Statistics 506 Midterm - Corrections

Name, uniquename:

Due October 29, 2017 at 9:30am via Canvas

Instructions

Answer each of the questions that follow without consulting your peers. Submit your answers as a single pdf with one question per page to Canvas.

You may refer to the class notes or resources but should not dicuss the exam questions with other students.

The original exam had eight questions and you have the opportunity to submit corrections for six of them - questions 3-8. Your final grade will be computed in the following way:

1) For questions 1 and 2, your scores from the in-class portion of the exam are final.
2) For questions 3-8, your score on the corrections will count for 80% and the in-class portion for 20%. If your in-class score is higher than this weighted average, you will recieve the in-class score. For example, if you scored 10/25 on question 3 for the in-class exam and recieve 25/25 on the corrections your score will be:

   $$\max(10, .2 * 10 + .8 * 25) = 22 \text{ points.}$$

   This is intended to reward performance on the in-class midterm.

Questions

3. [25 pts] For normally distributed data $X_i \sim_{iid} N(0, \sigma^2)$ there are two commonly used estimates of the variance σ^2:

   - the unbiased sample variance: $\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^n (X_i - \bar{X})^2$
   - and the maximum likelihood estimate: $\tilde{\sigma}^2 = \frac{1}{n}\sum_{i=1}^n (X_i - \bar{X})^2$.

In this question you will write simulation code for comparing the performance of these two estimators.

a. Write a function `f` that takes a numeric vector `x` and returns a length two vector containing the variance estimates $\hat{\sigma}^2$ and $\tilde{\sigma}^2$. [5 pts]

b. Write a second function, `g`, that accepts an `n` by `m` array and returns an `m` by `2` array with estimates of $\hat{\sigma}^2$ and $\tilde{\sigma}^2$ for each column of the input array. Your code should use vectorization and avoid unnecessary loops. [5 pts]

c. The mean-squared error (MSE) of the estimator $\hat{\sigma}^2$ as an estimate of σ^2 is the expectation $E[(\hat{\sigma}^2 - \sigma^2)^2]$. Likewise, the MSE of $\tilde{\sigma}^2$ is $E[(\tilde{\sigma}^2 - \sigma^2)^2]$. The MSE can be decomposed into terms for the bias and variance as follows:

- bias = $E[\hat{\sigma}^2] - \sigma^2$
- variance = $E[(\hat{\sigma}^2 - E[\hat{\sigma}^2])^2]$
- mse = bias^2 + variance

Complete the code skeleton below to compare $\hat{\sigma}^2$ $\tilde{\sigma}^2$ in terms of the MSE, bias, and variance using simulation for various values of n and σ^2. Your code should not add any additional `for` or `apply` loops and should make use of the function `g` you wrote in part b. [15 pts]

```
## Code skeleton for question 3
mcrep=1e3
sigsq_values = seq(.2, 2, 20)
n_values = seq(10, 30, 5)

for(sigsq in sigsq_values){
  for(n in n_values){

     ## insert your code here
     # generate data: each data set is n iid samples from N(0, sigsq)

     # compute estiamtes for all Monte Carlo replicates

     # compare performance

     # print results

  }
}
```

---

4. Assume we have a data frame `Loblolly` loaded in R with three columns `height`, `Seed`, and `age`, recording, respectively, the height in feet of 14 loblolly trees (Seeds 1-14) at 3, 5, 10, 15, 20, and 25 years. `Loblolly` has dimensions $14 * 6 = 84$ by 3. For each *chain of pipes* below: (i) determine the dimensions of the resulting data frame, (ii) concisely explain what each pipe is computing, and (iii) provide a descriptive replacement for column names denoted by `XX0` or `XX1`. You should assume that the `dplyr` package has been loaded. **Your explanations should each be a single sentence.** [30 pts total; 10 pts each]

a.

```
out =
  Loblolly %>%
  group_by(age) %>%
  summarize(XX0 = sum(height) / n(),
            XX1 = sqrt(sum({height - XX0}^2) / {n()-1})
  )
## i) dim(out) =
## ii) We are computing ...
## iii) XX0 =          , XX1=
```

b.

```
out =
  Loblolly %>%
  group_by(Seed) %>%
  mutate(height = height / height[1]) %>%
  filter(age==25) %>%
  ungroup %>%
  summarize(XX0 = median(height))
## i) dim(out) =
## ii) We are computing ...
## iii) XX0 =
```

c.

```
out =
  Loblolly %>%
  filter(age %% 5 == 0) %>%
  group_by(Seed) %>%
  mutate(
    XX0 = diff(c(0, height)) / diff(c(0, age)) # Slight change from in-class
  ) %>%
  ungroup %>%
  group_by(age) %>%
  summarize(XX1 = mean(XX0))
## i) dim(out) =
## ii)  We are computing ...
## iii) XX1 =
```

---

5. Consider the Loblolly data from the previous problem. Convert the long-form Loblolly data from question 4 into a data frame with one row per tree (i.e. value of `Seed`) and columns for the heights in each year. Rename these columns `yr3` . . . `yr25`. Standardize each column of heights using z-scores (you can do this before or after reshaping).

a. Write an R expression using `dplyr` and `tidyr` to accomplish the task above. Assume the Loblolly data are already present in a data frame named `Loblolly` [10 pts].

b. Write Stata commands to accomplish the task. Assume the data set has already been loaded into Stata in the long format described in question 4. [10 pts]

---

6. Consider an R data frame `population` with three columns - 'country', 'year', and 'population' - giving the populations of various countries from 1995 to 2013 as reported by the World Health Organization. Write R code to accomplish each of the tasks below. Be as concise as possible; each of the tasks can be accomplished in a single expression using pipes. [25 pts]

a. Count the number of times each country appears in the data and store this in a data frame `n_years`. [5 pts]

b. Use `n_years` from part a to create a reduced data frame `pop_complete` that includes only countries with all 19 years from 1995 to 2013 observed. *The data frame `pop_complete` should have the same columns as `population`.* [5 pts]

c. Using the reduced data set `pop_complete` from part (b), for each country compute the relative population growth from 1995 to 2000, from 2000 to 2005, and from 2005 to 2010. Store the results in an object `rel_growth`. [10 pts]

d. Use the `rel_growth` object of part (c) to find all countries whose population declined during all three periods. [5 pts]

7. In this problem you will write a function for generating bootstrap samples of a numeric vector `x`. You will return the samples as an object of class `bootstrap` and write S3 methods for this class. [25 pts total]

a. Define a `bootstrap` function that accepts: a numeric vector `x`, a function `f`, and an integer `n` indicating the number of bootstrap samples to draw. Set a default value of `n=1e3`. Your function should return an object of class `bootstrap` with two elements: 1) `obs` with the value of `f(x)`, 2) `boot_stats` a vector of length `n` containing `f` the values returned by `f` on each bootstrap sample. Your `bootstrap` function should also check that `f(x)` is a length one numeric vector and produce an error if not. [15 pts]

b. Write an S3 `print` method for objects of class bootstrap that displays on a single line the observed statistic and number of bootstrap samples. Be sure to label the values displayed. [5 pts]

c. Write an S3 `summary` method for objects of class bootstrap that displays the observed statistic, a confidence interval constructed using the percentile method, and the associated confidence level on a single line. The confidence level should be configurable using an argument `level` and have a default of .95. [5 pts]

---

8. Below is a short uncommented Stata program. Write a single sentence concisely describing what the program appears to be doing. If you don't recognize the value being computed, express it as a mathematical formula. Then, translate the Stata program into R. [20 pts]

```
*-------*
* Stata *
*-------*
import delimited mtcars.csv
keep mpg cyl wt
generate  mpg_wt = mpg*wt

collapse (count) n=mpg (sum) mpg_wt (mean) mpg wt \\\
  (sd) mpg_sd=mpg  wt_sd=wt, by cyl

generate r_hat = (mpg_wt - n*mpg*wt) / ((n-1)*mpg_sd*wt_sd)

keep cyl r_hat
```