

Problem Sets Prior to Test 1

Only turn in problems that are **not** bracketed. Bracketed problems are additional problems you can look at. Round brackets indicate problems that may help you with problems that are assigned; square brackets are additional problems on material that you should know, but you are not required to write up solutions; curly brackets are truly optional and may contain extra nuggets that you will not be required to know but may be interested in.

Additional assignments will be filled in over time.

notation	meaning
unbracketed	assigned problem – turn these in for grading
()	helper/warm-up problem
[]	additional problems (you are responsible for content, but don't turn them in)
{}	covers optional material

PS	Due	Source	Problems
0	Wed 2/1	handout	<p>1 Fill out personal information form.</p> <p>2 Visit the course web page</p> <ul style="list-style-type: none"> • http://www.calvin.edu/~rpruim/courses/s341/S17/ <p>3 Login to RStudio at http://rstudio.calvin.edu</p> <p>You should have recieved an email letting you know how to login and how to change your password.</p>
1	Wed 2/1	Rethinking 2	(2M4) cards 2M5 cards 2M7 cards
2	Mon 2/6	Rethinking 2	2E1–2E3 conditional probability 2H1–2H4 pandas
3	Wed 2/8	Rethinking 2 Rethinking 3	(2M1) grid 2M2 grid 2M3 earth or moon (3E1–3E7) posterior samples 3M1–3M3 globe [3M5]

PS	Due	Source	Problems
4	Mon 2/13 Wed 2/15	Rethinking 3 Rethinking 4	<p>3H1–3H5 <small>births</small></p> <p>4E1–4E2 <small>describing models</small> 4M1 <small>simulating prior</small> 4M2 <small>map</small></p> <p>I'm not particularly fond of how the author sets things up in 3H1–3H5. Here's a better way:</p> <pre> # load the birth1 and birth2 data vectors data(homeworkch3, package = "rethinking") # put them into a data frame Birth <- data_frame(first = birth1, second = birth2) # tally up the counts tally(~ first + second, data = Birth, margins = TRUE) ## second ## first 0 1 Total ## 0 10 39 49 ## 1 30 21 51 ## Total 40 60 100 # another way to summarize: Birth %>% # group by family type group_by(first, second) %>% summarise(# how many families of this type families = n(), # total boys in such families boys = sum(first + second), # total girls in such families girls = sum(2 - first - second)) ## Source: local data frame [4 x 5] ## Groups: first [?] ## ## first second families boys girls ## <dbl> <dbl> <int> <dbl> <dbl> ## 1 0 0 10 0 20 ## 2 0 1 39 39 39 ## 3 1 0 30 30 30 ## 4 1 1 21 42 0 </pre> <p>Take a look at the code I've posted online for a way to use something other than 0's and 1's. In any case, this summaries above are all you need to do the problems.</p>
5	Fri 2/17	Rethinking 4	4E4–4E5 4M3–4M6

PS	Due	Source	Problems
6	Mon 2/20	Rethinking 4	4H1 4H2 <small>Howell kids</small> 4H3 <small>log-kg</small>