**Creating & graphing summary / exploratory datasets with dplyr and ggplot**

Once again, the files you need to submit are your **R script and this document in word or PDF by 11:59pm on the due date**. 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 (where X is the number of the HW) and email them to Mike:

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

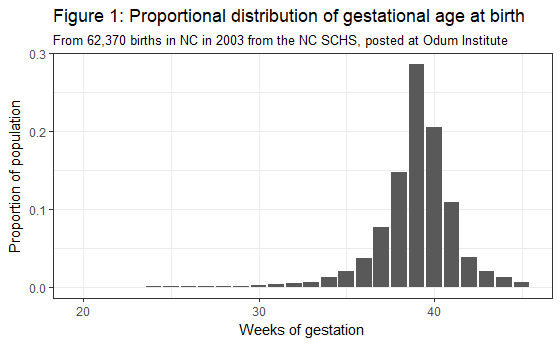
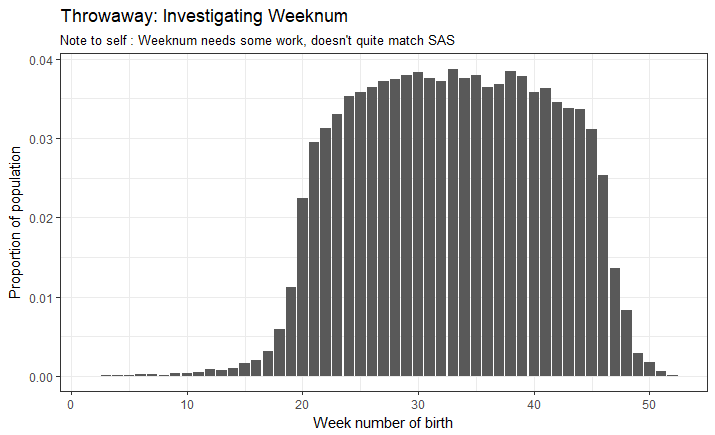
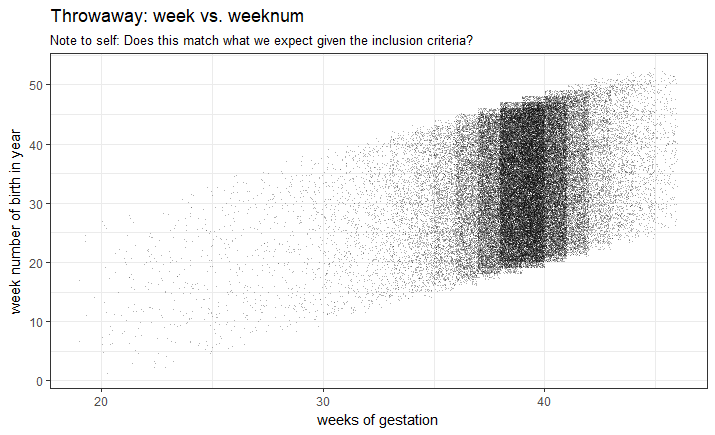
epid799b\_hwX\_lastname.r (R script).

**Readability.** A reminder, please, for your sake and the instructor’s, make efforts to comment, create code blocks, and write legible code. However, on answering questions in this document, please write as little as you can – short answers and sentence fragments are encouraged.

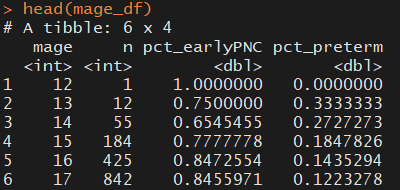
**Recoding**. Homeworks 2 through 5 extend on the preceding homework, so please build this onto the foundation build previously (e.g. recoding, etc.). If necessary, update your preceding homework sections with the released answers.

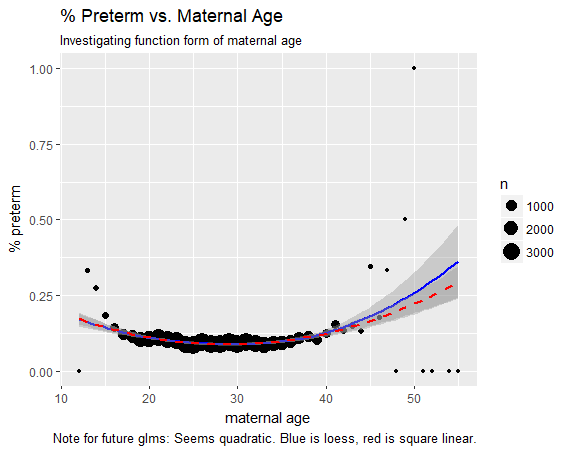
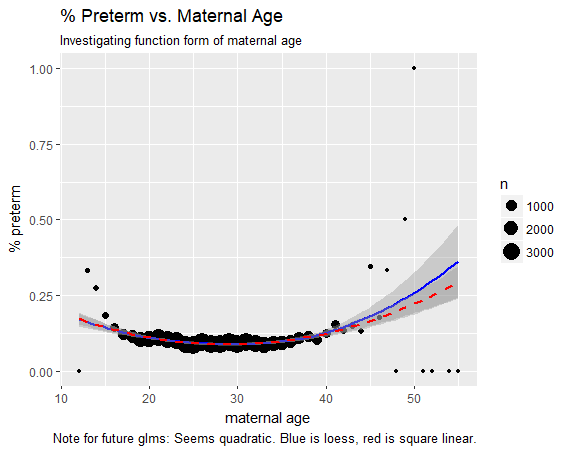
Complete the follow steps in R/RStudio:

# Part 1: Create some tables and graphs describing the new dataset and ensuring the inclusion criteria worked

1. **Mimic Graphs:** Mimic these graphs as best you can using ggplot2 and paste them in below each graph. If they aren’t exact, that’s fine! Until we know how to use dplyr thoroughly to create summary data, you’ll likely need to use the ..prop and ..count hidden variables (as covered in ggplot2 class) to create these graphs. These are tough concepts; take your time and don’t be afraid to help each other!
   1. 
   2. 
   3. 
   4. **Change a graph from questions a, b or c using ggplot’s “+” syntax to add a layer**. You can do anything you like, but as a suggestion, try taking a saved plot c and adding geom\_bin2d(binwidth=1, alpha=0.8).
   5. **Create your own!** Create and submit any ggplot2 graphic you like built from the births dataset and using our variables of interest. Use at least two geometry layers, a title, a subtitle, and a caption that describes your interpretation of the punchline of your graph.   
      Note that we do not yet have tools to dramatically transform the dataframe (one more week!), so you may or may not need to transform your data in order to make your planned graph.

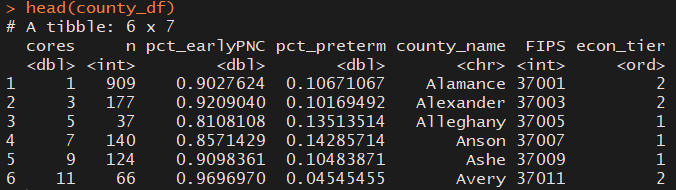
# Part 2: Functional form of mage with dplyr

1. **Create a summary table for maternal age:** Using dplyr, create a summary table for mage from the births dataset whose head looks like below: 
2. **Explore functional form**
   1. Create a ggplot similar to the below plot that demonstrates the functional form of maternal age to the risk of preterm birth. Note that the “weight” aesthetic will be important so that the larger center bins appropriately pull most of the quick loess model.
   2. **Optional Challenge:** Consider the “on the fly” linear square term model as a functional form – if you want to attempt it, check the help page for geom\_smooth, looking at the method and formula parameters. Also check out [UCLA's IDRE](https://stats.idre.ucla.edu/r/faq/how-can-i-explore-different-smooths-in-ggplot2/) help pages.



# Part 3: County-specific stories

1. **Create a County Summary Table:** Using similar dplyr techniques, we’ll create a county table. In this case we’ll also read in a helper table and do some work with it.
   1. Read in *birth format helper 2012.csv.* Using base R we’ve learned previously, perform these tasks to convert it to a table called county\_data:
      1. Filter to just include rows where variable is cores
      2. Rename the columns to variable, cores, county\_name and FIPS
      3. Drop the “variable” column
      4. Covert cores to a number if it is a character (check with str()!)
   2. Using dplyr and pipes, complete the same tasks to create county\_data from format\_helper in “one line” (even if you indent it). You’ll need the filter(), rename(), select() and mutate() functions.
   3. Download and read in county\_tiers.csv into a county\_tiers data.frame from the NC Department of Commerce (<https://www.nccommerce.com/research-publications/incentive-reports/county-tier-designations>). For convenience, the csv is available on the course website.
      1. What type of variable is econ\_tier ?
      2. convert econ-tier from integer to a factor using ordered() or factor() with the right parameters.
   4. In one dplyr statement, create *county\_df* data.frame that has its head() like below. You’ll need group\_by(), summarize() (with n()!), and two left\_join().



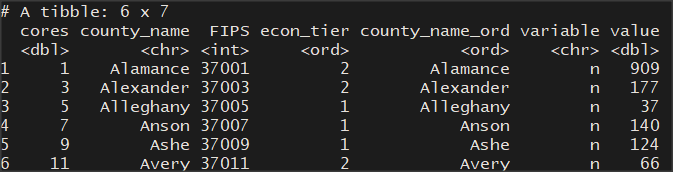
* 1. Write the data.frame you’ve created to a local file called “county\_birth\_summary.csv” We may use it later (e.g. to build a map) .

1. **Create a ggplot** (and paste in here) that describes the relationship between the percent of our exposure, outcome, and economic tier of the county. There are many ways to do this!
   1. **Challenge**: you might consider using plotly package to “hand” explore this interesting summary dataset.

1. **dplyr gather() and ordered factors.**

*Note, this question is just a few lines of R, but definitely a challenge. This is not the best way to map these aesthetics to answer this question; the point of this question is to challenge you to use ordered factors with dplyr:: gather (for later input as facets in ggplot).*

* 1. Create an ordered factor county\_name\_ord that’s ordered by the pct\_earlyPNC variable (this factor will specify the plot order below).
  2. Use dplyr gather() to create a data.frame that turns our variables n, pct\_earlyPNC and pct\_preterm into variables where the name is in one column and the value in another. See the head() below to understand the goal of gather for this question.



* 1. Use this newly created variable to facet your ggplot and produce something similar to below.

