1 Item 1 Solutions

1.1 Algorithm: bootstrap estimate of $\widehat{se}(r)$

Step 1: Let B be the number of bootstrap samples taken. With $n = 15$, do SRSWR from schools 1 to 15.

$$\begin{aligned} B_1^* &= \{(X_{11}^*, Y_{11}^*), (X_{21}^*, Y_{21}^*), \ldots, (X_{n1}^*, Y_{n1}^*)\} \\ B_2^* &= \{(X_{12}^*, Y_{12}^*), (X_{22}^*, Y_{22}^*), \ldots, (X_{n2}^*, Y_{n2}^*)\} \\ \vdots &= \vdots \\ B_B^* &= \{(X_{1B}^*, Y_{1B}^*), (X_{2B}^*, Y_{2B}^*), \ldots, (X_{nB}^*, Y_{nB}^*)\} \end{aligned} \tag{1.1}$$

Step 2: Let $S_{x_b}^*$ and $S_{y_b}^*$ be the standard deviations of the variables, X_b^* and Y_b^*, respectively, where $b = \{1, 2, \ldots, B\}$. Calculate the pearson product coefficient of correlation, r_b^*

$$r_b^* = \frac{\frac{1}{n-1}\sum_{i=1}^{n}\left(X_{ib}^* - \bar{X}_b^*\right)\left(Y_{ib}^* - \bar{Y}_b^*\right)}{S_{x_b}^* S_{y_b}^*} \tag{1.2}$$

to yield

$$r = \{r_1^*, r_2^*, \ldots, r_B^*\} \tag{1.3}$$

Step 3: Calculate bootstrap estimate of $\widehat{se}(r)$ using

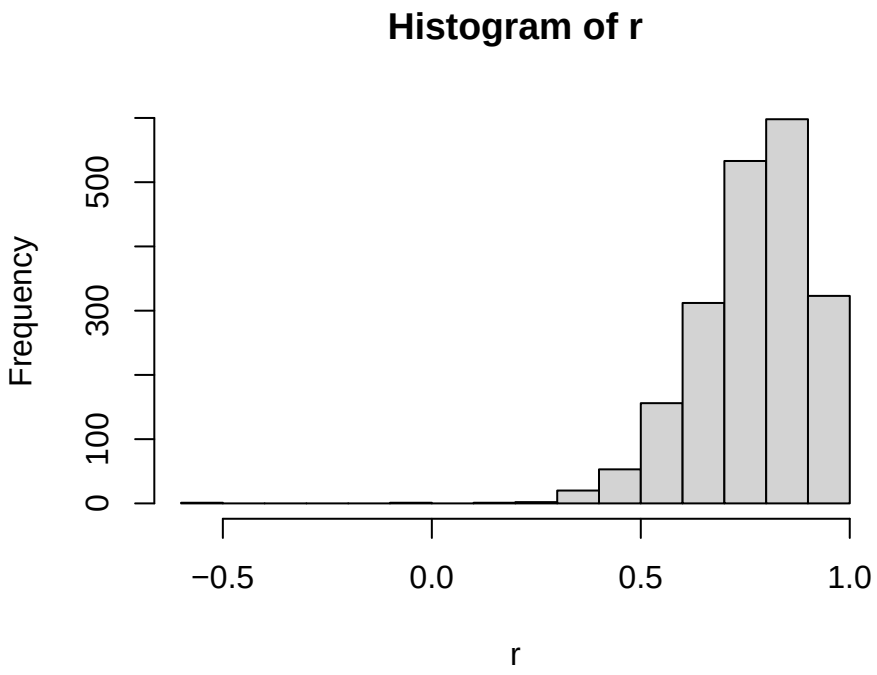
$$\widehat{se}(r) = \sqrt{\frac{\sum_{b=1}^{B}\left(r_b^* - \bar{r}^*\right)^2}{B-1}} \tag{1.4}$$

1.2 Algorithm implementation: bootstrap estimate of $\widehat{se}(r)$

```
CODE FILENAME: ../R/s02_i01_bs_sampling.R

# set seed for reproducibility
set.seed(seed)

r <-
  # Step 2: take the pearson correlation for each bootstrap sample
  sapply(
    # Step 1: 2000 bootstrap samples (with replacement) of size 15
    lapply(
      1:B,
      function(argu){ law_school_data[sample(law_school_data$School,
                                             n1,
                                             replace = TRUE),
                                      c(2,3)] }
    ),
    function(r_star){cor(r_star$LSAT,r_star$GPA, method = "pearson")}
  )

# Step 3: take the bootstrap estimate of the standard error
se_r_boot <- sd(r)

# take the percentile 95% CI for r
```

```
ci_r_boot <- c(quantile(r,.025),quantile(r,.975))
```

i. bootstrap estimate of the standard error of r: 0.13686
ii. 95% confidence interval for ρ (the true population correlation): (0.45104, 0.96251)
iii. a histogram showing the bootstrap distribution of the correlation r



1.3 Change maximum r_b^* and recompute bootstrap estimate of $\widehat{se}(r)$

```
CODE FILENAME: ../R/s03_i01_max_value.R

source("../R/s02_i01_bs_sampling.R")

r_max <- max(r)

r_replaced <- replace(r, r==r_max, 100*r_max)
r_replaced_max <- max(r_replaced)
se_r_replaced_boot <- sd(r_replaced)

se_percent_change <- ((se_r_replaced_boot - se_r_boot)/se_r_boot)*100
```

The original maximum of r_b^*'s is 0.99366 while the new one is 99.36555. Calculating the new $\widehat{se}(r^*)$ gives the value 2.20897. This meant an increase of 1514.01384% compared to the original value.

2 Item 2 Solutions

2.1 Complete table of α-quantiles

```
CODE FILENAME: ../R/s01_i02_alpha_quantile.R
```

```
source("../R/s02_i01_bs_sampling.R")

quant <- quantile(r,
                  probs = c(0.05, 0.10, 0.15, 0.20, 0.50,
                            0.70, 0.85, 0.90, 0.95),
                  names = TRUE)
```

Table 1: α-Quantiles of r^*_α

α	5%	10%	15%	20%	50%	70%	85%	90%	95%
value	0.524	0.586	0.622	0.659	0.788	0.852	0.904	0.923	0.947

2.2 Compute $\widetilde{se}_\alpha(r)$ for $\alpha = 0.95, 0.90, 0.85$

```
CODE FILENAME: ../R/s02_i02_se.R

source("../R/s01_i02_alpha_quantile.R")

robust_se <- function(quantile_vector, alpha) {
  return(
    (
      quantile_vector[
        names(quantile_vector) == paste0(as.character(alpha*100),"%")
        ] -
      quantile_vector[
        names(quantile_vector) == paste0(as.character((1-alpha)*100),"%")
        ]
    )/(2*qnorm(alpha))
  )
}

robust_se_r <- sapply(c(0.95, 0.90, 0.85),
                      function(x){robust_se(quant, alpha = x)})
```

Table 2: Estimated $\widetilde{se}_\alpha(r)$

α (%)	95%	90%	85%
value	0.129	0.131	0.136

2.3 Change maximum r^*_b and recompute $\widetilde{se}_\alpha(r)$

```
CODE FILENAME: ../R/s03_i02_replace.R
```

```
source("../R/s03_i01_max_value.R")
source("../R/s02_i02_se.R")

quant_replaced <- quantile(r_replaced,
                           probs = c(0.05, 0.10, 0.15, 0.20, 0.50,
                                     0.70, 0.85, 0.90, 0.95),
                           names = TRUE)

robust_se_r_replaced <- sapply(c(0.95, 0.90, 0.85),
                               function(x){robust_se(quant_replaced,
                                                     alpha = x)})
```

Table 3: Estimated $\widetilde{se}_\alpha(r)$ with maximum r_b^* replaced

α (%)	95%	90%	85%
value	0.129	0.131	0.136

The equal results shown in Table 2 and Table 3 demonstrates that indeed the estimator is robust.

3 Item 3 Solutions

3.1 Algorithm: $se(\hat{\beta}_2)$ estimation

Step 1: Let $\epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2), i = 1, \ldots, 24$. Under the model, $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N_p(\mathbf{0}, \sigma^2\mathbf{I}_p)$, estimate

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{3.1}$$

$$\begin{aligned} \hat{\sigma}^2 &= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}\right)^2, \quad \text{least squares OR} \\ \hat{\sigma}^2 &= \frac{1}{n-p}\sum_{i=1}^{n}\left(y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}\right)^2, \quad \text{maximum likelihood} \end{aligned} \tag{3.2}$$

Step 2: (a) Repeat B times: Let $e_i^* \sim N(0, \hat{\sigma}^2), i = 1, \ldots, n$. Compute $y_i^* = \mathbf{x}_i'\hat{\boldsymbol{\beta}} + e_i^*, i = 1, \ldots, n$ (b) Obtain $\hat{\beta}_2{}^*$ from the B OLS estimates for each b bootstrap dataset.

$$\hat{\boldsymbol{\beta}}_b^* = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}_b^* \tag{3.3}$$

Step 3: Calculate the standard error for the set of $\hat{\beta}_2{}^*$'s obtained in Step 2, b.

3.2 Algorithm implementation: $se(\hat{\beta}_2)$ estimation

The below implementation shows that the bootstrap estimate of the standard error of β_2 0.03719 while its usual estimate is 0.03711. The difference is very small (7.84742×10^{-5}).

```
CODE FILENAME: ../R/s02_i03_reg.R
```

```
X <- as.matrix(cbind(1,
                     researcher_salary$X_i1,
                     researcher_salary$X_i2,
                     researcher_salary$X_i3))

y <- as.matrix(researcher_salary$Y_i)

# BOOTSTRAP ESTIMATE ====================================================
# 0.03718712

# Step 1: Calculate beta_hat and sigma2_squared_mle
beta_hat <- solve(t(X)%*%X)%*%t(X)%*%y
resid <- y - X %*% beta_hat
sigma2_squared_lse <- (t(resid) %*% resid) / (n - ncol(X))

# Step 2.a: Generate bootstrap samples
set.seed(seed)
y_star_list <- lapply(
  1:B,
  function(x) {
    X%*%beta_hat + as.matrix(rnorm(n,
                                   mean = 0,
                                   sd = sqrt(sigma2_squared_lse)))
  }
)

# Step 2.b: Calculate beta_2_star
beta_2_star <- sapply(y_star_list,
                      function(y_star) {
                        (solve(t(X)%*%X)%*%t(X)%*%y_star)[3]
                        }
                      )

# Step 3: Get the sd of Calculate beta_2_star's
sd_boot <- sd(beta_2_star)

# USUAL ESTIMATE (see references section for code source) ================
# 0.03710865

vcov_beta_hat <- c(sigma2_squared_lse) * solve(t(X) %*% X)
sd_usual <- sqrt(diag(vcov_beta_hat))[3]
```

3.3 Algorithm: $\frac{\hat{\beta}_1}{\hat{\beta}_3}$ 95% CI estimation

Repeat step 1 up to step 2.a. of the Algorithm: $se(\hat{\beta}_2)$ estimation section.

Replace 2.b. with this: Obtain $\frac{\hat{\beta}_1^*}{\hat{\beta}_3^*}$ for each of the B bootstrap datasets.

Step 3: Calculate the 0.025^{th} and 0.975^{th} quantiles to get the 95% confidence interval.

3.4 Algorithm implementation: $\frac{\hat{\beta}_1}{\hat{\beta}_3}$ 95% CI estimation

We are 95% confident that the true value of the ratio is between 0.31051 and 2.14876. The interval is quite large but it includes 1, which means that the effects of the index of publication quality and index of success in obtaining granting support are likely to be equal.

```
CODE FILENAME: ../R/s04_i03_ratio.R

source("../R/s02_i03_reg.R")

# Step 2.b: Obtain the estimated ratio from the OLS estimates for each
# bootstrap estimate
ratio <- sapply(y_star_list,
                function(y_star) {
                  (solve(t(X)%*%X)%*%t(X)%*%y_star)[2]/
                    (solve(t(X)%*%X)%*%t(X)%*%y_star)[4]
                }
)

# Step 3: take the bootstrap estimate of the ratio
ci_ratio_boot <- c(quantile(ratio,.025),quantile(ratio,.975))
```

4 References

Sonnet, L. *Standard errors in OLS*. https://lukesonnet.com/teaching/inference/200d_standard_errors.pdf

5 Appendix

5.1 Code to read data for items 1 & 2

```
CODE FILENAME: ../R/s01_i01_load_data.R

# wd: /home/scientists/sci01/Projects/bootstrap/solutions/ps_01/child

data <- "../../../problems/ps_01/datasets/law_school_data.RData"

if (file.exists(data)) {
  print(paste(c("The file exists; loading", data), collapse = ' '))
  load(data)
} else {
  paste(c("The file does not exist; creating, loading and saving", data),
        collapse = ' ')
  law_school_data <- data.frame(
    'School' = 1:15,
    'LSAT' = c(576,635,558,578,666,580,555,661,
               651,605,653,575,545,572,594),
    'GPA' = c(3.39,3.30,2.81,3.03,3.44,3.07,3.00,3.43,
              3.36,3.13,3.12,2.74,2.76,2.88,2.96)
  )
```

```
  n1 <- dim(law_school_data)[1]
  seed <- 7
  B <- 2000

  save(law_school_data, seed, B, n1, file=data)
}

rm(data)
```

5.2 Code to read data for item 3

```
CODE FILENAME: ../R/s01_i03_load_data.R

# wd: /home/scientists/sci01/Projects/bootstrap/solutions/ps_01/child

data <- "../../../problems/ps_01/datasets/researcher_salary.RData"

if (file.exists(data)) {
  print(paste(c("The file exists; loading", data), collapse = ' '))
  load(data)
} else {
  paste(c("The file does not exist; creating, loading and saving", data),
        collapse = ' ')
  researcher_salary <- data.frame(
    'i' = 1:24,
    'X_i1' = c(3.5, 5.3, 5.1, 5.8, 4.2, 6.0, 6.8, 5.5, 3.1, 7.2, 4.5, 4.9,
               8.0, 6.5, 6.6, 3.7, 6.2, 7.0, 4.0, 4.5, 5.9, 5.6, 4.8, 3.9),
    'X_i2' = c(9,20,18,33,31,13,25,30,5,47,25,11,
               23,35,39,21,7,40,35,23,33,27,34,15),
    'X_i3' = c(6.1,6.4,7.4,6.7,7.5,5.9,6.0,4.0,5.8,8.3,5.0,6.4,
               7.6,7.0,5.0,4.4,5.5,7.0,6.0,3.5,4.9,4.3,8.0,5.0),
    'Y_i' = c(33.2,40.3,38.7,46.8,41.4,37.5,39.0,40.7,30.1,52.9,38.2,31.8,
              43.3,44.1,42.8,33.6,34.2,48.0,38.0,35.9,40.4,36.8,45.2,35.1)
  )
  n <- dim(researcher_salary)[1]
  seed <- 7
  B <- 2000

  save(researcher_salary, n, B, file = data)
}

rm(data)
```