section 1.7)
   - 2.8 Maximum (Analysis 4 section 1.8)

3. The outputs produced in Analysis 4 shall be exported into an Excel document and defined as:

   3.1 Tables
   - 3.1.1. A column for each arm and for overall
   - 3.1.2 A row containing the arm name and a row containing the arm population, as 'N=' the arm population
   - 3.1.3 This table will list all of the statistics produced in Analysis 4 sections 1 and 2
   - 3.1.4 The Standard Deviation will be next to mean in parentheses

   3.2 Graph Analysis 4 sections 1 and 2
   - 3.2.2 Box and Whiskers graph by arm and Overall Population with Age on the Y-Axis and Arm on the X-Axis.

Analyses 5&6: Sex Counts and Percents

1. Provide a frequency count and percentage of the SEX variable by arm. Percentages are calculated by using the frequency count of the sexes as the numerator and the arm subject count as the denominator. The denominator is based on the total number of subjects in the DM dataset that have been assigned to a treatment group. Formula depicted in Figure 1.

2. Provide a frequency count and percentage of the SEX variable by DSDECOD and arm. Percentages are calculated by using the frequency count of each sex by DSDECOD and arm as the numerator and the frequency count of each sex by arm as the denominator. Formula depicted in Figure 1.

3. The outputs produced in Analyses 5&6 shall be exported into an Excel workbook and defined as:

   - 3.1. For analysis 5, a table using the data output by section 2 consisting of the values of sex down the rows and the arms (and overall) across the columns. The cells at the intersection of the two will contain the counts and percents for each sex in the respective arm.
   - 3.2. For analysis 5, a bar graph by arm and for overall with percentage of subjects on the y-axis and sex on the x-axis.
   - 3.3. For analysis 6, a table from the data output by section 3 consisting of the disposition event terms and the sex in each disposition event term down the

rows, and the arms (and overall) across the columns. The cells at the intersection of the two will contain the counts of subjects of a sex in a given arm that had the disposition event and what percent of the sex had the disposition event.

Analyses 7&8: Race Counts and Percents

1. Provide a frequency count and percentage of the RACE variable by arm. Percentages are calculated by using the frequency count of the races as the numerator and the arm subject count as the denominator. The denominator is based on the total number of subjects in the DM dataset that have been assigned to a treatment group. Formula depicted in Figure 1.

2. Provide a frequency count and percentage of the RACE variable by DSDECOD and arm. Percentages are calculated by using the frequency count of each race by DSDECOD and arm as the numerator and the frequency count of each race by arm as the denominator. Formula depicted in Figure 1.

3. The outputs produced in Analyses 7&8 shall be exported into an Excel workbook and defined as:

    3.1. For analysis 7, a table using the data output by section 2 consisting of the values of race down the rows and the arms (and overall) across the columns. The cells at the intersection of the two will contain the counts and percents for each race in the respective arm.
    3.2. For analysis 7, a bar graph by arm and for overall with percentage of subjects on the y-axis and race on the x-axis.
    3.3. For analysis 8, a table from the data output by section 3 consisting of the disposition event terms and the race in each disposition event term down the rows, and the arms (and overall) across the columns. The cells at the intersection of the two will contain the counts of subjects of a race in a given arm that had the disposition event and what percent of the race had the disposition event.

Analyses 9&10: Ethnicity Counts and Percents

1. Provide a frequency count and percentage of the ETHNIC variable by arm. Percentages are calculated by using the frequency count of the ethnicities as the numerator and the arm subject count as the denominator. The denominator is based on the total number of subjects in the DM dataset that have been assigned to a treatment group. Formula depicted in Figure 1.

2. Provide a frequency count and percentage of the ETHNIC variable by DSDECOD and arm. Percentages are calculated by using the frequency count of each ethnicity by DSDECOD and arm as the numerator and the frequency count of each ethnicity by arm as the denominator. Formula depicted in Figure 1.

3. The outputs produced in Analyses 9&10 shall be exported into an Excel workbook and defined as:

    3.1. For analysis 9, a table using the data output by section 2 consisting of the values of ethnicity down the rows and the arms (and overall) across the columns. The cells at the intersection of the two will contain the counts and percents for each ethnicity in the respective arm.
    3.2. For analysis 9, a bar graph by arm and for overall with percentage of subjects

on the y-axis and ethnicity on the x-axis.

3.3. For analysis 10, a table from the data output by section 3 consisting of the disposition event terms and the ethnicity in each disposition event term down the rows, and the arms (and overall) across the columns. The cells at the intersection of the two will contain the counts of subjects of each ethnicity in a given arm that had the disposition event and what percent of the ethnicity had the disposition event.

Analyses 11&12: Country Counts and Percents

1. Provide a frequency count and percentage of the COUNTRY variable by arm. Percentages are calculated by using the frequency count of the countries as the numerator and the arm subject count as the denominator. The denominator is based on the total number of subjects in the DM dataset that have been assigned to a treatment group. Formula depicted in Figure 1.

2. Provide a frequency count and percentage of the COUNTRY variable by DSDECOD and arm. Percentages are calculated by using the frequency count of each country by DSDECOD and arm as the numerator and the frequency count of each country by arm as the denominator. Formula depicted in Figure 1.

3. The outputs produced in Analyses 11&12 shall be exported into an Excel workbook and defined as:

    3.1. For analysis 11, a table using the data output by section 2 consisting of the values of country down the rows and the arms (and overall) across the columns. The cells at the intersection of the two will contain the counts and percents for each country in the respective arm.
    3.2. For analysis 11, a bar graph by arm and for overall with percentage of subjects on the y-axis and country on the x-axis.
    3.3. For analysis 12, a table from the data output by section 3 consisting of the disposition event terms and the countries in each disposition event term down the rows, and the arms (and overall) across the columns. The cells at the intersection of the two will contain the counts of subjects of each country in a given arm that had the disposition event and what percent of the country had the disposition event.

Analyses 13&14: Site ID Counts and Percents

1. Provide a frequency count and percentage of the SITEID variable by arm. Percentages are calculated by using the frequency count of the site IDs as the numerator and the arm subject count as the denominator. The denominator is based on the total number of subjects in the DM dataset that have been assigned to a treatment group. Formula depicted in Figure 1.

2. Provide a frequency count and percentage of the SITEID variable by DSDECOD and arm. Percentages are calculated by using the frequency count of each site ID by DSDECOD and arm as the numerator and the frequency count of each country by arm as the denominator. Formula depicted in Figure 1.

3. The outputs produced in Analyses 13&14 shall be exported into an Excel workbook and defined as:

    3.1. For analysis 13, a table using the data output by section 2 consisting of the values of site ID down the rows and the arms (and overall) across the columns. The cells at the intersection of the two will contain the counts and

percents for each site ID in the respective arm.

3.2. For analysis 13, a bar graph by arm and for overall with percentage of subjects on the y-axis and site ID on the x-axis.

3.3. For analysis 14, a table from the data output by section 3 consisting of the disposition event terms and the site IDs in each disposition event term down the rows, and the arms (and overall) across the columns. The cells at the intersection of the two will contain the counts of subjects of each site ID in a given arm that had the disposition event and what percent of the site ID had the disposition event.

Analyses 15: Country and Site ID Counts and Percents

1. Provide a frequency count and percentage of the COUNTRY and SITEID variables together by arm. Percentages are calculated by using the frequency count of the pairs of country and site ID as the numerator and the arm subject count as the denominator. The denominator is based on the total number of subjects in the DM dataset that have been assigned to a treatment group. Formula depicted in Figure 1.

2. The outputs produced in Analyses 15 shall be exported into an Excel workbook and defined as:

   2.1 A table consisting of each country and site ID pair down the rows and the arms (and overall) across the columns. The cells at the intersection of the two will contain the counts and percents for each country and site ID pair in the respective arm.

Screenshots:

Analysis 1: Overview

Demographic Baseline Characteristics: Overview

NDA/BLA: 12345
Study: 123
Analysis run date: 2011-04-07 12:57:01 PM

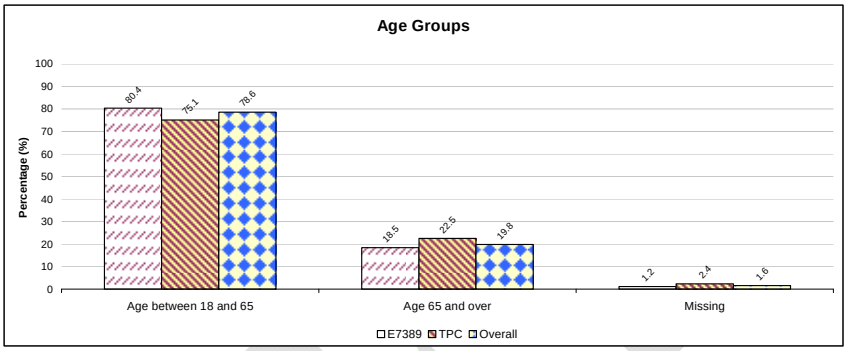
Demographic Baseline Characteristics		E7389 N=509		TPC N=253		Overall N=762	
Age	Mean (SE)	54.8 (10.4)		56.2 (10.3)		55.3 (10.4)	
	Min	28		27		27	
	Q1	47		49		48	
	Median	55		57		56	
	Q3	62		64		63	
	Max	85		81		85	
		Count	%	Count	%	Count	%
Age Group	Age between 18 and 65	409	80.4	190	75.1	599	78.6
	Age 65 and over	94	18.5	57	22.5	151	19.8
	Missing	6	1.2	6	2.4	12	1.6
Sex	F	509	100.0	253	100.0	762	100.0
Race	Asian/Pacific Islander	3	0.6	2	0.8	5	0.7
	Black	20	3.9	14	5.5	34	4.5
	White	471	92.5	232	91.7	703	92.3
	Other	15	2.9	5	2.0	20	2.6
Ethnicity	Missing	509	100.0	253	100.0	762	100.0

Analysis 2: Age Groups

Demographic Baseline Characteristics: Age Groups

NDA/BLA: 12345
Study: 123
Analysis run date: 2011-04-07 12:57:01 PM

Age Group	E7389 N=509		TPC N=253		Overall N=762	
	Count	%	Count	%	Count	%
Age between 18 and 65	409	80.4	190	75.1	599	78.6
Age 65 and over	94	18.5	57	22.5	151	19.8
Missing	6	1.2	6	2.4	12	1.6



Analysis 3: Age Groups by Disposition

Demographic Baseline Characteristics by Disposition Term: Age Groups

NDA/BLA: 12345
Study: 123
Analysis run date: 2011-04-07 12:57:01 PM

Counts in the Count column are the count of subjects in each demographic group per arm who had the given disposition term.
Percentages in the % column are the percentage of the demographic group per arm who had the disposition term.

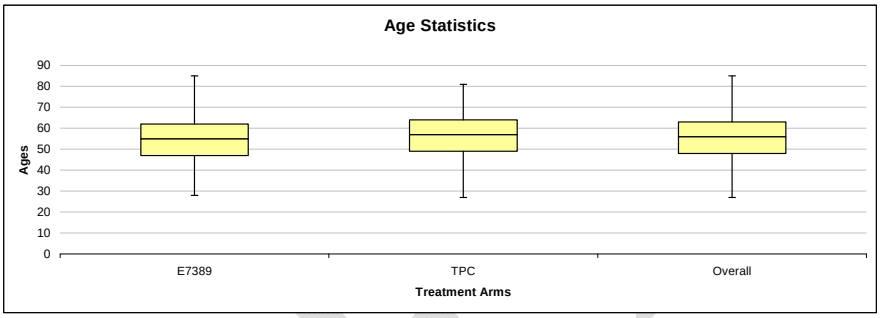
Standard Disposition Term	Age Group	E7389 N=509		TPC N=253		Overall N=762	
		Count	%	Count	%	Count	%
Adverse Event	Age between 18 and 65	36	8.8	18	9.5	54	9.0
	Age 65 and over	13	13.8	6	10.5	19	12.6
	Missing	1	16.7	0	0.0	1	8.3
Alive	Age between 18 and 65	306	74.8	131	68.9	437	73.0
	Age 65 and over	72	76.6	35	61.4	107	70.9
	Missing	2	33.3	2	33.3	4	33.3
Clinical Progression	Age between 18 and 65	51	12.5	30	15.8	81	13.5
	Age 65 and over	10	10.6	6	10.5	16	10.6
	Missing	1	16.7	0	0.0	1	8.3
Death	Age between 18 and 65	235	57.5	132	69.5	367	61.3
	Age 65 and over	57	60.6	31	54.4	88	58.3
	Missing	4	66.7	4	66.7	8	66.7
Informed Consent Obtained	Age between 18 and 65	409	100.0	190	100.0	599	100.0
	Age 65 and over	94	100.0	57	100.0	151	100.0
	Missing	6	100.0	6	100.0	12	100.0
Lost To Follow-Up	Age between 18 and 65	4	1.0	3	1.6	7	1.2
	Age 65 and over	1	1.1	2	3.5	3	2.0
	Missing	1	16.7	0	0.0	1	8.3
Other	Age between 18 and 65	3	0.7	8	4.2	11	1.8
	Age 65 and over	2	2.1	0	0.0	2	1.3
	Missing	1	16.7	1	16.7	2	16.7
Physician Decision	Age between 18 and 65	17	4.2	6	3.2	23	3.8
	Age 65 and over	1	1.1	5	8.8	6	4.0
	Missing	1	16.7	1	16.7	2	16.7
Progressive Disease	Age between 18 and 65	272	66.5	118	62.1	390	65.1
	Age 65 and over	64	68.1	35	61.4	99	65.6
	Missing	1	16.7	1	16.7	2	16.7
Randomized	Age between 18 and 65	409	100.0	190	100.0	599	100.0
	Age 65 and over	94	100.0	57	100.0	151	100.0
	Missing	6	100.0	6	100.0	12	100.0
Withdrawal By Subject	Age between 18 and 65	8	2.0	5	2.6	13	2.2
	Age 65 and over	1	1.1	0	0.0	1	0.7
	Missing	1	16.7	2	33.3	3	25.0

Analysis 4: Age Statistics

Demographic Baseline Characteristics: Age Statistics

NDA/BLA: 12345
Study: 123
Analysis run date: 2011-04-07 12:57:01 PM

Age Statistic	E7389 N=509	TPC N=253	Overall N=762
Mean (SE)	54.8 (10.4)	56.2 (10.3)	55.3 (10.4)
Mode	59	48	59
Min	28	27	27
Q1	47	49	48
Median	55	57	56
Q3	62	64	63
Max	85	81	85



Analysis 15: Country and Site ID

Demographic Baseline Characteristics: Country and Site ID

NDA/BLA: 12345
Study: 123
Analysis run date: 2011-04-07 12:57:01 PM

Country	Site ID	E7389 N=509		TPC N=253		Overall N=762	
		Count	%	Count	%	Count	%
ARG	2904	4	0.8	3	1.2	7	0.9
	2911	2	0.4	4	1.6	6	0.8
	2914	2	0.4	3	1.2	5	0.7
	2909	4	0.8	1	0.4	5	0.7
	2905	2	0.4	1	0.4	3	0.4
	2912	1	0.2	1	0.4	2	0.3
	2902	1	0.2	1	0.4	2	0.3
	2908	1	0.2	0	0.0	1	0.1
	2907	0	0.0	1	0.4	1	0.1
	2906	1	0.2	0	0.0	1	0.1
AUS	2502	8	1.6	1	0.4	9	1.2
	2505	2	0.4	6	2.4	8	1.0
	2501	3	0.6	1	0.4	4	0.5
	2507	3	0.6	0	0.0	3	0.4
	2506	2	0.4	0	0.0	2	0.3
	2503	0	0.0	2	0.8	2	0.3
BEL	1106	6	1.2	7	2.8	13	1.7
	1101	8	1.6	2	0.8	10	1.3
	1104	4	0.8	3	1.2	7	0.9
	1102	3	0.6	2	0.8	5	0.7
	1103	2	0.4	1	0.4	3	0.4

Output(s):

The Demographic Analysis Panel provides the following outputs:

1. Tables

   1.1. Separate tables contain the Demographic variables Race, Ethnicity, Sex, Country and Site ID and the derived variable Age Groups counts and percents for overall treatment population, treatment arm and Disposition categories.

   1.2. Table contains the descriptive statistics Mean, Median, Mode, Standard Deviation, Quartile 1, Quartile 3, Minimum and Maximum for the Age variable for overall treatment population, treatment arm and Disposition categories.

   1.3. Table contains the combined counts and percents for Age Groups, Race, Ethnicity and Sex and the Mean, Median, Standard Deviation, Minimum and Maximum for Age for overall treatment population and treatment arm.

	2. Charts 2.1. Charts with the Demographic variables Race, Ethnicity and Sex and the derived variable Age Groups as a percentage of the overall study population and by treatment arm. 2.2. Chart with the descriptive statistics Mean, Median, Mode, Standard Deviation, Quartile 1, Quartile 3, Minimum and Maximum for the Age variable by treatment arm.		
Assumptions:	The Demographic Analysis Panel assumes: 1. The User has access to the Demographics CDISC SDTM DM and DS datasets.		
Notes and Issues:	**Limitations:** 1. The Demographic Analysis Panel is performed according to the terms submitted by the sponsor – this means the analysis only displays the terminologies provided by the sponsor. 1.1. This constrains the automatic processes of the analysis since at the moment there is a lack of terminology consistency across sponsor submissions. 1.2. If the user wants to change terms they can use the grouping feature in the Script Launcher interface. 2. The Demographic Analysis Panel uses the planned treatment arm (ARM) if the actual treatment arm (ACTARM) variable is unavailable 3. No graphics are provided for site ID and country since these variables can have numerous values 4. This analysis only shows six treatment arm so if more than six treatment arms exist only the first six (in alphabetical order) will be shown		
Approval / Validation Status	Status Pending and Validation Required		
Created By:	Shannon Dennis	**Last Updated By:**	David Kretch
Date Created:	1/8/2010	**Date Last Updated:**	6/14/2010

This script is for example purposes only and should not be interpreted to represent best practices for industry or FDA

Appendix A – SAS Code

```
/*************************************************************************/
/*         PROGRAM NAME: Demographics Analysis Panel                     */
/*                                                                       */
/*          DESCRIPTION: Find subject counts and % per arm               */
/*                       Find subject counts and % per disposition and arm  */
/*                        for age group, sex, race, ethnicity, country,  */
/*                        and site ID                                    */
/*                       Find summary statistics for age                 */
/*                       Output to Excel template                        */
/*                                                                       */
/*      EVALUATION TYPE:                                                 */
/*                                                                       */
/*               AUTHOR: Shannon Dennis (shannon.dennis@fda.hhs.gov)     */
/*                       David Kretch (david.kretch@us.ibm.com           */
/*                                                                       */
/*                 DATE: December 29, 2009                               */
/*                                                                       */
/*  EXTERNAL FILES USED: Demographics_Template.xls -- Excel template     */
/*                       data_checks.sas -- Generic variable checks      */
/*                       sl_gs_output.sas -- Script Launcher settings output*/
/*                       err_output.sas -- Error output when missing vars   */
/*                                                                       */
/*  PARAMETERS REQUIRED: utilpath -- location of external SAS programs   */
/*                       ndabla -- NDA or BLA number                     */
/*                       studyid -- study number                         */
/*                       age1 - age8 -- upper bound of age bucket        */
/*                       ageunit -- unit of time age buckets are in      */
/*                       demout -- filename and path of output           */
/*                                                                       */
/*           LOCAL ONLY: studypath -- location of the drug study datasets   */
/*                       outpath -- location of the output               */
/*                       outfile -- filename of the output               */
/*                       templatepath -- location of the Excel template  */
/*                       template -- filename of the template            */
/*                                                                       */
/*   VARIABLES REQUIRED: DM -- ACTARM or ARM                             */
/*                              USUBJID                                  */
/*                       DS -- DSDECOD                                   */
/*                              USUBJID                                  */
/*                              DSSTDTC or DSTDDY                        */
/*                                                                       */
/* OTHER VARIABLES USED: DM -- ARMCD                                     */
/* IF AVAILABLE                 AGE                                      */
/*                              COUNTRY                                  */
/*                              ETHNIC                                   */
/*                              RACE                                     */
/*                              SEX                                      */
/*                              SITEID                                   */
/*                       DS -- DSCAT                                     */
/*                              DSSCAT                                   */
/*                              DSSEQ                                    */
/*                                                                       */
/*            MADE WITH: SAS 9.2                                         */
/*                                                                       */
/*                NOTES:                                                 */
/*                                                                       */
/*************************************************************************/

/* REVISIONS */
/*
2011-03-09  DK  Incorporated data checks & grouping/subsetting info
                Updated template style

2011-03-27  DK  Run location handling

2011-05-08  DK  Error/no subjects in DM handling

2011-05-18  DK  Made demographics by disposition keep only the last disposition event
```

```
                per subject/category/subcategory for categories other than protocol
milestone

2011-05-23  DK  Keep disposition events for terms informed consent obtained and
randomized

2011-06-02  DK  Fixed lkp_arm SQL to exclude screen failures
                Added steps to merge counts/percents onto arm numbers prior to tranpose
                to avoid missing variables
*/

/* the minoperator option allows the use of the in operator by the macro language */
options minoperator;
options missing='';

/* determine the run location by looking for the SL-set run_location macro variable */
%sysfunc(ifc(not %symexist(run_location),%nrstr(%let run_location = local;),));
%put RUN LOCATION: &run_location.;

%macro params;

        /* program parameters if the program is run locally */
        %if %upcase(&run_location.) = LOCAL %then %do;

                data macrovar; set sashelp.vmacro(keep=scope name where=(scope='GLOBAL' &
name ne 'RUN_LOCATION')); run;
                data _null_; set macrovar; call execute('%symdel
'||trim(left(name))||';');    run;
                proc datasets kill; quit;

                %global panel_title panel_desc;
                %let panel_title = Demographics;
                %let panel_desc = ;

                /* age bucket limits */
                %global age_grp1 age_grp2 age_grp3 age_grp4 age_grp5 age_grp6 age_grp7
age_grp8;
                %let age_grp1 = 1/12 yr;
                %let age_grp2 = 2 yr;
                %let age_grp3 = 5 yr;
                %let age_grp4 = 12 yr;
                %let age_grp5 = 16 yr;
                %let age_grp6 = 18 yr;
                %let age_grp7 = 65 yr;

                %global ageunit;
                %let ageunit = years;

                /* NDA/BLA number */
                /* study number */
                %global ndabla studyid;
                %let ndabla = 12345;
                %let studyid = 123;

                /* location of the study being examined */
                %let studypath = C:\Documents and Settings\MATTOK\Desktop\Data\NDA201532;

                libname inlib "&studypath.";

                /* retrieve datasets */
                data dm; set inlib.dm; run;
                data ds; set inlib.ds; run;

                /* location of external SAS programs */
                %global utilpath;
                %let utilpath = C:\Documents and
Settings\MATTOK\Desktop\SL_SAS_Progs\ZZ_Utilities;

                /* location and filename of the disposition panel template */
                %let templatepath = C:\Documents and
Settings\MATTOK\Desktop\SL_Templates\DemographicsGeneral;
```

```
		%let template = Demographics_Template.xls;

		/* location and filename of the output */
		%let outpath = C:\Documents and Settings\MATTOK\Desktop\Output;
		%let outfile = Demographics.xls;

		%global demout errout;
		%let demout = &outpath.\&outfile.;
		%let errout = &outpath.\Demographics Error Summary.xls;

		options noxwait xsync;

		/* copy the template to the output file */
		x "%str(copy %"&templatepath.\&template.%" %"&demout.%")";

		/* dummy grouping, subsetting, and dataset information datasets */
		data sl_datasets;
			datatype = ''; name = ''; partition_variable = ''; default = '';
			delete;
		run;
		data sl_group;
			group_name = ''; domain = ''; partition = ''; var_name = '';
var_value = ''; dsvg_grp_name = '';
			delete;
		run;
		data sl_subset;
			name = ''; domain = ''; partition = ''; var_name = ''; var_value =
'''; inner_operator = ''; outer_operator = '';
			delete;
		run;

	%end;

	/* program parameters if the program is run through Script Launcher */
	%else %do;

		/* map the user-defined Script Launcher panel option values onto panel
macro variables */
		data _null_;
		run;

	%end;

%mend params;

%params;

/* list of demographic variables to examine  */
%let dm_var = age_flag country ethnic race sex siteid country*siteid;
%let dm_by_ds_var = age_flag country ethnic race sex siteid;

/* data checks and data check output */
%include "&utilpath.\data_checks.sas";
%include "&utilpath.\sl_gs_output.sas";
%include "&utilpath.\err_output.sas";

/*****************/
/* SETUP ROUTINE */
/*****************/
%macro dm_setup;

	/* check whether there are subjects in DM */
	%chk_dm_subj_gt0;

	/* data checks */
	/* required variables */
	%chk_var(ds=dm,var=usubjid);
	%chk_var(ds=dm,var=age);
	%chk_var(ds=ds,var=dsdecod);
	%chk_var(ds=ds,var=usubjid);
```

```
	data rpt_chk_var_req; set rpt_chk_var; run;

	/* actual arm or planned arm */
	%chk_var(ds=dm,var=actarm);
	%chk_var(ds=dm,var=arm);

	/* insert a row into RPT_CHK_VAR_REQ indicating whether actual arm OR arm exists
*/
	proc sql noprint;
		insert into rpt_chk_var_req
		set chk = 'VAR',
		    ds = 'DM',
			var = 'ACTARM or ARM',
			condition = 'EXISTS',
			ind = %sysfunc(ifc(&dm_actarm. or &dm_arm.,1,0));
	quit;

	/* study day of disposition event OR start date of disposition event */
	%chk_var(ds=ds,var=dsstdtc);
	%chk_var(ds=ds,var=dsstdy);

	proc sql noprint;
		insert into rpt_chk_var_req
		set chk = 'VAR',
		    ds = 'DM/DS',
			var = 'DSSTDY or DSSTDTC',
			condition = 'EXISTS',
			ind = %sysfunc(ifc(&ds_dsstdy. or &ds_dsstdtc.,1,0));
	quit;

	/* optional variables */
	%chk_var(ds=dm,var=armcd);
	%chk_var(ds=dm,var=ageu);
	%chk_var(ds=dm,var=country);
	%chk_var(ds=dm,var=ethnic);
	%chk_var(ds=dm,var=race);
	%chk_var(ds=dm,var=sex);
	%chk_var(ds=dm,var=siteid);

	%chk_var(ds=ds,var=dscat);
	%chk_var(ds=ds,var=dsscat);
	%chk_var(ds=ds,var=dsseq);

	/* set indicator whether all required variables are present */
	%global setup_req_var;
	proc sql noprint;
		select (case when count = 0 then 1 else 0 end) into: setup_req_var
		from (select count(1) as count
	      from rpt_chk_var_req
	      where ind ne 1);
	quit;

	%if &dm_subj_gt0. and &setup_req_var. %then %do;

		%global setup_success;
		%let setup_success = 1;

		/* used ACTARM if it is available */
		%if &dm_actarm. %then %do;
			proc datasets library=work;
				modify dm;
				rename %if &dm_arm. %then arm=plannedarm;
					actarm=arm;
			quit;
		%end;

		/* arm info */
		proc sql noprint;
			create table lkp_arm as
			select arm, count(1) as arm_count
			from dm
```

```
                where armcd not in ('SCRNFAIL','NOTASSGN')
                /*and upcase(arm) ne 'SCREEN FAILURE'*/
                group by arm
                order by arm;
        quit;

        /* get arm counts and total count; assign arm numbers */
        data lkp_arm;
                set lkp_arm end=eof;

                arm_num = _n_;

                /* format arm name */
                arm_display = arm;
                if arm_display ne '' and not anylower(arm_display) then do;
                        length arm_word $50;
                        i = 1;
                        arm_word = scan(arm_display,i);
                        do while (arm_word ne '');
                                if length(arm_word) > 3 and not
anydigit(compress(arm_word))
                                                        then
substr(arm_display,index(arm_display,compress(arm_word)),length(compress(arm_word))) =
propcase(compress(arm_word));
                                if compress(arm_word) in ('MG' 'KG')
                                        then
substr(arm_display,index(arm_display,compress(arm_word)),length(compress(arm_word))) =
lowcase(arm_word);
                                if compress(arm_word) = ('ML')
                                        then
substr(arm_display,index(arm_display,compress(arm_word)),length(compress(arm_word))) =
'mL';
                                i = i + 1;
                                arm_word = scan(arm_display,i);
                        end;
                end;

                retain total_count 0;
                total_count = total_count + arm_count;

                call symputx('arm_name_'||put(arm_num,8. -
l),trim(arm_display),'g');
                call symputx('arm_count_'||put(arm_num,8. -l),trim(put(arm_count,8.
-l)),'g');
                if eof then do;
                        call symputx('arm_count',trim(put(_n_,8. -l)),'g');
                        call symputx('total_count',trim(put(total_count,8. -
l)),'g');
                end;
        run;

        data lkp_arm_out;
                retain arm arm_count;
                set lkp_arm(keep=arm_display arm_count total_count
rename=(arm_display=arm)) end=eof;
                output;
                if eof then do;
                        arm = 'Overall';
                        arm_count = total_count;
                        output;
                end;
                keep arm arm_count;
        run;

        data dm_original;
                set dm;
        run;

        /* look up arm numbers for each subject */
        data dm;
                set dm;
```

```
                %if &dm_armcd. %then %do;
                        if armcd in ('SCRNFAIL','NOTASSGN') then delete;
                %end;
                /*if upcase(arm) = 'SCREEN FAILURE' then delete;*/

                if _n_ = 1 then do;
                        declare hash h(dataset:'lkp_arm');
                        h.definekey('arm');
                        h.definedata('arm_num');
                        h.definedone();
                end;

                length arm_num 8.;
                call missing(arm_num);
                rc = h.find();
                drop rc;

                if not &dm_age. then age = .;
                if not &dm_country. then country = 'Missing';
                if not &dm_ethnic. then ethnic = 'Missing';
                if not &dm_race. then race = 'Missing';
                if not &dm_sex. then sex = 'Missing';
                if not &dm_siteid. then siteid = 'Missing';

                %let i = 1;
                %let var = %scan(country ethnic race sex,&i.);
                %do %while (&var. ne );
                        %if &&&dm_&var.. %then %do;
                                %if &&&dm_&var._len. < 9 %then %do;
                                        length &var._ext $9;
                                        if &var. = '' then &var._ext = 'Missing';
                                        else &var._ext = &var.;
                                        rename &var.=dm_&var.;
                                        rename &var._ext=&var.;
                                %end;
                                %else %do;
                                        if &var. = '' then &var. = 'Missing';
                                %end;
                        %end;
                        %let i = %eval(&i.+1);
                        %let var = %scan(country ethnic race sex,&i.);
                %end;


                ethnic = propcase(ethnic);
                race = propcase(race);
        run;

        /*****************/
        /* AGE BUCKETING */
        /*****************/

        /* determine number of age flag buckets */
        data lkp_age;
                length age_sl 8. ageu_sl $6;
                %let i = 1;
                %do %while (%symexist(age_grp&i.));
                        %if (%length(&&&age_grp&i.) = 0) %then %goto age_exit;
                        %else %do;
                                age_sl = %sysevalf(%scan(&&&age_grp&i.,1,%str( )));
                                /*ageu_sl = ifc(not
missing("%scan(&&&age_grp&i.,2)"),lowcase("%scan(&&&age_grp&i.,2)"),'years');*/
                                ageu_sl = "&ageunit.";
                                output;
                        %end;
                        %let i = %eval(&i. + 1);
                %end;
                %age_exit: %let age_count = %eval(&i. - 1);
        run;
```

```
		/* this gets rid of duplicate rows */
		/* the database puts the catchall category in a separate row, whose value
gets written as its min, */
		/* which is a duplicate of the next to last bucket's max */
		proc sort data=lkp_age nodupkey; by age_sl ageu_sl; run;

		/* convert age input from Script Launcher to age in years */
		data lkp_age;
			set lkp_age end=eof;

			label min_age='Age Bucket Min Age'
			max_age='Age Bucket Max Age'
				min_age_yr='Age Bucket Min Age in Years'
			max_age_yr='Age Bucket Max Age in Years';

			max_age = age_sl;
			min_age = lag(max_age);

			/* convert SL ages to ages in years */
			select (lowcase(compress(ageu_sl,,'ak')));
				when ('yr','year','years') max_age_yr = age_sl;
				when ('mo','month','months') max_age_yr = age_sl/12;
				when ('wk','week','weeks') max_age_yr =
age_sl/52.178571428571428571428571428571;
				when ('dy','day','days') max_age_yr = age_sl/365.25;
				when ('hr','hour','hours') max_age_yr = age_sl/8766;
				otherwise max_age_yr = .;
			end;

			min_age_yr = lag(max_age_yr);

			/* first age bucket */
			if _n_ = 1 then do;
				min_age = 0;
				min_age_yr = 0;
			end;

			output;

			/* last age bucket */
			if eof then do;
				min_age = max_age;
				min_age_yr = max_age_yr;
				max_age = 200;
				max_age_yr = 200;
				output;
			end;
		run;

		/* assign age bucket labels */
		data lkp_age;
			set lkp_age end=eof;

			length min_ageu $6 max_ageu $6
			min_ageu_txt $6 max_ageu_txt $6;

			select (lowcase(compress(ageu_sl,,'ak')));
				when ('yr','year','years') ageu_txt =
ifc(age_sl<=1,'year','years');
				when ('mo','month','months') ageu_txt =
ifc(age_sl<=1,'month','months');
				when ('wk','week','weeks') ageu_txt =
ifc(age_sl<=1,'week','weeks');
				when ('dy','day','days') ageu_txt =
ifc(age_sl<=1,'day','days');
				when ('hr','hour','hours') ageu_txt =
ifc(age_sl<=1,'hour','hours');
				otherwise max_age_yr = .;
			end;

			max_ageu = ageu_txt;
```

```
                        min_ageu = lag(max_ageu);

                        if min_ageu =: max_ageu =: 'year' then do;
                                min_ageu_txt = '';
                                max_ageu_txt = '';
                        end;
                        else do;
                                min_ageu_txt = min_ageu;
                                max_ageu_txt = max_ageu;
                        end;

                        length age_flag $50;
                        if _n_ = 1 then age_flag = 'Age under '||trim(put(max_age,fract8. -
l))||' '||trim(max_ageu_txt);
                        else if eof then age_flag = 'Age '||trim(put(min_age,fract8. -
l))||' '||trim(min_ageu_txt)||' and over';
                        else age_flag = 'Age between '||trim(put(min_age,fract8. -l))||'
'||trim(min_ageu_txt)||
                                        ' and '||trim(put(max_age,fract8. -l))||'
'||trim(max_ageu_txt);
                        age_flag = compbl(age_flag);

                        /* add order number */
                        age_order = _n_;

                        output;

                        /* add missing age bucket */
                        if eof then do;
                                age_flag = 'Missing';
                                call
missing(min_age,min_age_yr,min_ageu,max_age,max_age_yr,max_ageu); age_order = 99;
                                output;
                        end;

                        drop age_sl ageu_sl ageu_txt min_ageu_txt max_ageu_txt;
                run;

                data lkp_age;
                        retain age_flag min_age min_ageu max_age max_ageu min_age_yr
max_age_yr age_order;
                        set lkp_age;
                run;

                /* add the age flags to the DM dataset */
                proc sql noprint;
                        create table dm_age_flag as
                        select a.*, b.age_flag
                        from dm a,
                      (select a.usubjid,
                              (case when b.age_flag is not missing then b.age_flag else
'Missing' end) as age_flag
                                from (select usubjid,
                                                %if &dm_ageu. %then %do;
                                        (case
                                                                when upcase(ageu) like
'YEAR_' then age
                                                                when upcase(ageu) like
'MONTH_' then age*12
                                                                when upcase(ageu) like
'WEEK_' then age*52.178571428571428571428571428571
                                                                when upcase(ageu) like
'DAY_' then age*365.25
                                                                when upcase(ageu) like
'HOUR_' then age*8766
                                                                else .
                                                end) as age
                                                %end;
                                                %else %do;
                                                age
                                                %end;
```

```
                                        from dm) a
                                left join lkp_age b
                              on b.min_age_yr <= age < max_age_yr)   b
                        where a.usubjid = b.usubjid
                        order by usubjid;

                        drop table dm;
                quit;

                proc datasets library=work nolist;
                        change dm_age_flag=dm;
                quit;

                proc sort data=dm; by usubjid; run;
                proc sort data=ds; by usubjid; run;

                /* merge demographic information onto the disposition domain dataset */
                data ds_dm;
                        merge ds(in=a)
                              dm(in=b);
                        by usubjid;
                        if a and b;

                        if not anylower(dsdecod) then dsdecod = propcase(dsdecod);

                        if not &ds_dscat. then dscat = 'Missing';
                        if not &ds_dsscat. then dsscat = 'Missing';

                        %if &ds_dsstdtc. %then %do;
                                format dsstdt e8601da.;
                                dsstdt = input(dsstdtc,?? e8601da.);
                        %end;

                        if upcase(dsdecod) = 'DEATH' then order = 100;
                        else order = 1;
                run;

                /* keep one of each kind of disposition event per subject, category, and
subcategory */
                proc sort data=ds_dm;
                        by usubjid
                           dscat
                           dsscat
                           %if &ds_dsstdtc. %then dsstdt;
                           %else %if &ds_dsstdy. %then dsstdy;
                           order
                           %if &ds_dsseq. %then dsseq;
                           dsdecod;
                run;

                data ds_dm
                     ds_dm_pre;
                        set ds_dm;
                        by usubjid dscat dsscat;

                        if upcase(dscat) = 'PROTOCOL MILESTONE'
                        or upcase(dsdecod) =: 'INFORMED CONSENT OBTAINED'
                        or upcase(dsdecod) =: 'RANDOMIZED'
                        then output ds_dm;
                        else if last.dsscat then output ds_dm;
                        else output ds_dm_pre;
                run;

                /* sort by subject and disposition event, removing duplicates, to avoid
double counting */
                proc sort data=ds_dm dupout=ds_dm_dup nodupkey;
                        by usubjid dsdecod;
                run;

        %end;
        %else %do;
```

```
		%global setup_success;
		%let setup_success = 0;

		%if not &dm_subj_gt0. %then %put ERROR: There are no subjects in DM;
		%if not &setup_req_var. %then %put ERROR: Required variables are missing;
	%end;

%mend dm_setup;


/*****************************************/
/* DEMOGRAPHICS INFORMATION FOR VARIABLE VAR */
/*****************************************/
%macro dm(var);

	%put DEMOGRAPHICS: &var.;

	data _null_;
		var = "&var.";
		outds = translate(var,'_','*');
		varlist = translate(var,',','*');
		keylist = substr(var,1,length(trim(var)) -
index(reverse(left(trim(var))),'*'));

		call symputx('outds',outds);
		call symputx('varlist',varlist);
		call symputx('keylist',keylist);
	run;

	proc sql noprint;
		create table dm_&outds. as
		select &varlist.,
			%do di = 1 %to &arm_count.;
		   arm_&di._count label="&&&arm_name_&di. Count",
				100*arm_&di._count/&&&arm_count_&di. as arm_&di._pct
label="&&&arm_name_&di. %",
			%end;
			total_count label='Total Count',
			100*total_count/&total_count. as total_pct label='Total %'
		from (select &varlist.,
			%do di = 1 %to &arm_count.;
				sum(case when arm_num = &di. then 1 else 0 end)
as arm_&di._count,
			%end;
			count(1) as total_count
		from dm
		group by &varlist.)
		order by &keylist., total_count desc;
	quit;

%mend dm;


/*****************************************/
/* DEMOGRAPHIC STATISTICS FOR VARIABLE VAR */
/*****************************************/
%macro dm_stat(ds,var);

	%put DEMOGRAPHIC STATISTICS FOR &var. in &ds.;

	%let outds = %sysfunc(ifc(%sysfunc(index(&ds.,ds)),ds,dm));

	proc sort data=&ds.;
		by %if &outds. = ds %then dsdecod; arm_num;
	run;

	/* get statistics by arm */
	proc univariate data = &ds. noprint;
		var &var.;
		output out=&outds._&var._stat_arm_x
		mean=mean
```

```
                median=median
                min=min
                max=max
                mode=mode
                std=std
                                q1=q1
                q3=q3
                ;
                by %if &outds. = ds %then dsdecod; arm_num;
        run;

        data &outds._&var._stat_arm_x;
                retain %if &outds. = ds %then dsdecod; arm_num mean std mode min q1 median
q3 max;
                set &outds._&var._stat_arm_x;

                /* calculate values used to build the chart in Excel */
                chart_pct25 = q1;
                chart_pct50 = median - q1;
                chart_pct75 = q3 - median;
                chart_min = q1 - min;
                chart_max = max - q3;
        run;

        /* merge statistics onto arm numbers to ensure that every arm has an observation
*/
        data &outds._&var._stat_arm_x;
                merge lkp_arm(in=a keep=arm_num)
                      &outds._&var._stat_arm_x(in=b);
                by arm_num;
                if a;
        run;

        proc transpose data=&outds._&var._stat_arm_x
                        out=&outds._&var._stat_arm
                                        prefix=arm_;
                id arm_num;
                %if &outds. = ds %then by dsdecod;;
        run;

        /* get statistics overall */
        proc univariate data = &ds. noprint;
                var &var.;
                output out=&outds._&var._stat_all_x
                mean=mean
                median=median
                min=min
                max=max
                mode=mode
                std=std
                                q1=q1
                q3=q3
                ;
                by %if &outds. = ds %then dsdecod;;
        run;

        data &outds._&var._stat_all_x;
                retain %if &outds. = ds %then dsdecod; mean std mode min q1 median q3 max;
                set &outds._&var._stat_all_x;

                /* calculate values used to build the chart in Excel */
                chart_pct25 = q1;
                chart_pct50 = median - q1;
                chart_pct75 = q3 - median;
                chart_min = q1 - min;
                chart_max = max - q3;
        run;

        proc transpose data=&outds._&var._stat_all_x

out=&outds._&var._stat_all(rename=(col1=arm_%eval(&arm_count.+1)));
```

```
		%if &outds. = ds %then by dsdecod;;
	run;

	/* combine per-arm statistics and overall statistics */
	data &outds._&var._stat;
		set &outds._&var._stat_arm;
		set &outds._&var._stat_all;
	run;

	/* put the variables in order */
	/* combine mean and standard deviation */
	data &outds._&var._stat(drop=%do i = 1 %to &arm_count.+1; arm_&i._n %end;)
	  &outds._&var._chart(keep=stat %do i = 1 %to &arm_count.+1; arm_&i._n %end;);
		retain %if &outds. = ds %then dsdecod; _name_ _label_ %do i = 1 %to
&arm_count.; arm_&i. %end;;
		set &outds._&var._stat(rename=(%do i = 1 %to &arm_count. + 1;
arm_&i.=arm_&i._n  %end;));

		%do i = 1 %to &arm_count. + 1;
			length arm_&i. $15;
			arm_&i. = compress(arm_&i._n);

			retain mean_&i._n;
			if _name_ = 'mean' then do;
				mean_&i._n = arm_&i._n;
			end;
			else if _name_ = 'std' then do;
				arm_&i. = trim(put(mean_&i._n,8.1 -l))||'
('||trim(put(arm_&i._n,8.1 -l))||')';
			end;

			%if &i. <= &arm_count. %then %do;
				label arm_&i.="&&&arm_name_&i.";
			%end;
			%else %do;
				label arm_&i.='Total';
			%end;

			drop mean_&i._n;
		%end;

		rename arm_%eval(&arm_count.+1) = total;

		if _name_ = 'mean' then delete;
		else do;
			select (_name_);
				when ('std')    _label_ = 'Mean (SE)';
				when ('mode')   _label_ = 'Mode';
				when ('max')    _label_ = 'Max';
				when ('q3')     _label_ = 'Q3';
				when ('median') _label_ = 'Median';
				when ('q1')     _label_ = 'Q1';
				when ('min')    _label_ = 'Min';


				otherwise;
			end;
		end;

		if index(_name_,'chart') then do;
			_label_ = _name_;
			output &outds._&var._chart;
		end;
		else output &outds._&var._stat;

		rename _label_=stat;
		label _label_=' ';
		drop _name_;
	run;

/*	proc datasets library=work nolist nodetails; delete &outds._&var._stat_:; quit;*/
```

```
%mend dm_stat;


/*************************************************************/
/* DEMOGRAPHICS BY DISPOSITION INFORMATION FOR VARIABLE VAR */
/*************************************************************/
%macro dm_by_ds(var);

	%put DEMOGRAPHICS BY DISPOSITION: &var.;

	/* find the counts of demographic subgroups per disposition event and arm */
	proc freq data=ds_dm noprint;
		table &var.*arm_num*dsdecod / out=ds_&var._ct_c(keep=&var. arm_num dsdecod
count) sparse;
	run;

	/* and the counts of demographic subgroups per arm */
	/* and the counts of demographic subgroups */
	proc freq data=dm noprint;
		table &var.*arm_num / out=ds_&var._ct_sac(keep=&var. arm_num count
rename=(count=subgroup_arm_count)) sparse;
		table &var. / out=ds_&var._ct_sc(keep=&var. count
rename=(count=subgroup_count)) sparse;
	run;

	/* merge disposition event counts and subgroup counts and calculate percentage of
subgroup */
	data ds_&var._ct_cp;
		merge ds_&var._ct_c(in=a)
		      ds_&var._ct_sac(in=b);
		by &var. arm_num;

		if subgroup_arm_count ne 0 then percent = 100*count/subgroup_arm_count;
		/*else percent = 0;*/

		drop subgroup_arm_count;
	run;

	proc sort data=ds_&var._ct_cp; by &var. dsdecod arm_num; run;

	/* merge arm counts and percentages onto arm numbers to ensure that every arm has
an observation */
	proc sort data=ds_&var._ct_cp; by arm_num; run;

	data ds_&var._ct_cp;
		merge lkp_arm(in=a keep=arm_num)
		      ds_&var._ct_cp(in=b);
		by arm_num;
		if a;
		/* set missing counts and percentages to zero */
		array num{*} _numeric_;
		do i = 1 to dim(num);
			if a and not b and num(i) = . then num(i) = 0;
		end;
		drop i;
	run;

	proc sort data=ds_&var._ct_cp; by &var. dsdecod; run;

	/* tranpose so arm counts are stored in separate columns */
	proc transpose data=ds_&var._ct_cp out=ds_&var._count(drop=_name_ _label_)
prefix=arm_ suffix=_count;
		by &var. dsdecod;
		id arm_num;
		var count;
	run;

	/* and again for percentages */
	proc transpose data=ds_&var._ct_cp out=ds_&var._percent(drop=_name_) prefix=arm_
suffix=_percent;
```

```
		by &var. dsdecod;
		id arm_num;
		var percent;
	run;

	data ds_&var.;
		merge ds_&var._count(in=a)
		      ds_&var._percent(in=b);
		by &var. dsdecod;
	run;

	/* find total counts and percentages */
	data ds_&var.;
		retain dsdecod &var. %do di = 1 %to &arm_count.; arm_&di._count
arm_&di._percent %end;;
		merge ds_&var.(in=a)
		ds_&var._ct_sc(in=b);
		by &var.;

		total_count = %do di = 1 %to &arm_count; arm_&di._count + %end; 0;
		if subgroup_count ne 0 then total_percent =
100*total_count/subgroup_count;
		/*else total_percent = 0;*/

		drop subgroup_count;
	run;

	proc sort data=ds_&var.; by dsdecod descending total_percent; run;

/*	proc datasets library=work nolist nodetails; delete ds_&var._:; quit;*/

%mend dm_by_ds;


/*****************************/
/* FORMAT DATASETS FOR OUTPUT */
/*****************************/
%macro dm_outfmt;

	%put FORMAT FOR OUTPUT;

	/* sort age datasets by age order */
	/* look up age order from LKP_AGE dataset */
	data dm_age_flag;
		set dm_age_flag;
		if _n_ = 1 then do;
			declare hash h(dataset:'lkp_age');
			h.definekey('age_flag');
			h.definedata('age_order');
			h.definedone();
		end;

		length age_order 8.;
		call missing(age_order);
		rc = h.find();
		drop rc;
	run;

	proc sort data=dm_age_flag out=dm_age_flag(drop=age_order); by age_order; run;

	data ds_age_flag;
		set ds_age_flag;
		if _n_ = 1 then do;
			declare hash h(dataset:'lkp_age');
			h.definekey('age_flag');
			h.definedata('age_order');
			h.definedone();
		end;

		length age_order 8.;
		call missing(age_order);
```

```
		rc = h.find();
		drop rc;
	run;

	proc sort data=ds_age_flag out=ds_age_flag(drop=age_order); by dsdecod age_order;
run;

	/* sort the race dataset, putting other and missing last */
	data dm_race;
		set dm_race;
		if upcase(race) = 'MISSING' then order = 3;
		else if upcase(race ) = 'OTHER' then order = 2;
		else order = 1;
	run;

	proc sort data=dm_race out=dm_race(drop=order); by order race; run;

	/* blank out the category terms on subsequent rows */
	%let fi = 1;
	%let var = %scan(&dm_var.,&fi.,%str( ));
	%do %while (%length(&var.) > 0);
		%if %sysfunc(count(%sysfunc(compress(&var.,*,k)),*)) > 0 %then %do;
			data _null_;
				key =
translate(reverse(substr(reverse("&var"),index(reverse("&var."),'*')+1)),' ','*');
				lastkey = reverse(scan(reverse(key),1));
				call symputx('f_key',key);
				call symputx('f_lastkey',lastkey);
				call symputx('f_var',translate("&var.",'_','*'));
			run;

			data dm_&f_var.;
				set dm_&f_var.;
				by &f_key. notsorted;
				%let fki = 1;
				%let fk_var = %scan(&f_key.,&fki.);
				%do %while (&fk_var. ne );
					if not first.&fk_var. then &fk_var. = '';
					%let fki = %eval(&fki. + 1);
					%let fk_var = %scan(&f_key.,&fki.);
				%end;
			run;
		%end;
		%let fi = %eval(&fi. + 1);
		%let var = %scan(&dm_var.,&fi.,%str( ));
	%end;

	%let fi = 1;
	%let var = %scan(&dm_by_ds_var.,&fi.);
	%do %while (&var. ne );
		data ds_&var.;
			set ds_&var.;
			by dsdecod notsorted;
			if not first.dsdecod then dsdecod = '';

			array num{*} _numeric_;
			do i = 1 to dim(num);
				if num(i) = . then num(i) = 0;
			end;
			drop i;
		run;
		%let fi = %eval(&fi. + 1);
		%let var = %scan(&dm_by_ds_var.,&fi.);
	%end;

	/* make overall tables */
	data dm_overall_stat;
		retain stat %do i = 1 %to &arm_count.; arm_&i. blank_&i. %end; total;
		set dm_age_stat;
		%do i = 1 %to &arm_count.; call missing(blank_&i.); %end;
		if upcase(stat) = 'MODE' then delete;
```

```
	run;

	data dm_overall;
		retain var val;
		set dm_age_flag(in=a)
		    dm_sex(in=b)
			dm_race(in=c)
			dm_ethnic(in=d)
			;

		length var $20 val $100;
		if a then do;
			var = 'Age Group';
			val = age_flag;
		end;
		else if b then do;
			var = 'Sex';
			val = sex;
		end;
		else if c then do;
			var = 'Race';
			val = race;
		end;
		else if d then do;
			var = 'Ethnicity';
			val = ethnic;
		end;
		drop age_flag sex race ethnic;
	run;

	data dm_overall;
		set dm_overall;
		by var notsorted;
		if not first.var then var = '';
	run;

%mend dm_outfmt;


/***********************/
/* DEMOGRAPHICS OUTPUT */
/***********************/
%macro dm_out;

	%put DEMOGRAPHICS OUTPUT TO EXCEL;

	/* general analysis information */;
	data info;
		length val $20 info $100;
		val = 'NDA/BLA'; info = "&ndabla."; output;
		val = 'Study'; info = "&studyid."; output;
		val = 'Date'; info = compbl(put(date(),e8601da.)||'
'||put(time(),timeampm11.)); output;
		val = 'Custom Datasets'; info = /*"&sl_custom_ds."*/''; output;
		val = 'Arm Variable'; info = ifc(&dm_actarm.,'actual treatment arm
(ACTARM)',
                                                     'planned treatment arm (ARM)');
output;
		val = 'Arm Count'; info = "&arm_count."; output;

		/* row counts of the datasets written to Excel */
		val = 'Dm Overall';
		dsid = open('dm_overall'); info =
ifc(dsid,compress(attrn(dsid,'nobs')),'0'); rc = close(dsid); output;
		val = 'Dm Age Group';
		dsid = open('dm_age_flag'); info =
ifc(dsid,compress(attrn(dsid,'nobs')),'0'); rc = close(dsid); output;
		val = 'Dm Sex';
		dsid = open('dm_sex'); info = ifc(dsid,compress(attrn(dsid,'nobs')),'0');
rc = close(dsid); output;
		val = 'Dm Race';
```

```
                dsid = open('dm_race'); info = ifc(dsid,compress(attrn(dsid,'nobs')),'0');
rc = close(dsid);      output;
                val = 'Dm Ethnicity';
                dsid = open('dm_ethnic'); info =
ifc(dsid,compress(attrn(dsid,'nobs')),'0'); rc = close(dsid); output;
                val = 'Dm Country';
                dsid = open('dm_country'); info =
ifc(dsid,compress(attrn(dsid,'nobs')),'0'); rc = close(dsid); output;
                val = 'Dm Site ID';
                dsid = open('dm_siteid'); info =
ifc(dsid,compress(attrn(dsid,'nobs')),'0'); rc = close(dsid); output;
                val = 'Dm Country-Site ID';
                dsid = open('dm_country_siteid'); info =
ifc(dsid,compress(attrn(dsid,'nobs')),'0'); rc = close(dsid); output;

                val = 'Ds Age Group';
                dsid = open('ds_age_flag'); info =
ifc(dsid,compress(attrn(dsid,'nobs')),'0'); rc = close(dsid); output;
                val = 'Ds Sex';
                dsid = open('ds_sex'); info = ifc(dsid,compress(attrn(dsid,'nobs')),'0');
rc = close(dsid); output;
                val = 'Ds Race';
                dsid = open('ds_race'); info = ifc(dsid,compress(attrn(dsid,'nobs')),'0');
rc = close(dsid);      output;
                val = 'Ds Ethnicity';
                dsid = open('ds_ethnic'); info =
ifc(dsid,compress(attrn(dsid,'nobs')),'0'); rc = close(dsid); output;
                val = 'Ds Country';
                dsid = open('ds_country'); info =
ifc(dsid,compress(attrn(dsid,'nobs')),'0'); rc = close(dsid); output;
                val = 'Ds Site ID';
                dsid = open('ds_siteid'); info =
ifc(dsid,compress(attrn(dsid,'nobs')),'0'); rc = close(dsid); output;

                drop dsid rc;

        run;

        %if not %symexist(run_location) %then %let run_location = LOCAL;

        /* local runs use the Microsoft Jet database-based Excel LIBNAME engine */
        %if %upcase(&run_location.) = LOCAL %then %do;
                libname xls excel "&demout." ver=2003;
        %end;
        /* Script Launcher runs use the PCFILES LIBNAME Engine */
        %else %do;
                libname xls pcfiles path="&demout.";
        %end;

        proc datasets library=xls nolist nodetails;
                delete arms
                       age_groups
                       info
                          age_stats
                          dm_overall_stat
                          dm_overall
                          dm_age
                          dm_age_stat
                          dm_sex
                          dm_race
                          dm_ethnicity
                          dm_country
                          dm_siteid
                          dm_country_siteid
                          ds_age
                          ds_sex
                          ds_race
                          ds_ethnicity
                          ds_country
                          ds_siteid
                          ;
```

```
        quit;

        data xls.arms; set lkp_arm_out; run;
        data xls.age_groups; set lkp_age(keep=age_flag where=(upcase(age_flag) ne
'MISSING')); run;
        data xls.info; set info; run;

        data xls.dm_overall_stat; set dm_overall_stat; run;
        data xls.dm_overall; set dm_overall; run;
        data xls.dm_age; set dm_age_flag; run;
        data xls.dm_age_stat; set dm_age_stat; run;
        data xls.age_stats; set dm_age_chart; run;
        data xls.dm_sex; set dm_sex; run;
        data xls.dm_race; set dm_race; run;
        data xls.dm_ethnicity; set dm_ethnic; run;
        data xls.dm_country; set dm_country; run;
        data xls.dm_siteid; set dm_siteid; run;
        data xls.dm_country_siteid; set dm_country_siteid; run;

        data xls.ds_age; set ds_age_flag; run;
        data xls.ds_sex; set ds_sex; run;
        data xls.ds_race; set ds_race; run;
        data xls.ds_ethnicity; set ds_ethnic; run;
        data xls.ds_country; set ds_country; run;
        data xls.ds_siteid; set ds_siteid; run;

        libname xls clear;

%mend dm_out;


%macro demographics;

        %dm_setup;

        %if &setup_success. %then %do;

                /* do all demographic variables */
                %let i = 1;
                %let var = %scan(&dm_var.,&i.,%str( ));
                %do %while (%length(&var.) > 0);

                        %dm(&var.);

                        %let i = %eval(&i. + 1);
                        %let var = %scan(&dm_var.,&i.,%str( ));
                %end;

                /* do all demographic variables by disposition event term */
                %let i = 1;
                %let var = %scan(&dm_by_ds_var.,&i.);
                %do %while (%length(&var.) > 0);

                        %dm_by_ds(&var.);

                        %let i = %eval(&i. + 1);
                        %let var = %scan(&dm_by_ds_var.,&i.);
                %end;

                %dm_stat(dm,age);

                %dm_outfmt;

                /* do preprocessing of Script Launcher datasets */
                %group_subset_pp;

                %dm_out;

                /* create grouping & subsetting output */
                %group_subset_xls_out(gs_file=&demout.);
```

```
	%end;
	%else %do;
		%error_summary(err_file=&errout.,
			err_nosubj=%sysfunc(ifc(&dm_subj_gt0.,0,1)),
			err_missvar=%sysfunc(ifc(&setup_req_var.,0,1))
			);
	%end;

%mend demographics;

%demographics;
```