Stat 021 Homework 2

Suzanne Thornton

Due: Fri, Sept 20, 12:00pm

Instructions: A **pdf** version of your homework must be submitted to Gradescope by **noon** on the due date. The course passcode is **MPKJ4Z**. If you are having trouble getting your *.Rmd* file to compile, you need to get help with this **before** the due date.

You are allowed to hand in **only one** late homework assignment throughout the semester. If you need to hand in this particular assignment late, you must let me know via email by noon on the due date.

You are encouraged to study with your peers to help complete the homework assignments but no copying is allowed. If I see that two or more homework assignments are copied, all students involved will receive a grade of 0 on that assignment and will forfeit (perhaps retroactively) the opportunity to hand in a late homework.

Q 1) Create a non-normal data set of sample size $n = 20$ by setting a random seed and drawing your observations from a Beta distribution with $\alpha = 5$ and $\beta = 1.23$.(5 points)

a) Create a tibble (with one column) of your simulated data. Print the first few rows of this tibble using the *head()* function.
b) Test if this data looks like it comes from a Normally distributed population using a Normal probability plot. Comment on the behavior of the data and what the deviations from Normality look like.
c) Create 24 other data sets each of sample size $n = 20$, and each also drawn from a Beta distribution with $\alpha = 5$ and $\beta = 1.23$. It might be useful to create a matrix of size 20×25 and then break this apart into individual data sets, just make sure the end result is a *tibble*. For each of these 25 data sets, compute the mean (check out the *colMeans* function) and store these 25 means in a new tibble called *beta_means*. Print the first few rows of this tibble, *beta_means*.
d) Now test if the data in *beta_means* looks like it comes from a Normally distributed population using a Normal probability plot as in part (b). Does this data look like it's Normally distributed? Why or why not?

Hint: Make sure you set the random number seed in R so that your results are reproducible. (The command for that is *set.seed(100)* although you can use any number you want, not just 100. Look at the R documentation on the R function *rbeta()*.)

Q 2) Read the following article: https://fivethirtyeight.com/features/science-isnt-broken and watch this John Oliver clip: https://youtu.be/0Rnq1NpHdmw (some adult language content). (5 points)

a) Briefly define p-hacking and researcher degrees of freedom.

b) Why should you not believe a finding from any single scientific study? When should you consider a finding to be reliable?

c) What are replication studies? Why are they rare?

Q 3) The data file "vapor_pressure.txt" contains the vapor pressure (in mm Hg) of water for various temperatures (in deg K). Look at the data file before you try to do anything with it! There is an extra line of information between the variable names and the data itself. You'll need to figure out how to make sure that R does not include this extra line as part of your data.

a) Read the data into R and produce a scatter plot of vapor pressure by temperature. Does it seem likely that a straight-line model will be adequate to describe what you see? (4 pts)

b) Fit a simple linear regression model with *vapor_pressure* as the predictor variable and *temperature* as the response variable. Plot the regression line over a scatter plot and identify the intercept and the slope of the model. What are your conclusions regarding model adequacy? (2 pts)

c) From physical chemistry the Clausius-Clapeyron equation states that, for p_v = vapor pressure and T = temperature,

$$ln(p_v) \propto -1/T.$$

Create a data set that contains the *vapor_pressure* data but has an additional column corresponding to the natural logarithm of the values of *vapor_pressure*. Now repeat part b using temperature and this new variable. Are your conclusions regarding model adequacy different from those in part (b)? Why or why not? (4 pts)

Hint: The natural logarithm function in base R is simply *log()*.