Problem set 8

Siwei Dai

November 19, 2021

NOTE: Start with the file `ps8_2021.Rmd` (available from the github repository at https://github.com/UChicago-pol-methods/IntroQSS-F21/tree/main/assignments). Modify that file to include your answers. Make sure you can "knit" the file (e.g. in RStudio by clicking on the `Knit` button). Submit both the Rmd file and the knitted Pbb via Canvas.

In this assignment we will return to data from an experiment that measured the effect of constituent names in emails on legislator replies. The published paper is:

Butler, D. M., & Broockman, D. E. (2011). *Do politicians racially discriminate against constituents? A field experiment on state legislators.* AJPS.

The data file is `Butler_Broockman_AJPS_2011_public_csv.csv` and it is found in the `data/legislators_email` directory of the course github repository.

To load the data you can either download and read in the local file, or you can read in the url from github. Note that reading in by the url will only work when you have an internet connection:

```
file <- './../data/legislators_email/Butler_Broockman_AJPS_2011_public_csv.csv'
bb <- read_csv(file)
```

Question 1: Inference from a single random variable

(1a) Create an object called `theta_hat` which is the mean of the `reply_atall` variable in the data set.

```
# your code here
theta_hat <-  mean(bb$reply_atall)
theta_hat
```

```
## [1] 0.5653427
```

(1b) Create an object called `se_hat` which is the estimate of the standard error of the mean of the `reply_atall` variable in the data set, using the formula based on the unbiased sample variance.

```
# your code here
se_hat <- sqrt(var(bb$reply_atall)/length(bb$reply_atall))
se_hat
```

```
## [1] 0.007112145
```

(1c) The formula for the normal approximation-based confidence intervals is below

$$CI_n = \left(\hat{\theta}_n - z_{1-\alpha/2} \times \hat{\text{se}},\ \hat{\theta}_n + z_{1-\alpha/2} \times \hat{\text{se}}\right)$$

z_c describes the c-th quantile of the standard normal distribution. For 95% confidence intervals, $\alpha = 0.05$, so we want to find $z_{1-\alpha/2} = z_{0.975}$. Using qnorm, get the 97.5-th quantile of the standard normal distribution.

```
# your code here
qnorm(p = 0.975)
```

```
## [1] 1.959964
```

(1d) Using `theta_hat`, `se_hat`, and your answer to the previous question, report the 95% normal approximation-based confidence intervals for the estimate of `theta_hat`

```
# your code here
CI_1d <- c(theta_hat + c(-1, 1) * qnorm(p = 0.975) * se_hat)
CI_1d
```

```
## [1] 0.5514031 0.5792822
```

(1e) Interpret what the 95% confidence interval means.
The probability that the true value of the estimand falls in (0.5514031, 0.5792822) is 95%.

(1f) To get the 90% confidence intervals, we will set α as 0.10. So we want to find $z_{1-\alpha/2} = z_{0.95}$. Using qnorm, get the 95-th quantile of the standard normal distribution.

```
# your code here
z_0.95 <- qnorm(p = 0.95)
```

(1g) Using your answer from the question above, report the 90% normal approximation-based confidence intervals for the estimate of `theta_hat`.

```
# your code here
CI_1g <- c(theta_hat + c(-1, 1) * se_hat * z_0.95)
CI_1g
```

```
## [1] 0.5536442 0.5770411
```

(1h) Create a vector of 1000 bootstrapped estimates of the sample mean of `reply_atall`. Save this vector as an object. Report the standard deviation across the estimates. The standard deviation of your bootstrapped estimates should be similar to your answer to 1b above.

Note: This should look very much like your solution to (2e) on hw 7, but you should be sampling with replacement from `bb$reply_atall`.

```
# your code here
bootsp_mean <- map(1:1000, ~sample(bb$reply_atall, replace = TRUE)) %>%
    map(mean) %>%
    unlist
bootsp_mean_sd <- bootsp_mean %>%
    sd()
bootsp_mean_sd
```

```
## [1] 0.007266542
```

(1i) We can compare the distribution of the estimator under the bootstrap procedure and under the normal approximation. Using the `quantile()` function and your saved vector of 1000 bootstrapped estimates of the sample mean, report the 2.5th and 97.5th quantiles of the estimates under the bootstrap. These cover 95% of the empirical distribution of the bootstrap. How do they compare to your 95% normal approximation-based confidence intervals in your answer to 1d above?

```r
# your code here
quantile(bootsp_mean, probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.5511422 0.5793425
```

```r
CI_1d
```

```
## [1] 0.5514031 0.5792822
```

```r
# The CI of estimates from bootstraps are close to the CI we gather from the sample we observe
```

Question 2: Inference from linear models

(2a) Using `lm_robust`, regress `reply_atall` on `treat_deshawn` interacted with `leg_republican`. Print the model object. Save the vector of coefficients as `theta_hats`.

```r
# your code here
model2a <- lm_robust(reply_atall ~ treat_deshawn * leg_republican, data = bb)
theta_hats <- model2a %>%
    coef()
model2a
```

```
##                                 Estimate Std. Error    t value      Pr(>|t|)
## (Intercept)                   0.52976190 0.01361950 38.8973007 2.385445e-288
## treat_deshawn                 0.01596300 0.01923401  0.8299363  4.066156e-01
## leg_republican                0.09949293 0.02000774  4.9727219  6.829449e-07
## treat_deshawn:leg_republican -0.07550407 0.02848382 -2.6507709  8.056896e-03
##                                 CI Lower    CI Upper   DF
## (Intercept)                   0.50306151  0.55646230 4855
## treat_deshawn                -0.02174436  0.05367037 4855
## leg_republican                0.06026870  0.13871715 4855
## treat_deshawn:leg_republican -0.13134525 -0.01966290 4855
```

(2b) From the model object above, report and interpret the standard errors and 95% confidence intervals on `treat_deshawn` and `treat_deshawn:leg_republican`. Do the confidence intervals include zero? If so/if not, what does that imply?

The standard error on 'treat_deshawn' means that the standard deviation of the sample mean, an estimator of "treat_deshawn", is 0.019234; the standard error on "treat_deshawn:leg_republican" means the standard deviation of the sample mean for the interactive effect of "treat_deshawn" and "leg_republican" is 0.0284838.

The 95% confidence interval for "treat_deshawn" is (-0.0217444, 0.0536704), which means under the assumption that the estimates are normally distributed acorss all samples, the probability of the true estimand falling within the CI is 95%. Similarly, the 95% confidence interval for "treat_deshawn:leg_republican" is (-0.1313453, -0.0196629) means that assuming the estimates are normally distributed, the probability of the true value of the estimand falling within the CI is 95%.

The CI on "treat_deshawn" include 0, which means we would fail to reject the null hypothesis at a p-value $p = 0.05$; the CI on "treat_deshawn:leg_republican" doesn't include 0, which menas we would reject the null hypothesis at a p-value $p \leq 0.05$.

(2c) Using `map()` and `slice_sample(, replace = TRUE)`, take 1000 bootstrap re-samples with replacement of the same size as the original data from the `bb` dataset. Save your bootstrapped samples as an object.

```
# your code here
bootsp <- map(1:1000, ~slice_sample(bb, n = nrow(bb), replace = TRUE))
```

(2d) Using `map()` again, run the same regression as above on *each* of your bootstrapped samples; extract coefficient estimates; and use `bind_rows()` to create a matrix where each row represents estimates from one of your bootstrap samples, and each column is one of the coefficients.

```
# your code here
tbl2d <- bootsp %>%
    map(~lm_robust(reply_atall ~ treat_deshawn * leg_republican, data = .) %>%
            coef()) %>%
    bind_rows
tbl2d
```

```
## # A tibble: 1,000 x 4
##    `(Intercept)` treat_deshawn leg_republican `treat_deshawn:leg_republican`
##            <dbl>         <dbl>          <dbl>                          <dbl>
##  1         0.509       0.0312          0.131                          -0.0973
##  2         0.523       0.0282          0.115                          -0.0642
##  3         0.554       0.00284         0.0534                         -0.0351
##  4         0.515       0.0493          0.0911                         -0.0899
##  5         0.527       0.0241          0.102                          -0.0766
##  6         0.534       0.00313         0.0986                         -0.0766
##  7         0.519       0.0370          0.144                          -0.119
##  8         0.546       0.00892         0.0817                         -0.0872
##  9         0.531       0.00862         0.0965                         -0.0788
## 10         0.547       0.0192          0.0958                         -0.0862
## # ... with 990 more rows
```

(2e) Report the bootstrapped estimates of the standard errors of each of the coefficients. To do this, get the standard deviations of each of the columns.

```
# your code here
# OR can use map(1:ncol(tbl2d), ~sd(tbl2d[[.]]))
bootsp_se <- apply(tbl2d, 2, sd)
bootsp_se
```

```
##                    (Intercept)                  treat_deshawn
##                     0.01355862                     0.01902987
##                 leg_republican treat_deshawn:leg_republican
##                     0.01985205                     0.02833139
```

(2f) Produce normal approximation-based confidence intervals for each of the coefficients using the bootstrapped standard errors, inserted into the same formula for confidence intervals as presented in 1c. Compare these to the standard errors from your original `lm_robust()` model object in question 2a.

```r
# your code here
bootsp_CI <- bind_cols(term = names(theta_hats),
                       mean = theta_hats,
                       se = bootsp_se)  %>%
    mutate('CI_lower' = mean - qnorm(0.975) * se,
           'CI_upper' = mean + qnorm(0.975) * se)

bootsp_CI
```

```
## # A tibble: 4 x 5
##   term                             mean     se CI_lower CI_upper
##   <chr>                           <dbl>  <dbl>    <dbl>    <dbl>
## 1 (Intercept)                    0.530  0.0136   0.503    0.556
## 2 treat_deshawn                  0.0160 0.0190  -0.0213   0.0533
## 3 leg_republican                 0.0995 0.0199   0.0606   0.138
## 4 treat_deshawn:leg_republican -0.0755 0.0283  -0.131   -0.0200
```

```r
bind_cols(coef_lower = model2a$conf.low,
          coef_upper = model2a$conf.high)
```

```
## # A tibble: 4 x 2
##   coef_lower coef_upper
##        <dbl>      <dbl>
## 1     0.503      0.556
## 2    -0.0217     0.0537
## 3     0.0603     0.139
## 4    -0.131     -0.0197
```

```r
# The CIs from bootstraps are pretty close to those in the model
```