**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 and email them to Mike:

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

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

**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 3 on this assignment explicitly asks you to structure your code, but in the future these directions will not be explicit. Consider reviewing the style guides referenced in class (snake case or camel case?) at least once before you start writing code.

This is largely a class about learning to code in R… not also to perfectly elucidate epidemiologic concepts or on scientific writing. Please feel free to write short-hand, use incomplete sentences, or give one-word answers.

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. **Comments and code structure**
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 throughout the following questions:
	1. Include a descriptive header and your research question for the overall analysis (put this as a #comment in your code header).
	2. Include “comment bars” for visual breaks.
	3. Include descriptive comments, both at the start of a line and after a line of runnable code.
	4. Include at least one comment with a relevant, clickable web reference
	5. Organize your code into collapsible code blocks
	6. Make your first code block loading basic libraries and setting your working directory with setwd().

1. **Read the file**
This assignment will use the dataset births.csv from Sakai.
	1. Read in the “full” births2012.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. (*In practice you will likely use* tidyverse::read\_csv *but in this case use* base::read.csv)
	2. Print out the first few rows. What function did you use?
	3. **Optional Challenge**: Read [Strings as Factors: An Authorized Biography](https://simplystatistics.org/2015/07/24/stringsasfactors-an-unauthorized-biography/) and [StringsAsFactors = \*sigh\*.](http://notstatschat.tumblr.com/post/124987394001/stringsasfactors-sigh)
2. **Basic operations on datasets**
Using all of these functions in in code: summary, head, tail, str, names, dim.
	1. Create a code chunk called “Exploring Data”
	2. Report the number of records and fields in the dataset.
	3. Report the five-number summary for the WKSGEST and MDIF variables.
	4. Pick three variables we will use for our analysis and report their type (numeric, character, etc.).
3. **Subsetting, loading packages & making graphics**
	1. In the same “Exploring” code chunk, 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".
	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?
	3. **Optional Challenges**: 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.
4. **Recoding …. other basic operations on data columns.**
Return to the full dataset (not the head or sample set from Q4). Recode the following variables we plan to use in our main analysis. Note that we will learn faster ways to do many of these steps, but for now, use what you know. We’ll be coding integers, factors, and dates.
Suggestion: start a new section called “Recoding.”
5. Prenatal Care
	1. Create a table of mdif and paste it here
	2. What do mdif 88 and mdif 99 mean?
	3. Assign 99 to R’s missing value (and 88, eventually, to no early PNC).
	4. Create a relevant univariate plot for mdif and paste it in.
	5. Create a numeric variable pnc5 that is a 1 if month prenatal care began is less than or equal to 5, 0 if later, and **NA** if missing.
	6. Create a factor variable pnc5\_f based on pnc5 with labels "No Early PNC" and "Early PNC"
	7. Check your result with a bivariate table of pnc5\_f and mdif (useNA=”always” will be relevant here). Paste it in here.
6. Preterm Birth
	1. Create a histogram of wksgest and paste it here.
	2. Create an integer variable called preterm that is a 1 if wksgest is at least 20 weeks but less than 37 weeks, 0 if at least 37 weeks, and missing if missing.
	3. Create a factor variable preterm\_f based on the preterm integer variable following the same coding as preterm, labeling the right levels as “term” and “preterm.”
	4. **Optional Challenge**: Create a factor variable preterm\_f in one step, based on the original wksgest integer variable following the same coding as preterm using the cut() function, labeling the right levels as “term” and “preterm.”
7. Plurality
	1. We’ll be using plurality as a selection critieria. Create a table of plur to confirm it is coded properly.
8. Maternal Age
	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 five number summary.
9. Cigarette Use
	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. Convert that integer variable smoke to a factor variable smoker\_f with levels “Smoker” and “Non-smoker”
	3. Make sure your recoding is right by creating a two-way frequency table between the new variable smoke\_f (row variable) and the old variable cigdur (column variable). Paste the output here.
10. Date of Birth
	1. Find the lubridate package’s github page and skim it
	2. Run ?lubridate and vignette(“lubridate”) to explore the package
	3. Load the lubridate (or tidyverse!) package at the top of your script (suggestion: a chunk called “packages & data”)
	4. Use helpful functions like lubridate::ymd() to convert the dob string variable into a date-typed version called dob\_d
11. Sex
	1. Using skills from the previous questions, code the factor variable sex\_f as “M”, “F”, or missing. Use a table to confirm your recoding.
12. Maternal Race / Ethnicity

**First, an important note on race-ethnicity**!

We will be following some traditional public health conventions in coding race and ethnicity together and treating it as a categorical variable. However, deep, appropriate treatment of race and ethnicity is anything but simple, and individuals we assign to these groupings may not agree with their grouped assignment in this study.

Race and ethnicity are powerful social constructs with real impacts whose nuanced meanings – through individual identification, other ascribing, “passing” as one race/ethnicity or another, etc. – change over space and time. Quite literally a person of similar phenotype and genealogy may consider by themselves White non-Hispanic in North Carolina but may not have if they were born in Texas, or in South America, or may not “pass” as their life-long identified race-ethnicity in different places. Some ethnicities have been selectively absorbed into the “White” race while others have been excluded, and these treatments have changed, even in the US context, over time and in both directions.

It is important to reflect on the data generation process for all variables, but perhaps especially identifies like race-ethnicity, in your dataset. In this dataset, mothers (or their nurses) may self- or other-assigned race-ethnicity when filling out the dataset. We would expect most mothers would agree with the race-ethnicity data on their form, but discrepancies may exist.

The groupings we will propose simplify the real and experienced multi-racial and multi-ethnic identities that may more accurately describe individuals or groups. Deconstructing race-ethnicity is beyond the scope of this class, but should you regularly analyze data including race-ethnicity, we recommend explicitly seeking guidance on these constructs from books, classes and community trainings.

Further, race and ethnicity are inherently tied to questions of power and prejudice, and these constructs are not only between people, but multi-level (internalized within a person, directly interpersonal, institutional, and cultural) and operate through time (historical trauma, centuries of disenfranchisement or advantage, etc.). Individual race-ethnicity often stands in for these many complicated constructs. There are community groups and trainings one can attend to better understanding race and anti-racism in a power context, which we believe directly informs research design, analysis and interpretation of questions related to race-ethnicity.
**Overall**: It has been years (but not so many!) since dominant scientific culture conceptualized race-ethnicity as primarily a biological phenomenon rather than predominantly a social one with social and biological impacts. Be wary in any public health analysis or interpretation, whether explicitly or implicitly, that treats race and ethnicity like chiefly genetic, inherent, unchanging, or biological construct.

*That said, let’s code race-ethnicity:*
	1. Use a bivariate table() to investigate the covariate relationships of maternal race and ethnicity.
	2. Create a raceeth\_f factor variable from mrace and methnic for these (limited!) categories: White non-Hispanic, White Hispanic, African American (any methnic value), American Indian or Alaska Native (any methnic value) and Other (including unknown or any missing values).

	**Note** that there are many elegant ways to do this more complicated recoding, only [some of which](https://stackoverflow.com/questions/31156957/use-of-switch-in-r-to-replace-vector-values) we have the ability to use now. Some that you may try, in rough order of increasing complexity/elegance, include:
		* + Traditional conditional subsetting, line by line
			+ Nested ifelse() statements
			+ The recode() function in the dplyr package
			+ Create a helper data frame and using merge() on race and ethnicity together
			+ Vectorize() switch() function
			+ The sapply() function with switch()
	3. Create one or more tables (*note table(a,b,c) is valid syntax*) to confirm coding of the race-ethnicity variable
13. **Optional Challenges**: Useful early Packages
	1. Use the tableone:: package to print out a Table 1 of some of your variables of interest
	2. Use the mice:: package to print out misingness patterns
	3. Use ggplot2:: package with ggpairs() to print a plot matrix of a sample() (for time/size reasons) of your data