Section 1. Clean the Data

1. Three %LET statements were used to create macro variables **mydir, in, out** to hold the paths for the directory containing input folder and output results.
2. A LIBNAME statement was used to create a library **input** in the directory **&in.**
3. A PROC FORMAT VALUE statement was used to create formats for character variables **Runaway, Income, Children, Child_support, Grade_level, Grades, ADHD, Gender, Ethnicity**.
4. A DATA step was used to create a dataset **demog**, an external file `&in\demographics.csv` was identified to the DATA step using a INFILE statement (DSD TRUNCOVER FIRTSTOBS=2), and subsequent INPUT statement codifies the format and move the data for all variables into SAS.
    a. A PROC MEANS statement was used with **demog** dataset, the statistics **mean, min, max** showed possible errors for **Birthdate, Survey_Year, Gender, Ethnicity** variables.
    b. Subsequent frequency tables were used for the above four variables.
    c. For **Birthdate** variable, and the value of 11/8/990 and 12/18/989 were replaced with 11/8/1990 and 12/18/1989 using an **MDY()** function respectively because it was assumed the 1 was accidentally omitted.
    d. For **Survey_Year** variable, the value of 7 was replaced with 2007 because it was assumed the 200 was accidentally omitted.
    e. **Gender** takes meaningful values in (-9, -8, 1, 2). Therefore, the value of 3 was replaced with -9, assuming a negative sign was accidentally omitted.
    f. **Ethnicity** takes meaningful values in (1, 2, 3, 4). Thus, the value of 0 was replaced with a missing value.
    g. A FORMAT statement was used to permanently associate a format with variables **Birthdate** (mmddyy10.)**, Gender, Ethnicity.**
    h. A LABEL statement was used to associate labels with all variables**.**
5. A DATA step was used to create a dataset **bkgrd_1**, and a subsequent SET statement is to process the existing SAS data set `input.background_part1` as input for the DATA step.

a. A INPUT(COMPRESS()) function was used with **Brothers** variable to first remove any blanks, and then convert the character variable to a numeric character **Bro.**

b. A LABEL statement was used to associate a label with **Bro** variable.

c. In the DATA statement, DROP **Brothers** from output, and RENAME **Bro** to **Brothers.**

6. A DATA step was used to create a dataset **bkgrd_2**, and a subsequent SET statement is to process the existing SAS data set `input.background_part2` as input for the DATA step.

   a. A INPUT(COMPRESS())function was used with **Self_regulation** variable to first remove any blanks, and then convert the character variable to a numeric character **Self_reg.**

   b. A LABEL statement was used to associate a label with **Self_reg** variable.

   c. In the DATA statement, DROP **Self_regulation** from output, and RENAME **Self_reg** to **Self_regulation** .

7. APPEND datasets **bkgrd_1** and **bkgrd_2** into a new dataset **bkgrd** using a SET statement.

   a. An ARRAY and an IF-THEN statement were used to convert all negative values in numeric variables **_numeric_** to missing values.

   b. A PROC MEANS statement was used with **bkgrd** dataset, the statistics **mean, min, max** showed possible errors for variables **Runaway, Child_support, Grade_level .**

   c. Subsequent frequency tables were used for the above three variables.

   d. **Runaway** takes meaningful values in (1, 2, 3). Thus, the value of 0 was replaced with a missing value.

   e. **Child_support** takes meaningful values in (0, 1). Values > 1 were replaced with 1.

   f. **Grade_level** takes meaningful values in (1-7). Values greater than 7 were replaced with 7.

   g. **Age_first_arrest** < **Age_first_offense** isn't reasonable, use IF-THEN to set them equal.

   h. **Friends** = 1000 is unreasonable. Recode it to the second maximum number 50.

   i. **Detention_jail** = 101 is unreasonable. Recode it to the second maximum number 10.

   j. A FORMAT statement was used to permanently associate a format with variables **Runaway, Income, Children, Child_support, Grade_level, Grades, ADHD.**

k. In the DATA statement, RENAME **caseid** to **ID** in the output for later merging**.**

8. PROC SORT datasets **demog** and **bkgrd** by **ID** variable for further merging.

9. A DATA step was used to create a dataset **clean**.

   a. A MERGE statement was used to merge datasets **demog** and **bkgrd**.

   b. IN=a and IN=b dataset options and an IF statement were used to keep the observations have both **demog** and **bkgrd** data.

   c. A frequency table was used for **ID** variable. A duplicate observation with **ID** = 54411 and a missing **ethnicity** value was then deleted using an IF-THEN statement.

   d. A PROC PRINT VAR statement was used to display all variables from the dataset **clean** and determine their order in the report.

Section 2. Output an RTF File

1. A PROC MEANS statement was used to analyze summary statistics for the dataset **clean** where **grades** were the CLASS variable and **emotionality** was the independent variable. An OUTPUT OUT= statement was then used to output the statistics to a dataset **output**. Also, rename these summary statistics.
2. PROC SORT the datasets **output** by descending **_type_** and ascending **grades** variables.
3. A DATA step was used to create a dataset **summary.** Read data from the dataset **output.** A BY descending **_type_ grades** statement was used with an IF-THEN statement to let **grades** take a value of -1 if **_type_** = 0. According to the PROC FORMAT VALUE part in Section 1., **grades** taking a value of -1 is formatted to output "Overall".
4. ODS options CENTER NODATE NONUMBER NOPROCTITLE ESCAPECHAR were used to control the output. Turn ODS TRACE ON.
5. A one-way ANOVA was performed on dataset **clean** where **grades** variable was the CLASS variable and **emotionality** = **grades** was the MODEL. An OUTSTAT statement was used to output the parameter estimates to a dataset **est**. QUIT the procedure.
6. An ODS RTF statement was used to create a `&out\OneWay_Emotionality_Grades.rtf`
    a. A STARTPAGE=YES option was used to let a later boxplot display on a new page.
    b. A BODYTITLE option was used to let the title display as a bodytitle other than a header.
    c. A PROC PRINT statement was used with the dataset **summary**.
    d. SYTLE() options were used to customize the font, color, and style for all variables.
    e. A LABEL statement was used to associate labels with all variables.
    f. A TITLE statement was used to create a title. Its font, color, and style were customized.
    g. RUN to end the PROC PRINT procedure, and start a new line for the following text content to display.
    h. A TITLE statement was used to clean all titles.
    i. An ODS TEXT statement was used to add a text. Its font, color, and style were customized.

The **F** and **p-value** were read from the output of dataset **est.**

j. A same one-way ANOVA was again performed. An ODS SELECT BOXPLOT statement was used to output the generated graph to the RTF file. QUIT the procedure.

7. Close ODS RTF files, turn ODS TRACE OFF, and QUIT.

Section 3. Create a MACRO

1. A %MACRO Macro Statement was used to create a macro **oneway** to soft code Section_2. The macro **oneway** takes two arguments: **DEP** (the dependent variable) and **IND** (the independent variable).

2. A CRL+H shortcut was used to find all text “emotionality” and “grades”, then replace them with the macro variables “**&dep.**” and “**&ind.**”, respectively.

3. Copy the code in Section_2 into Section_3.

4. A DATA step was used to create a null dataset **_null_** to define macro variables.

5. Read data from the dataset **est** output from the ANOVA test.

6. IF **_type_** = “error”, THEN a CALL SYMPUTX() statement was used to convert **df** variable to a macro variable **&df2**, which represents the second degree of freedom.

7. IF **_type_** != “error”, A CALL SYMPUTX() statement was used to convert **df** variable to a macro variable **&df1**, which represents the first degree of freedom.

A SUM statement was used to assign a new variable **k** with a value of **df** + 1, which represents the level.

8. Three CALL SYMPUTX() statements were used to convert **k**, ROUND(**F**, 0.01), PUT(**prob**,pvalue9.4) variables to macro variables **&k., &f., &p_val.**, respectively. END IF_THEN procedure.

9. IF prob < 0.05, THEN a CALL SYMPUTX() statement was used to convert a character **string** “was” to a macro variable **&sgfnt.** ELSE a CALL SYMPUTX() statement was used to convert a character **string** “was not” to the macro variable **&sgfnt.**

10. These macro variables were then used to replace their corresponding parts in the conclusion text in the ODS TEXT statement.

11. A %MEND macro Statement was used to end the macro **oneway** after ODS TRACE OFF.

12. Use **%oneway(DEP=,IND=)** statements to run the macro with designated arguments.