PS132B Problem Set 3

Due 12:29PM Thursday February 23, 2023

Please submit this assignment by uploading your R Markdown code file (`.Rmd`) AND either an html (`.html`) or pdf (`.pdf`) output onto bCourses before the due time, with easy-to-recognize file names (e.g., `pset3_KirkBansak.Rmd`). Your homework will be graded based on completeness, accuracy, and readability of both code and written answers.

Unless a package is explicitly mentioned in the problem set, you should not use any `R` packages beyond the following packages that are automatically loaded by default when you open up `RStudio`: `base`, `datasets`, `grDevices`, `graphics`, `methods`, `stats`, `utils`. In addition, you may also use `ggplot2`.

The point allocation in this problem set is given by:

Q1.1	Q1.2	Bonus
12	6	5

Q2.1	Q2.2	Q2.3	Q2.4	Q2.5	Q2.6	Q2.7	Q2.8	Q2.9
1	5	5	5	5	5	15	5	5

Q3.1	Q3.2	Q3.3	Q3.4	Q3.5	Q3.6	Bonus	**Total (Bonus)**
1	5	5	10	5	5	5	**100 (10)**

Q1: Programming a Bootstrap Function

Begin by running the following two lines of code.

```
set.seed(123)
x <- rexp(1500, rate = 2)
```

This will create a vector of data, `x`, which we will pretend is a random sample of data (1500 observations) on a single variable.

1) Write your own function that will compute a bootstrap percentile confidence interval for any specified univariate (single variable) statistic of interest. Your function should be named `boot_univariate` and should have four arguments:

   - `datvec`: a vector of data
   - `statint`: a function corresponding to the univariate statistic of interest
   - `B`: number of bootstrap resamples
   - `alpha`: type I error tolerance corresponding to desired confidence interval size (e.g. 0.05 corresponds to a 95% confidence interval)

   The function should return the resulting confidence interval.

2) After you have written and loaded your function, you should be able to run the following line of code:

```
boot_univariate(datvec = x, statint = median,
                B = 10000, alpha = 0.05)
```

   Report the results of this line of code, and explain what the results represent.

BONUS) Using your bootstrap function, again applied to `x`, compute and report a 95% confidence interval for the interquartile range (i.e. difference between 75th and 25th percentiles). Hint: You may need to program *another*, separate function that computes the interquartile range.

Q2: 2006 California Congressional Election Results

In this question, we will analyze vote returns for California House elections and vote choice in a presidential election. Specifically, our goal in this question is to predict the proportion of votes that a Democratic candidate for a House seat wins in a "swing district": one where the support for Democratic and Republican candidates is about equal and the incumbent is a Democrat.

1) Load the data set `ca2006.csv`, a slightly modified version of the 2006 House election return data from the PSCL library

   - The data set contains the following variables:

     `district`: California Congressional district

     `prop_d`: proportion of votes for the Democratic candidate

     `dem_pres_2004`: proportion of two-party presidential vote for Democratic candidate in 2004 in Congressional district

     `dem_pres_2000`: proportion of two-party presidential vote for Democratic candidate in 2000 in Congressional district

     `dem_inc`: An indicator equal to 1 if the Democrat is the incumbent

     `contested`: An indicator equal to 1 if the election is contested

2) Create a plot of the proportion of votes for the Democratic candidate (`prop_d`), against the proportion of the two-party vote for the Democratic presidential candidate in 2004 (`dem_pres_2004`) in the district. Be sure to clearly label the axes and provide an informative title for the plot

3) Regress the proportion of votes for the Democratic candidate, against the proportion of the two-party vote for the Democratic presidential candidate in 2004 in the district. Print the results and add the bivariate regression line to the plot.

4) Using the bivariate regression and a function you have written yourself (**not the `predict()` function**!), report the predicted vote share for the Democratic candidate if `dem_pres_2004` $= 0.5$

5) Now, regress `prop_d` against: `dem_pres_2004`, `dem_pres_2000`, and `dem_inc`.

6) Using the multivariate regression from 5) and a function you have written yourself, report the predicted vote share for the Democratic candidate if:

   `dem_pres_2004` $= 0.5$

   `dem_pres_2000` $= 0.5$

   `dem_inc` $= 1$

7) Implement the bootstrap to characterize the uncertainty for our response variable predictions (you do not need to program a bootstrap function for this part). Specifically, for 10000 bootstrap iterations (B):

   a) Randomly select 53 rows, the number of districts in California in 2006, *with replacement*.

   b) Using the randomly selected (bootstrapped) data set, fit the bivariate and multivariate regressions specified earlier.

   c) Using the fitted regressions, predict the expected vote share for the Democratic candidate for each regression, using the values and functions from 4) and 6).

   d) Store the predictions from both regressions.

   You should set the seed to π prior to running the bootstrap—i.e. `set.seed(pi)`.

8) Report 95% Confidence Intervals for both predictions. In addition, create histograms for both predictions.

9) We will say the model predicts that the Democrat wins if the predicted vote share is greater than 50%. Based on the results of the bootstrap, what proportion of time does each model predict the Democrat will win?

Q3: Predicting Support for Bill Clinton in 1992

This problem will use a data set (again, modified from the PSCL package) to predict whether a voter will vote for Bill Clinton. The data comes from self-reported voting behavior in the 1992 Presidential election

1) Load the data set `vote92.csv`. It contains

   `clintonvote`: an indicator equal to 1 if the voter supports Clinton and 0 otherwise

   `dem`: an indicator equal to 1 if the voter is a Democrat

   `female`: an indicator equal to 1 if the voter is a woman

`clintondist`: a measure of the candidate's self assessed ideological distance from Clinton

2) What proportion of respondents report voting for Bill Clinton?

3) Using a logistic regression, regress `clintonvote` on `dem`, `female`, and `clintondist`

4) Write a function to predict the probability that a voter supports Clinton based on a logistic regression.

5) Using your function from 4) report the probability a female, Democrat, with `clintondist` $= 1$ votes for Clinton.

6) Now use a linear regression to predict `clintonvote` as a function of `dem`, `female`, and `clintondist`. For all voters (rows) in the data, use the fitted linear regression to compute their predicted probabilities of voting for Clinton. Do the same for the logistic regression. Plot the predicted probabilities from the logistic regression (on the x-axis) against those from the linear regression (on the y-axis).

Bonus) For this question, we're going to use the predicted probabilities for all voters from logistic regression, and we're going to visualize how well they perform using a calibration plot. We will construct the calibration plot "from scratch" (i.e. without using any specialized libraries).

To do this, we will construct 10 bins of data, where each bin corresponds to an interval of width 0.1, starting with the bin $[0.0, 0.1)$. This first bin corresponds to all data points with a predicted probability greater than or equal to 0 AND less than 0.1. The next bin is $[0.1, 0.2)$, and so on.

For each bin, compute (a) the mean predicted probability in that bin and (b) the actual proportion of positives (proportion of data points whose true response variable value is 1) in that bin. For each bin, plot (a) on the x-axis and (b) on the y-axis, creating a plot with 10 points on it. Connect the points with a line. In addition, add a dashed "identity line" (the $y = x$ line) to the plot.

The closeness with which the plotted points trace along the identity line is a rough visualization of how well the predicted probabilities are "calibrated."