Topic 0B: Introductions – First SAS Program

```
PROC PRINT DATA=AMY.FGHM113;
RUN;
/* This is a comment */
```

Topic 1B – Importing Data from CSV File

```
PROC PRINT DATA=BIO.PULSECSV;
RUN;
```

Topic 2A – Permanently Labeling Variables

```
/* Notice the x-axis label is PULSE1 - not very descriptive */
PROC SGPLOT DATA=BIO.PULSECSV;
histogram pulse1;
RUN;
/* View the first 5 observations of the original data */
PROC PRINT DATA=BIO.PULSECSV (OBS=5);
RUN;
/* Show information about dataset, including list of variables */
/* VARNUM option: variables listed in order they appear in data */
PROC CONTENTS DATA=BIO.PULSECSV VARNUM;
RUN;
/* Create new dataset with DATA step */
/* SET statement contains old dataset name */
/* LABEL statement provides descriptive labels for variables*/
DATA BIO.PULSE_STEP1;
SET BIO.PULSECSV;
LABEL     HEIGHT         =    "Height (cm)"
          WEIGHT         =    "Weight (kg)"
          AGE            =    "Age (years)"
          GENDER         =    "Gender"
          SMOKES         =    "Regular smoker?"
          ALCOHOL        =    "Regular drinker?"
          EXERCISE       =    "Frequency of exercise"
          TRT            =    "Whether the student ran or sat "
          PULSE1         =    "First pulse measurement (bpm)"
          PULSE2         =    "Second pulse measurement (bpm)"
          YEAR           =    "Year of class";
RUN;
/* View the first 5 observations of the NEW data */
PROC PRINT DATA=BIO.PULSE_STEP1 (OBS=5);
RUN;
```

```
/* Show information about NEW dataset */
PROC CONTENTS DATA=BIO.PULSE_STEP1 VARNUM;
RUN;
/* Histogram for NEW data - now x-axis has descriptive label for variable
*/
PROC SGPLOT DATA=BIO.PULSE_STEP1;
histogram pulse1;
RUN;
```

Topic 2B – Translating Categorical Variables

```
/* View the first 5 observations of the original data */
PROC PRINT DATA=BIO.PULSECSV (OBS=5);
RUN;

/* Show information about dataset, including list of variables */
/* VARNUM option: variables listed in order they appear in data */
PROC CONTENTS DATA=BIO.PULSECSV VARNUM;
RUN;

/* Frequency Tables for Categorical Variables - Without Labels or
Translations */
PROC FREQ DATA=BIO.PULSECSV;
TABLES GENDER SMOKES ALCOHOL EXERCISE TRT;
RUN;

/* PROC FORMAT creates the translations for codes */
/* It DOES NOT apply these translations to variables */
/* One PROC FORMAT can create translations that apply to many datasets */
/* One translation can be applied to many different variables */
/* This procedures just creates the translations which will be needed */
PROC FORMAT;
/* The following is one VALUE statement which defines SAS format GDR */
/* It will eventually be used for the variable GENDER in our data */
VALUE     GDR       1 = "Male"
                    2 = "Female";
/* Now another VALUE statement which defines the SAS format YN */
/* It will be used for the variables SMOKES and ALCOHOL in our data */
VALUE     YN        1 = "Yes"
                    2 = "No";
/* Now we define two more to use for our variables EXERCISE and TRT */
VALUE     EXER      1 = "High"
                    2 = "Moderate"
                    3 = "Low";
VALUE     TREAT     1 = "Ran"
                    2 = "Sat";
RUN;
```

```
/* Create new dataset with DATA step */
/* SET statement contains old dataset name */
/* LABEL statement provides descriptive labels for variables*/
/* FORMAT statement connects formats with variables for translations */
/* It is usually best to try to put all of your data manipulation in one
   DATA step but you may need to create it in steps, running to check as
you go */
DATA BIO.PULSE_STEP2;
SET BIO.PULSECSV;
LABEL     HEIGHT         =    "Height (cm)"
          WEIGHT         =    "Weight (kg)"
          AGE            =    "Age (years)"
          GENDER         =    "Gender"
          SMOKES         =    "Regular smoker?"
          ALCOHOL        =    "Regular drinker?"
          EXERCISE       =    "Frequency of exercise"
          TRT            =    "Whether the student ran or sat "
          PULSE1         =    "First pulse measurement (bpm)"
          PULSE2         =    "Second pulse measurement (bpm)"
          YEAR           =    "Year of class";
/* FORMAT statement can provide format many variables at once */
/* Format name follows variable name or names and ends with a period */
/* The statement doesn't end until the semicolon */
FORMAT    GENDER              GDR.
          SMOKES ALCOHOL      YN.
          EXERCISE            EXER.
          TRT                 TREAT.;
RUN;

/* View the first 5 observations of the NEW data */
/* Notice we now see the translations in the print of the data */
PROC PRINT DATA=BIO.PULSE_STEP2 (OBS=5);
RUN;

/* Show information about NEW dataset */
/* The format names are shown in the variable list table */
PROC CONTENTS DATA=BIO.PULSE_STEP2 VARNUM;
RUN;

/* Frequency Tables for Categorical Variables - With Labels and
Translations */
PROC FREQ DATA=BIO.PULSE_STEP2;
TABLES GENDER SMOKES ALCOHOL EXERCISE TRT;
RUN;
```

Topic 2C – Using a Formatted SAS Dataset

```
/* If we try to use the dataset in any way that would require the
   translations assigned in the earlier FORMAT statement in the DATA
   step, we will get an error!! */
PROC PRINT DATA=BIO.PULSE_STEP2 (OBS=5);
RUN;

/* The dataset has the formats below permanently assigned to certain
   variables. We do NOT need to rerun the DATA step used to assign the
   formats however we DO need to resubmit the PROC FORMAT so SAS will
   know the translations used */
PROC FORMAT;
VALUE     GDR       1 = "Male"
                    2 = "Female";
VALUE     YN        1 = "Yes"
                    2 = "No";
VALUE     EXER      1 = "High"
                    2 = "Moderate"
                    3 = "Low";
VALUE     TREAT     1 = "Ran"
                    2 = "Sat";
RUN;

PROC PRINT DATA=BIO.PULSE_STEP2 (OBS=5);
RUN;

PROC FREQ DATA=BIO.PULSE_STEP2;
TABLES GENDER SMOKES ALCOHOL EXERCISE TRT;
RUN;
```

Topic 4 – Frequency Distributions

```
/* View the first 5 observations of the STEP1 data */
PROC PRINT DATA=BIO.PULSE_STEP1 (OBS=5);
RUN;
/* Show information about STEP1 data - Review variable names */
PROC CONTENTS DATA=BIO.PULSE_STEP1 VARNUM;
RUN;
/* Note: raw data for our categorical variables are coded as numbers */
/* We have not yet created translations for variables in STEP1 Data */
/* We will cover handling that scenario in the translation tutorials */

/* Frequency Distribution for one categorical variable */
PROC FREQ DATA=BIO.PULSE_STEP1;
TABLES GENDER;
RUN;
```

```
/* Frequency Distributions: many categorical variable simultaneously */
/* The PLOTS=FREQPLOT option adds a bar chart to the results */
PROC FREQ DATA=BIO.PULSE_STEP1;
TABLES GENDER SMOKES ALCOHOL EXERCISE TRT / PLOTS=FREQPLOT;
RUN;
/* Bar Charts using SGPLOT */
PROC SGPLOT DATA=BIO.PULSE_STEP1;
VBAR GENDER; /* Vertical Bars */
RUN;
PROC SGPLOT DATA=BIO.PULSE_STEP1;
HBAR GENDER; /* Horizontal Bars */
RUN;
/* In this course, you must have the coded values translated in the final
output. We will cover the FREQ procedure with translations in our
tutorial on that topic. For our 1st assignment, the data of interest is
NOT coded, the raw data is text Notice the difference when you complete
the assignment */
```

Topic 5A – Numeric Measures – PROC MEANS

```
/* View the first 5 observations of the STEP1 data */
PROC PRINT DATA=BIO.PULSE_STEP1 (OBS=5);
RUN;

/* Show information about STEP1 data - Review variable names */
PROC CONTENTS DATA=BIO.PULSE_STEP1 VARNUM;
RUN;

/* Note raw data for our categorical variables are coded as numbers */
/* We have not yet translated the variables in the STEP1 Data */
/* We cover how to handle that scenario in the translation tutorials */
/* Not be an issue here as we focus on only Quantitative variables */

/* Default output for PROC MEANS */
PROC MEANS DATA=BIO.PULSE_STEP1;
VAR HEIGHT;
RUN;

/* Selecting specific values using keywords in PROC MEANS */
/* Documentation:
http://support.sas.com/documentation/cdl/en/proc/67916/HTML/default/viewe
r.htm#n1qnc9bddfvhzqn105kqitnf29cp.htm#n11nn4xu6p9wvjn1lh60wtvvn538 */
PROC MEANS DATA=BIO.PULSE_STEP1 N MEAN STD MIN Q1 MEDIAN Q3 MAX MAXDEC=3;
VAR HEIGHT WEIGHT AGE PULSE1 PULSE2;
RUN;

/* The large table may not copy well, you may prefer the results
   if you run a PROC MEANS for each variable individually */
```

Topic 5B – Creating Histograms and Boxplots

```
/* View the first 5 observations of the STEP1 data */
PROC PRINT DATA=BIO.PULSE_STEP1 (OBS=5);
RUN;
/* Show information about STEP1 data - Review variable names */
PROC CONTENTS DATA=BIO.PULSE_STEP1 VARNUM;
RUN;
/* Note: raw data for our categorical variables are coded as numbers */
/* We have not yet created translations for variables in STEP1 Data */
/* We will cover handling that scenario in the translation tutorials */
/* This will not be an issue here as we focus on only Quantitative
variables */
/* Default output for PROC SGPLOT with HISTOGRAM statement */
PROC SGPLOT DATA=BIO.PULSE_STEP1;
HISTOGRAM HEIGHT;
RUN;
/* Adding Density Curves to histograms with the DENSITY statement */
PROC SGPLOT DATA=BIO.PULSE_STEP1;
  /* We still start with a HISTOGRAM statement */
HISTOGRAM HEIGHT;
  /* Now we add "Best" Normal distribution for this data */
DENSITY HEIGHT / TYPE = NORMAL;
  /* Now we add "Best" Guess at the TRUE distribution for this data */
DENSITY HEIGHT / TYPE = KERNEL;
RUN;
/* Repeat for the rest of our quantitative variables */
/* Each requires a new PROC SGPLOT */
PROC SGPLOT DATA=BIO.PULSE_STEP1;
HISTOGRAM WEIGHT;
DENSITY WEIGHT / TYPE = NORMAL;
DENSITY WEIGHT / TYPE = KERNEL;
RUN;
PROC SGPLOT DATA=BIO.PULSE_STEP1;
HISTOGRAM AGE;
DENSITY AGE / TYPE = NORMAL;
DENSITY AGE / TYPE = KERNEL;
RUN;
PROC SGPLOT DATA=BIO.PULSE_STEP1;
HISTOGRAM PULSE1;
DENSITY PULSE1 / TYPE = NORMAL;
DENSITY PULSE1 / TYPE = KERNEL;
RUN;
PROC SGPLOT DATA=BIO.PULSE_STEP1;
HISTOGRAM PULSE2;
DENSITY PULSE2 / TYPE = NORMAL;
DENSITY PULSE2 / TYPE = KERNEL;
RUN;
```

```
/* Default output for PROC SGPLOT with VBOX statement */
PROC SGPLOT DATA=BIO.PULSE_STEP1;
VBOX HEIGHT;
RUN;
/* Default output for PROC SGPLOT with HBOX statement */
PROC SGPLOT DATA=BIO.PULSE_STEP1;
HBOX HEIGHT;
RUN;
/* I am sure there are numerous interesting options for the
   VBOX and HBOX statements but I don't usually modify these plots */
/* Repeat for the remaining variables using VBOX (my preference) */
PROC SGPLOT DATA=BIO.PULSE_STEP1;
VBOX WEIGHT;
RUN;
PROC SGPLOT DATA=BIO.PULSE_STEP1;
VBOX AGE;
RUN;
PROC SGPLOT DATA=BIO.PULSE_STEP1;
VBOX PULSE1;
RUN;
PROC SGPLOT DATA=BIO.PULSE_STEP1;
VBOX PULSE2;
RUN;
```

Topic 5C – Creating QQ-Plots

```
/* View the first 5 observations of the STEP1 data */
PROC PRINT DATA=BIO.PULSE_STEP1 (OBS=5);
RUN;
/* Show information about STEP1 data - Review variable names */
PROC CONTENTS DATA=BIO.PULSE_STEP1 VARNUM;
RUN;
/* Note: raw data for our categorical variables are coded as numbers */
/* We have not yet created translations for variables in STEP1 Data */
/* We will cover handling that scenario in the translation tutorials */
/* This will not be an issue here as we focus on only Quantitative
variables */
/* Default output for PROC SGPLOT with HISTOGRAM statement */
PROC UNIVARIATE DATA=BIO.PULSE_STEP1 NOPRINT;
VAR HEIGHT;
/* Three different versions of a similar plot which compares
   sample to theoretical normal distribution) - the difference is
   in what exactly is plotted against each other - more agreement
   indicates that the sample is more normally distributed
   We will want the first - QQPLOT */
QQPLOT / NORMAL(MU=EST SIGMA=EST) ; /* Quantile-Quantile plot */
PROBPLOT / NORMAL(MU=EST SIGMA=EST); /* Probability Plot */
PPPLOT; /* Probability-Probability Plot */
```

```
/* You can also create a histograms using this procedure */
/* SAS still puts these first in the output ... */
HISTOGRAM;
HISTOGRAM / NORMAL(MU=EST SIGMA=EST NOPRINT) KERNEL ;
RUN;
/* One nice thing about UNIVARIATE is we can request multiple plots */
PROC UNIVARIATE DATA=BIO.PULSE_STEP1 NOPRINT;
VAR HEIGHT WEIGHT AGE PULSE1 PULSE2;
QQPLOT / NORMAL(MU=EST SIGMA=EST) ;
HISTOGRAM / NORMAL(MU=EST SIGMA=EST NOPRINT) KERNEL ;
RUN;
```

Topic 6A –Two-Way (Contingency) Tables

```
/* The dataset has the formats below permanently assigned to certain
   variables. We do NOT need to rerun the DATA step used to assign the
   formats however we DO need to resubmit the PROC FORMAT so SAS will
   know the translations used */
PROC FORMAT;
VALUE     GDR       1 = "Male"
                    2 = "Female";
VALUE     YN        1 = "Yes"
                    2 = "No";
VALUE     EXER      1 = "High"
                    2 = "Moderate"
                    3 = "Low";
VALUE     TREAT     1 = "Ran"
                    2 = "Sat";
RUN;

/* PROC FREQ is also used to create two-way tables but the command is
   different than for frequency distributions for one variable at a time.
   Here we use a * between variables, multiple requests can be placed in
   One TABLES statement and variables can be grouped in parentheses. The
   variables to the left of the * are the rows and those to the right,
   the columns */
PROC FREQ DATA=BIO.PULSE_STEP2;
TABLES ALCOHOL*EXERCISE (GENDER SMOKES ALCOHOL EXERCISE)*TRT;
RUN;
```

Topic 6B –Two-Way (Contingency) Tables

```
/* The dataset has the formats below permanently assigned to certain
variables. we do NOT need to rerun the DATA step used to assign the
formats however we DO need to resubmit the PROC FORMAT so SAS will know
the translations used */
PROC FORMAT;
VALUE GDR      1 = "Male"
               2 = "Female";
VALUE YN       1 = "Yes"
               2 = "No";
VALUE EXER     1 = "High"
               2 = "Moderate"
               3 = "Low";
VALUE TREAT    1 = "Ran"
               2 = "Sat";
VALUE BMI_TWO  1 = "< 25"
               2 = "25+";
VALUE BMI_FOUR 1 = "< 18.5"
               2 = "[18.5, 25)"
               3 = "[25, 30)"
               4 = "30+";
VALUE WT_TWO   1 = "87kg or Below"
               2 = "More than 87kg";
VALUE WT_FOUR  1 = "55kg or Below "
               2 = "(55kg, 60kg]"
               3 = "(60kg, 67kg]"
               4 = "(67kg, 79kg]"
               5 = "More than 79kg";
RUN;

/* We remove an unusual observation that is likely an error,
   We create a binary and multi-level version of weight,
   and label and format these new variables */
DATA BIO.PULSE_STEP5;
SET BIO.PULSE_STEP4;
/* Remove observation with likely error - very high resting pulse */
IF PULSE1 > 140 THEN DELETE;
/* BINARY WEIGHT */
IF 0 <= WEIGHT <= 87 THEN BinaryWT = 1;
IF 87< WEIGHT <= 120 THEN BinaryWT = 2;
/* Multi-level WEIGHT */
IF 0 < WEIGHT <= 55 THEN WTGroups = 1;
IF 55 < WEIGHT <= 60 THEN WTGroups = 2;
IF 60 < WEIGHT <= 67 THEN WTGroups = 3;
IF 67 < WEIGHT <= 79 THEN WTGroups = 4;
IF 79 < WEIGHT <= 120 THEN WTGroups = 5;
LABEL     BinaryWT = "Binary Weight"
          WTGroups = "Weight Categories";
```

```
FORMAT    BinaryWT WT_TWO.
          WTGroups WT_FOUR.;
RUN;

/* We add three options to our use of PROC FREQ, CHISQ and FISHER to
   obtain p-values for tests and EXPECTED to obtain the expected cell
   counts. Remember that we use a * between variables, multiple requests
   can be placed in one TABLES statement and variables can be grouped in
   parentheses. The variables to the left of the * are the rows and those
   to the right, the columns */
PROC FREQ DATA=BIO.PULSE_STEP5;
TABLES TRT*(GENDER SMOKES ALCOHOL EXERCISE) / CHISQ FISHER EXPECTED;
RUN;

PROC FREQ DATA=BIO.PULSE_STEP5;
TABLES GENDER*(BinaryWT WTGroups) / CHISQ FISHER EXPECTED;
RUN;
```

Topic 7A – Numeric Summaries by Groups

```
/* The dataset has the formats below permanently assigned to certain
   variables. We do NOT need to rerun the DATA step used to assign the
   formats however we DO need to resubmit the PROC FORMAT so SAS will
   know the translations used */
PROC FORMAT;
VALUE     GDR       1 = "Male"
                    2 = "Female";
VALUE     YN        1 = "Yes"
                    2 = "No";
VALUE     EXER      1 = "High"
                    2 = "Moderate"
                    3 = "Low";
VALUE     TREAT     1 = "Ran"
                    2 = "Sat";
RUN;

/* PROC MEANS is also used to create numerical summaries by groups
   Only one grouping variable can be defined but multiple quantitative
   variables can be requested in the VAR statement */
PROC MEANS DATA=BIO.PULSE_STEP2 N MEAN STD MIN Q1 MEDIAN Q3 MAX MAXDEC=3;
CLASS GENDER;  /* Categorical variable - defines groups */
VAR PULSE1;    /* Quantitative variable or variables to be summarized */
RUN;

PROC MEANS DATA=BIO.PULSE_STEP2 N MEAN STD MIN Q1 MEDIAN Q3 MAX MAXDEC=3;
CLASS SMOKES;
VAR PULSE1;
RUN;
```

```
PROC MEANS DATA=BIO.PULSE_STEP2 N MEAN STD MIN Q1 MEDIAN Q3 MAX MAXDEC=3;
CLASS ALCOHOL;
VAR PULSE1;
RUN;
PROC MEANS DATA=BIO.PULSE_STEP2 N MEAN STD MIN Q1 MEDIAN Q3 MAX MAXDEC=3;
CLASS EXERCISE;
VAR PULSE1;
RUN;
PROC MEANS DATA=BIO.PULSE_STEP2 N MEAN STD MIN Q1 MEDIAN Q3 MAX MAXDEC=3;
CLASS TRT;
VAR PULSE1;
RUN;
```

Topic 7B – Side-By-Side Boxplots

```
/* The dataset has the formats below permanently assigned to certain
   variables. We do NOT need to rerun the DATA step used to assign the
   formats however we DO need to resubmit the PROC FORMAT so SAS will
   know the translations used */
PROC FORMAT;
VALUE     GDR       1 = "Male"
                    2 = "Female";
VALUE     YN        1 = "Yes"
                    2 = "No";
VALUE     EXER      1 = "High"
                    2 = "Moderate"
                    3 = "Low";
VALUE     TREAT     1 = "Ran"
                    2 = "Sat";
RUN;

/* PROC SGPLOT and VBOX or HBOX are also used to create boxplots by
   groups We must add one option of CATEGORY to define the grouping
   variable As always, we must request each plot with a separate SGPLOT
   procedure */

/* Boxplots of PULSE1 by GENDER */
PROC SGPLOT DATA=BIO.PULSE_STEP2;
VBOX PULSE1 / CATEGORY = GENDER;
RUN;

/* Boxplots of PULSE1 by SMOKES */
PROC SGPLOT DATA=BIO.PULSE_STEP2;
VBOX PULSE1 / CATEGORY = SMOKES;
RUN;
```

```
/* Boxplots of PULSE1 by ALCOHOL */
PROC SGPLOT DATA=BIO.PULSE_STEP2;
VBOX PULSE1 / CATEGORY = ALCOHOL;
RUN;

/* Boxplots of PULSE1 by EXERCISE */
PROC SGPLOT DATA=BIO.PULSE_STEP2;
VBOX PULSE1 / CATEGORY = EXERCISE;
RUN;

/* Boxplots of PULSE1 by TRT */
PROC SGPLOT DATA=BIO.PULSE_STEP2;
VBOX PULSE1 / CATEGORY = TRT;
RUN;
```

Topic 7C – Two Independent Samples T-test

```
/* The dataset has the formats below permanently assigned to certain
variables. We do NOT need to rerun the DATA step used to assign the
formats however we DO need to resubmit the PROC FORMAT so SAS will know
the translations used */
PROC FORMAT;
VALUE GDR      1 = "Male"
               2 = "Female";
VALUE YN       1 = "Yes"
               2 = "No";
VALUE EXER     1 = "High"
               2 = "Moderate"
               3 = "Low";
VALUE TREAT    1 = "Ran"
               2 = "Sat";
VALUE BMI_TWO  1 = "< 25"
               2 = "25+";
VALUE BMI_FOUR 1 = "< 18.5"
               2 = "[18.5, 25)"
               3 = "[25, 30)"
               4 = "30+";
VALUE WT_TWO   1 = "87kg or Below"
               2 = "More than 87kg";
VALUE WT_FOUR  1 = "55kg or Below "
               2 = "(55kg, 60kg]"
               3 = "(60kg, 67kg]"
               4 = "(67kg, 79kg]"
               5 = "More than 79kg";
RUN;
```

```
/* PROC TTEST can be used for
    (1) One Sample T-tests (not shown here)
    (2) Paired T-tests (not shown here)
    (3) Two Independent Samples T-tests (THIS TUTORIAL) */
/* Here we want to compare the mean resting pulse between those
   who ran and those who sat to see if there are any differences
   in these two treatment groups */
PROC TTEST DATA=BIO.PULSE_STEP5;
CLASS TRT;   /* Binary Explanatory Variable */
VAR PULSE1;  /* Quantitative Response Variable */
RUN;

/* Numeric Summaries of PULSE1 by TRT */
/* We add the CLM option for confidence intervals for the mean in each
group */
PROC MEANS DATA = BIO.PULSE_STEP6 N MEAN STD MIN Q1 MEDIAN Q3 MAX CLM;
CLASS TRT;
VAR PULSE1;
RUN;

/* Side-by-side boxplots */
PROC SGPLOT DATA = BIO.PULSE_STEP6;
VBOX PULSE1 / CATEGORY = TRT;
RUN;

/* Let's look at another comparison - mean weight between males and
females */
PROC TTEST DATA=BIO.PULSE_STEP5;
CLASS GENDER;   /* Binary Explanatory Variable */
VAR WEIGHT;    /* Quantitative Response Variable */
RUN;

/* Numeric Summaries of WEIGHT by GENDER */
/* We add the CLM option for confidence intervals for the mean in each
group */
PROC MEANS DATA = BIO.PULSE_STEP6 N MEAN STD CLM;
CLASS GENDER;
VAR WEIGHT;
RUN;

PROC MEANS DATA = BIO.PULSE_STEP6 MIN Q1 MEDIAN Q3 MAX;
CLASS GENDER;
VAR WEIGHT;
RUN;

/* Side-by-side boxplots */
PROC SGPLOT DATA = BIO.PULSE_STEP6;
VBOX WEIGHT / CATEGORY = GENDER;
RUN;
```

Topic 7D – One-Way ANOVA (Analysis of Variance)

```
/* The dataset has the formats below permanently assigned to certain
variables. We do NOT need to rerun the DATA step used to assign the
formats however we DO need to resubmit the PROC FORMAT so SAS will know
the translations used */
PROC FORMAT;
VALUE GDR      1 = "Male"
               2 = "Female";
VALUE YN       1 = "Yes"
               2 = "No";
VALUE EXER     1 = "High"
               2 = "Moderate"
               3 = "Low";
VALUE TREAT    1 = "Ran"
               2 = "Sat";
VALUE BMI_TWO  1 = "< 25"
               2 = "25+";
VALUE BMI_FOUR 1 = "< 18.5"
               2 = "[18.5, 25)"
               3 = "[25, 30)"
               4 = "30+";
VALUE WT_TWO   1 = "87kg or Below"
               2 = "More than 87kg";
VALUE WT_FOUR  1 = "55kg or Below "
               2 = "(55kg, 60kg]"
               3 = "(60kg, 67kg]"
               4 = "(67kg, 79kg]"
               5 = "More than 79kg";
RUN;

/* PROC GLM can be used for One-way ANOVA (Analysis of Variance) */
/* This procedure can also perform more complex ANOVA models     */
PROC GLM DATA=BIO.PULSE_STEP5;
CLASS WTGROUPS;   /* Categorical Explanatory Variable */
/* MODEL statement is MODEL Y = X */
/* The quantitative response is on the left of the equals */
/* The categorical explanatory variable is on the right */
MODEL HEIGHT = WTGROUPS;
LSMEANS WTGROUPS / adjust= TUKEY ;
LSMEANS WTGROUPS / adjust= BON ;
RUN;
QUIT;

/* Numeric Summaries of HEIGHT by WTGROUPS */
/* We add the CLM option for confidence intervals for the mean in each
group */
```

```
PROC MEANS DATA = BIO.PULSE_STEP6 N MEAN STD CLM;
CLASS WTGROUPS;
VAR HEIGHT;
RUN;

PROC MEANS DATA = BIO.PULSE_STEP6 MIN Q1 MEDIAN Q3 MAX;
CLASS WTGROUPS;
VAR HEIGHT;
RUN;

/* Side-by-side boxplots */
PROC SGPLOT DATA = BIO.PULSE_STEP6;
VBOX HEIGHT / CATEGORY = WTGROUPS;
RUN;

/* Next Analysis of AGE vs WTGROUPS*/
PROC GLM DATA=BIO.PULSE_STEP5;
CLASS WTGROUPS;
MODEL AGE = WTGROUPS;
LSMEANS WTGROUPS / adjust= TUKEY ;
LSMEANS WTGROUPS / adjust= BON ;
RUN;
QUIT;

/* Numeric Summaries of AGE by WTGROUPS */
/* We add the CLM option for confidence intervals for the mean in each
group */
PROC MEANS DATA = BIO.PULSE_STEP6 N MEAN STD CLM;
CLASS WTGROUPS;
VAR AGE;
RUN;

PROC MEANS DATA = BIO.PULSE_STEP6 MIN Q1 MEDIAN Q3 MAX ;
CLASS WTGROUPS;
VAR AGE;
RUN;

/* Side-by-side boxplots */
PROC SGPLOT DATA = BIO.PULSE_STEP6;
VBOX AGE / CATEGORY = WTGROUPS;
RUN;
```

Topic 7E – Non-Parametric Tests for Case CQ

```
/* The dataset has the formats below permanently assigned to certain
variables. We do NOT need to rerun the DATA step used to assign the
formats however we DO need to resubmit the PROC FORMAT so SAS will know
the translations used */
PROC FORMAT;
VALUE GDR      1 = "Male"
               2 = "Female";
VALUE YN       1 = "Yes"
               2 = "No";
VALUE EXER     1 = "High"
               2 = "Moderate"
               3 = "Low";
VALUE TREAT    1 = "Ran"
               2 = "Sat";
VALUE BMI_TWO  1 = "< 25"
               2 = "25+";
VALUE BMI_FOUR 1 = "< 18.5"
               2 = "[18.5, 25)"
               3 = "[25, 30)"
               4 = "30+";
VALUE WT_TWO   1 = "87kg or Below"
               2 = "More than 87kg";
VALUE WT_FOUR  1 = "55kg or Below "
               2 = "(55kg, 60kg]"
               3 = "(60kg, 67kg]"
               4 = "(67kg, 79kg]"
               5 = "More than 79kg";
RUN;

/* Wilcoxon-Rank Sum Test (also known as Mann-Whitney U-test) */
/* Non-parametric alternative to Two-Sample T-Test  */
PROC NPAR1WAY DATA=BIO.PULSE_STEP5 WILCOXON;
CLASS GENDER;   /* Categorical Explanatory Variable */
VAR HEIGHT;     /* Quantitative Response Variable */
RUN;
QUIT;

/* Interestingly, we still use the WILCOXON option to request the
   Kruskal-Wallis test - SAS chooses based upon th number of levels
   of the categorical explanatory variable */
PROC NPAR1WAY DATA=BIO.PULSE_STEP5 WILCOXON;
CLASS WTGROUPS;   /* Categorical Explanatory Variable */
/* We can ask for multiple response variables simultaneously */
VAR HEIGHT AGE;  /* Quantitative Response Variable List */
RUN;
QUIT;
```

Topic 8A – Setting up Data for Paired T-test

```
/* The dataset has the formats below permanently assigned to certain
variables. We do NOT need to rerun the DATA step used to assign the
formats however we DO need to resubmit the PROC FORMAT so SAS will know
the translations used */
PROC FORMAT;
VALUE GDR       1 = "Male"
                2 = "Female";
VALUE YN        1 = "Yes"
                2 = "No";
VALUE EXER      1 = "High"
                2 = "Moderate"
                3 = "Low";
VALUE TREAT     1 = "Ran"
                2 = "Sat";
VALUE BMI_TWO   1 = "< 25"
                2 = "25+";
VALUE BMI_FOUR  1 = "< 18.5"
                2 = "[18.5, 25)"
                3 = "[25, 30)"
                4 = "30+";
VALUE WT_TWO    1 = "87kg or Below"
                2 = "More than 87kg";
VALUE WT_FOUR   1 = "55kg or Below "
                2 = "(55kg, 60kg]"
                3 = "(60kg, 67kg]"
                4 = "(67kg, 79kg]"
                5 = "More than 79kg";
RUN;

/* Create differences between Pulse 2 and Pulse 1 for each observation */
DATA BIO.PULSE_STEP6;
SET BIO.PULSE_STEP5;
/* Calculate difference between pulse after vs. before */
DIFF_2v1 = PULSE2 - PULSE1;
/* Label the new variable */
LABEL DIFF_2v1 = "DIFF (Pulse2 - Pulse1)";
RUN;

/* Check Data */
PROC PRINT DATA = BIO.PULSE_STEP6 (OBS=10);
/* Print only the variables of interest */
/* Allows to print in order you wish */
VAR TRT PULSE2 PULSE1 DIFF_2V1;
RUN;

PROC CONTENTS DATA = BIO.PULSE_STEP6 VARNUM;
RUN;
```

```
/* Side-by-side Boxplots of differences by treatment */
PROC SGPLOT DATA=BIO.PULSE_STEP6;
VBOX DIFF_2V1 / CATEGORY = TRT;
RUN;
```

Topic 8B – Exploratory Analysis of Difference

```
/* The dataset has the formats below permanently assigned to certain
variables. We do NOT need to rerun the DATA step used to assign the
formats however we DO need to resubmit the PROC FORMAT so SAS will know
the translations used */
PROC FORMAT;
VALUE GDR      1 = "Male"
               2 = "Female";
VALUE YN       1 = "Yes"
               2 = "No";
VALUE EXER     1 = "High"
               2 = "Moderate"
               3 = "Low";
VALUE TREAT    1 = "Ran"
               2 = "Sat";
VALUE BMI_TWO  1 = "< 25"
               2 = "25+";
VALUE BMI_FOUR 1 = "< 18.5"
               2 = "[18.5, 25)"
               3 = "[25, 30)"
               4 = "30+";
VALUE WT_TWO   1 = "87kg or Below"
               2 = "More than 87kg";
VALUE WT_FOUR  1 = "55kg or Below "
               2 = "(55kg, 60kg]"
               3 = "(60kg, 67kg]"
               4 = "(67kg, 79kg]"
               5 = "More than 79kg";
RUN;
```

```
/* Numeric Summaries of DIFFERENCES by TRT */
/* We add the CLM option for confidence intervals for the mean in each
group */
PROC MEANS DATA = BIO.PULSE_STEP6 N MEAN STD CLM;
CLASS TRT;
VAR DIFF_2V1;
RUN;
```

```
PROC MEANS DATA = BIO.PULSE_STEP6 MIN Q1 MEDIAN Q3 MAX;
CLASS TRT;
VAR DIFF_2V1;
RUN;

/* QQ-plots for DIFFERENCES for each TRT */
/* WHERE statement allows you to select a portion of the observations
   based upon the condition specified - here we look at TRT = 1 */
PROC UNIVARIATE DATA = BIO.PULSE_STEP6 NOPRINT;
WHERE TRT=1;
QQPLOT DIFF_2V1 / NORMAL(MU=EST SIGMA=EST);
RUN;

/* Here we use TRT = 2 in the WHERE statement*/
PROC UNIVARIATE DATA = BIO.PULSE_STEP6 NOPRINT;
WHERE TRT=2;
QQPLOT DIFF_2V1 / NORMAL(MU=EST SIGMA=EST);
RUN;

/* Side-by-Side Boxplots */
PROC SGPLOT DATA=BIO.PULSE_STEP6;
VBOX DIFF_2V1 / CATEGORY = TRT;
RUN;

/* Here you might also want the individual boxplots */
/* The WHERE statement can again accomplish this task */
TITLE "TREATMENT = RAN";
PROC SGPLOT DATA=BIO.PULSE_STEP6;
WHERE TRT=1;
VBOX DIFF_2V1 ;
RUN;
TITLE; /* CLEAR THE TITLE! */

TITLE "TREATMENT = SAT";
PROC SGPLOT DATA=BIO.PULSE_STEP6;
WHERE TRT=2;
VBOX DIFF_2V1 ;
RUN;
TITLE; /* CLEAR THE TITLE! */
```

Topic 8C – Paired T-test and Non-Parametric Alternatives

```
/* The dataset has the formats below permanently assigned to certain
variables. We do NOT need to rerun the DATA step used to assign the
formats however we DO need to resubmit the PROC FORMAT so SAS will know
the translations used */
```

```
PROC FORMAT;
VALUE GDR      1 = "Male"
               2 = "Female";
VALUE YN       1 = "Yes"
               2 = "No";
VALUE EXER     1 = "High"
               2 = "Moderate"
               3 = "Low";
VALUE TREAT    1 = "Ran"
               2 = "Sat";
VALUE BMI_TWO  1 = "< 25"
               2 = "25+";
VALUE BMI_FOUR 1 = "< 18.5"
               2 = "[18.5, 25)"
               3 = "[25, 30)"
               4 = "30+";
VALUE WT_TWO   1 = "87kg or Below"
               2 = "More than 87kg";
VALUE WT_FOUR  1 = "55kg or Below "
               2 = "(55kg, 60kg]"
               3 = "(60kg, 67kg]"
               4 = "(67kg, 79kg]"
               5 = "More than 79kg";
RUN;

/* Paired T-Test using PROC TTEST and the PAIRED statement
   using the original pulse measurements - among only those who RAN */
PROC TTEST DATA = BIO.PULSE_STEP6 ;
WHERE TRT=1; /* Selects the current group - those who RAN */
PAIRED PULSE2*PULSE1 ; /* Think of * as subtraction - strange but true*/
RUN;

/* We can also base the test on the differences we calculated */
/* One-Sample T-Test using PROC TTEST for the differences we
   calculated - among only those who RAN*/
/* The H0 option gives the hypothesized value from the
   Null Hypothesis */
/* Similar code can be used for one-sample t-test */
PROC TTEST DATA = BIO.PULSE_STEP6 H0=0;
WHERE TRT=1; /* Selects the current group - those who RAN */
VAR DIFF_2v1 ;
RUN;

/* One-Sample T-Test using PROC UNIVARIATE for the differences we
   calculated - among only those who RAN*/
/* The MU0 option gives the hypothesized value from the
   Null Hypothesis */
/* Similar code can be used for one-sample t-test */
/* This code automatically gives the results for the
   Sign test and the Wilcoxon Signed-Rank test. */
PROC UNIVARIATE DATA = BIO.PULSE_STEP6 MU0=0 CIBASIC;
```

```
WHERE TRT=1; /* Selects the current group - those who RAN */
VAR DIFF_2v1 ;
QQPLOT / NORMAL(MU=EST SIGMA=EST) ;
RUN;

/* Calculate the numeric summaries of the differences for those who ran
*/
PROC MEANS DATA = BIO.PULSE_STEP6 N MEAN STD CLM;
WHERE TRT=1;
VAR DIFF_2v1 ;
RUN;

PROC MEANS DATA = BIO.PULSE_STEP6 MIN Q1 MEDIAN Q3 MAX;
WHERE TRT=1;
VAR DIFF_2v1 ;
RUN;

PROC SGPLOT DATA=BIO.PULSE_STEP6;
WHERE TRT=1;
VBOX DIFF_2V1 / CATEGORY = TRT;
RUN;
```

Topic 9A – Basic Scatterplots

```
/* The dataset has the formats below permanently assigned to certain
variables. We do NOT need to rerun the DATA step used to assign the
formats however we DO need to resubmit the PROC FORMAT so SAS will know
the translations used */
PROC FORMAT;
VALUE     GDR       1 = "Male"     2 = "Female";
VALUE     YN        1 = "Yes"      2 = "No";
VALUE     EXER      1 = "High"     2 = "Moderate"      3 = "Low";
VALUE     TREAT     1 = "Ran"      2 = "Sat";
RUN;

/* PROC SGPLOT */
/* SCATTER statement will produce a simple scatterplot */
PROC SGPLOT DATA=BIO.PULSE_STEP2;
SCATTER Y = WEIGHT X = HEIGHT;
RUN;

/* LOESS statement will produce a scatterplot with an extra LOESS line
   This gives a smooth curve through the data. T he SMOOTH option must
   be greater than 0. The larger the value, the more smoothing. Usually
   you want to find a balance - you want to see any non-linear patterns
   but you don't want to want to try to fit the data exactly */
/* Values can be data dependent so some trial and error may be useful */
/* Too Little Smoothing */
PROC SGPLOT DATA=BIO.PULSE_STEP2;
LOESS Y = WEIGHT X = HEIGHT / SMOOTH=0.1;
RUN;

/* Too Much Smoothing */
PROC SGPLOT DATA=BIO.PULSE_STEP2;
LOESS Y = WEIGHT X = HEIGHT / SMOOTH=1.2;
RUN;

/* Reasonable Smoothing */
PROC SGPLOT DATA=BIO.PULSE_STEP2;
LOESS Y = WEIGHT X = HEIGHT / SMOOTH=0.5;
RUN;
PROC SGPLOT DATA=BIO.PULSE_STEP2;
LOESS Y = WEIGHT X = HEIGHT / SMOOTH=0.6;
RUN;

/* Default smoothing */
PROC SGPLOT DATA=BIO.PULSE_STEP2;
LOESS Y = WEIGHT X = HEIGHT;
RUN;
```

Topic 9B – Grouped Scatterplots

```
/* The dataset has the formats below permanently assigned to certain
variables. We do NOT need to rerun the DATA step used to assign the
formats however we DO need to resubmit the PROC FORMAT so SAS will know
the translations used */
PROC FORMAT;
VALUE     GDR       1 = "Male"
                    2 = "Female";
VALUE     YN        1 = "Yes"
                    2 = "No";
VALUE     EXER      1 = "High"
                    2 = "Moderate"
                    3 = "Low";
VALUE     TREAT     1 = "Ran"
                    2 = "Sat";
RUN;

/* PROC SGPLOT */
/* SCATTER statement will produce a scatterplot
   the GROUP option defines a categorical variable
   which varies the colors of points */
```

```
PROC SGPLOT DATA=BIO.PULSE_STEP2;
SCATTER Y = WEIGHT X = HEIGHT / GROUP = GENDER;
RUN;

/* LOESS statement can also use the GROUP option, here we
   look at SMOOTH = 0.5 and the default smoothing */

/* Reasonable Smoothing */
PROC SGPLOT DATA=BIO.PULSE_STEP2;
LOESS Y = WEIGHT X = HEIGHT / GROUP = GENDER SMOOTH = 0.5;
RUN;

PROC SGPLOT DATA=BIO.PULSE_STEP2;
LOESS Y = WEIGHT X = HEIGHT / GROUP = GENDER;
RUN;
```

Topic 9C – Pearson’s Correlation Coefficient

```
/* The dataset has the formats below permanently assigned to certain
variables. We do NOT need to rerun the DATA step used to assign the
formats however we DO need to resubmit the PROC FORMAT so SAS will know
the translations used */
PROC FORMAT;
VALUE     GDR       1 = "Male"
                    2 = "Female";
VALUE     YN        1 = "Yes"
                    2 = "No";
VALUE     EXER      1 = "High"
                    2 = "Moderate"
                    3 = "Low";
VALUE     TREAT     1 = "Ran"
                    2 = "Sat";
RUN;

/* PROC CORR */
PROC CORR DATA=BIO.PULSE_STEP2;
VAR HEIGHT WEIGHT;
RUN;
```

Topic 9D – Simple Linear Regression - EDA

```
/* The dataset has the formats below permanently assigned to certain
variables. We do NOT need to rerun the DATA step used to assign the
formats however we DO need to resubmit the PROC FORMAT so SAS will know
the translations used */
PROC FORMAT;
VALUE     GDR       1 = "Male"
                    2 = "Female";
VALUE     YN        1 = "Yes"
                    2 = "No";
VALUE     EXER      1 = "High"
                    2 = "Moderate"
                    3 = "Low";
VALUE     TREAT     1 = "Ran"
                    2 = "Sat";
RUN;

/* PROC REG */
PROC REG DATA=BIO.PULSE_STEP2;
MODEL WEIGHT = HEIGHT;
RUN;
QUIT;
```

Topic 9E – Simple Linear Regression - Inference

```
/* The dataset has the formats below permanently assigned to certain
variables. We do NOT need to rerun the DATA step used to assign the
formats however we DO need to resubmit the PROC FORMAT so SAS will know
the translations used */
PROC FORMAT;
VALUE GDR      1 = "Male"
               2 = "Female";
VALUE YN       1 = "Yes"
               2 = "No";
VALUE EXER     1 = "High"
               2 = "Moderate"
               3 = "Low";
VALUE TREAT    1 = "Ran"
               2 = "Sat";
VALUE BMI_TWO  1 = "< 25"
               2 = "25+";
VALUE BMI_FOUR 1 = "< 18.5"
               2 = "[18.5, 25)"
               3 = "[25, 30)"
               4 = "30+";
RUN;
```

```
/* We now add a few skills to our PROC REG code*/
/* The PLOTS=DIAGNOSTICS(UNPACK) will provide the 9 graphs
   individually instead of together in the panel */
/* the CLB option provides confidence intervals for
   the parameters (estimated intercept and estimated slope) */

PROC REG DATA=BIO.PULSE_STEP4 PLOTS=DIAGNOSTICS(UNPACK);
MODEL WEIGHT = HEIGHT / CLB ;

RUN;
```