Problem Set 3

Stats 506, Fall 2018

Due: Monday November 19, 5pm

Instructions

- Submit the assignment by the due date via canvas. If you intend to utilize late days, please upload partial progress to Canvas and comment that you will utilize late days *before* the assignment is due. In your comment please indicate how many days you intend to use.
- Use Rmarkdown to create and submit a single html or pdf with your answers to each question along with supporting evidence in the form of tables and graphs.
- All tables and graphs should be neatly labeled and appear polished.
- All question ask you to use *R*. You should submit your code for each problem as `ps3_qX.R`.
- You should submit a single compressed archive (`.zip`) which contains the following files:
  - `ps3.pdf` or `ps3.html`
  - `ps3.Rmd`
  - `ps3_q1.R`
  - `ps3_q2.R`
  - `ps3_q3.R` (Optionally)

  All files should be executable without errors.
- All files read, sourced, or referred to within scripts should be assumed to be in the same working directory (`./`).
- Your code should be clearly written and it should be possible to assess it by reading it. Use appropriate variable names and comments. Your style will be graded using the style rubric (./StyleRubric.html) [15 points].
- Some of these exercises may require you to use commands or techniques that were not covered in class or in the course notes. You can use the web as needed to identify appropriate approaches. Part of the purpose of these exercises is for you to learn to be resourceful and self sufficient. Questions are welcome at all times, but please make an attempt to locate relevant information yourself first.

Question 1 [60 points]

First, repeat question 3 parts a-c from problem set 1 using `data.table` for all computations and data manipulations.

Then, formulate and state a question answerable using the RECS data. Your question should be similar in scope to (one of) parts a-c above and should rely on one or more variables not previously used. Answer your question (using data.table) and provide supporting evidence in the form of nicely formatted graphs and/or tables.

Question 2 [25 points]

In this question you will design a Monte Carlo study in R to compare the performance of different methods that adjust for multiple comparisons (https://xkcd.com/882/). You can read more about each of these methods by referring to `help(p.adjust)` in R and the references listed there.

Throughout this question, let $n = 1000$, $p = 100$ and

$$\beta_i = \begin{cases} 1 & i \in \{1, \ldots, 10\}, \\ 0 & \text{else.} \end{cases}$$

Let $X \in \mathbb{R}^{n \times p}$ with $X \sim N(0_p, \Sigma)$ and $Y \sim N(X\beta, \sigma^2 I_n)$ where I_n is an $n \times n$ identify matrix and Σ is a $p \times p$, symmetric, positive definite covariance matrix.

a. Write a function that accepts matrices `X` and `beta` and returns a `p` by `mc_rep` matrix of p-values corresponding to p-values for the hypothesis tests:

$$H_0 : \beta_i = 0, H_1 : \beta_i \neq 0.$$

In addition to `X` and `beta` your function should have arguments `sigma` (σ) and `mc_rep` controlling the error variance of Y and number of Monte Carlo replicates, respectively. Your function should solve the least-squares problems using the QR decomposition of $X'X$. This decomposition should only be computed once each time your function is called.

i. Refer to the course notes to find $\hat{\beta}$.

ii. Use Y and $\hat{Y} = X\hat{\beta}$ to estimate the error variance for each Monte Carlo trial m:

$$\hat{\sigma}^2_m = \frac{1}{n-p} \sum_{i=1}^{n} (Y_{im} - \hat{Y}_{im})^2$$

iii. Use the result from ii and the QR decomposition to find the variance of $\hat{\beta}_i$, $v_i = \hat{\sigma}^2 (X'X)^{-1}_{ii}$. [Note: you will need to do some algebra to determine how to compute $(X'X)^{-1}$ using Q and R. Or you can use the function `chol2inv()`.]

iv. Form $Z_i = \hat{\beta}_i / \sqrt{v_i}$ and find $p = 2(1 - \Phi^{-1}(|Z_i|))$.

Test your function with a specific X and Y by comparing to the output from appropriate methods applied to the object returned by `lm(Y ~ 0 + X)`. It's okay if there is some finite precision error less than ~1e-3 in magnitude. Hint: use `set.seed()` to generate the same Y inside and outside the scope of the function for the purpose of testing.

b. Choose Σ and σ as you like. Use the Cholesky factorization of Σ to generate a single X. Pass X, β, and σ to your function from the previous part.

c. Write a function `evaluate` that takes your results and a set of indices where $\beta \neq 0$, and returns Monte Carlo estimates for the following quantities:

- The family wise error rate
- The false discovery rate
- The sensitivity
- The specificity.

See this page (https://en.wikipedia.org/wiki/Sensitivity_and_specificity#Sensitivity_index) for additional details.

d. Apply your function from the previous part to the matrix of uncorrected P-values generated in part B. Use the function `p.adjust()` to correct these p-values for multiple comparisons using 'Bonferroni', 'Holm', 'BH' (Benjamini-Hochberg), and 'BY' (Benjamini-Yekuteli). Use your `evaluate()` function for each set of adjusted p-values.

e. Produce one or more nicely formatted graphs or tables reporting your results. Briefly discuss what you found.

Question 3 (Optional) [30 points]

This is a bonus question related to problem 6 from the midterm. First, review the script written in Stata available here (https://github.com/jbhender/Stats506_F18/tree/master/solutions/PS3). In this question, you will work through various options for translating this analysis into R. You may submit all or some of these, but each part must be entirely correct to earn the points listed.

a. Write a translation using `data.table` for the computations. [5 pts]

b. Write a function to compute the univariate regression coefficient by group for arbitrary dependent, independent, and grouping variables. Use `data.table` for computations within your function. Test your function by showing it produces the same results as in part a. [10 pts]

c. Compute the regression coefficients using the dplyr verb `summarize_at()`. [5 pts]

d. Write a function similar to the one in part b to compute arbitrary univariate regression coefficients by group. Use `dplyr` for computations within your function. You should read the "Programming with dplyr" vignette at `vignette('programming', 'dplyr')` before attempting this. Warning: this may be difficult to debug! [10 points]