**Basic Programming with R**

The files you need to submit are your **R script and this document in PDF**. Paste in your code and output into this document where it is specifically requested. If directions are given but nothing is asked directly, an “OK” response will suffice. Name your files following this convention:

epid799b\_hw1\_lastname.pdf (Homework document with answers filled in)

epid799b\_hw1\_lastname.r (R script).

For this assignment, 10 points will be taken away for failing to follow all instructions.

**Readability.** Both your graders, and in “real” life, your colleagues and future-you will appreciate greatly your well-written, structured and documented code. In your R script, include a header for the name of the assignment, programmer name and etc., and use comments in your code to indicate the task the set of commands are for. Comment liberally! Question 2 on this assignment explicitly asks you to structure your code, but in the future these directions will not be explicit.

You may need certain packages that are not pre-installed. If needed, install the packages first, and load the packages by using the command **library(*functionname*)**.

Complete the follow steps in R/RStudio:

1. **Read the file (10 points)**This assignment will use the dataset births.csv from Sakai.
   1. Read in your births.csv file into a data.frame called “births,” as we have done in class. This time, use the “stringsAsFactors” parameter set to F or FALSE to avoid R automatically coding character strings as factors. We’ll cover more of this later. (5 points)
   2. Print out the first few rows. What function did you use? (5 points)
   3. **Optional Challenge**: Read in any or all of the existing SAS, STATA and/or tab delimited versions of the file directly into R. You will need outside packages and other functions besides read.csv(). Report what packages you used and how the data.frames compare to the one read through read.csv().
2. **Comments and code structure: (20 points)**Now that you’ve got the code loaded, let’s think ahead and plan to write well-structured code. Show you can use these organization and commenting techniques:
   1. Include a descriptive header. (5 points)
   2. Include “comment bars” for visual breaks. (5 points)
   3. Include descriptive comments, both at the start of a line and after a line of runnable code. (10 points)
   4. **Optional Challenge**: Organize your code into collapsible code blocks (not covered in class: google to find out how, or explore the RStudio keyboard shortcut menu)
3. **Basic operations on datasets. (20 points)**Using some or all of the functions: summary, head, tail, str, names, dim…
   1. Report the number of records and fields in the dataset. Pick three variables and report their type (numeric, character, etc.). (10 points)
   2. Report the five-number summary for the WKSGEST and MDIF variables. (10 points)
4. **Subsetting, loading packages & making graphics. (20 points)**
   1. Using the selection operators [], create a smaller version of the births file called births\_sample with only the first 1000 rows and with only these variables:"MAGE", "MDIF", "VISITS", "WKSGEST", "MRACE". (10 points)
   2. Plot this smaller dataset all at once using the base plot(births\_sample) function. Paste the plot in here. What does this plot show? (10 points)
   3. **Optional Challenge**: Install and load the package “car”, and plot the smaller dataset using scatterplotMatrix(). If you are familiar with ggplot2, you might try the ggpairs() function in the GGally package. If you’re comfortable with functions, consider using the sample() function to get a random sample for births\_sample, instead of the first 1000 rows.
5. **Basic operations on data columns. (30 points)**Returning to the original, full births dataset and recode the below variables. Note that we will learn faster ways to do many of these steps, but for now, use what you know.
6. Maternal Age: (10 points)
   1. Create a table of MAGE, and paste it here.
   2. What does MAGE=99 mean, specifically? See the code book.
   3. Assign MAGE=99 to R’s “missing” value.
   4. Create and paste in a univariate graph of MAGE. Consider using boxplot, density plot [hint: plot(density(x))] or histogram.
   5. Create a centered mage\_centered variable in the dataset, equal to the MAGE minus the mean of MAGE. Report its fivenumber summary.
7. Prenatal Care: (10 points)
   1. Create a table of MDIF and paste it here
   2. What do MDIF 88 and MDIF 99 mean?
   3. Assign 88 and 99 to NA.
   4. Create a univariate plot for MDIF and paste it in.
8. Cigarette Use: (10 points)
   1. Recode the existing cigdur character variable to a new integer variable smoke so that it is coded as 1 for “Y”, 0 for “N”, and missing otherwise.
   2. Make sure your recoding is right by creating a two-way frequency table between the new variable smoke (row variable) and the old variable cigdur (column variable). Paste the output here.
9. **Project** (Pass/Fail)  
   Settle on a dataset and plan for your class project. As a reminder (and you can read the project guidelines in the syllabus): **This project is for you**! If you are new to R you may benefit from replicating an analysis you’ve done in another language. If you have coded in R in the past, you may wish to start by translating an old project and spend the bulk of your time in additional, more advanced analyses. Or you may wish to start from scratch with a new analysis. If you are interested in hearing about other datasets, reach out to the TAs.
   1. Briefly describe the background of your project. Briefly! In three bullet points if possible.
   2. What is the objective of your project?
   3. Tell us a little about your dataset: approximate number of records and size, data types and cleanliness…whatever makes sense to you.