1 Item 1

1.1 Algorithm

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^*

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

to yield

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

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

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

1.2 Algorithm implementation

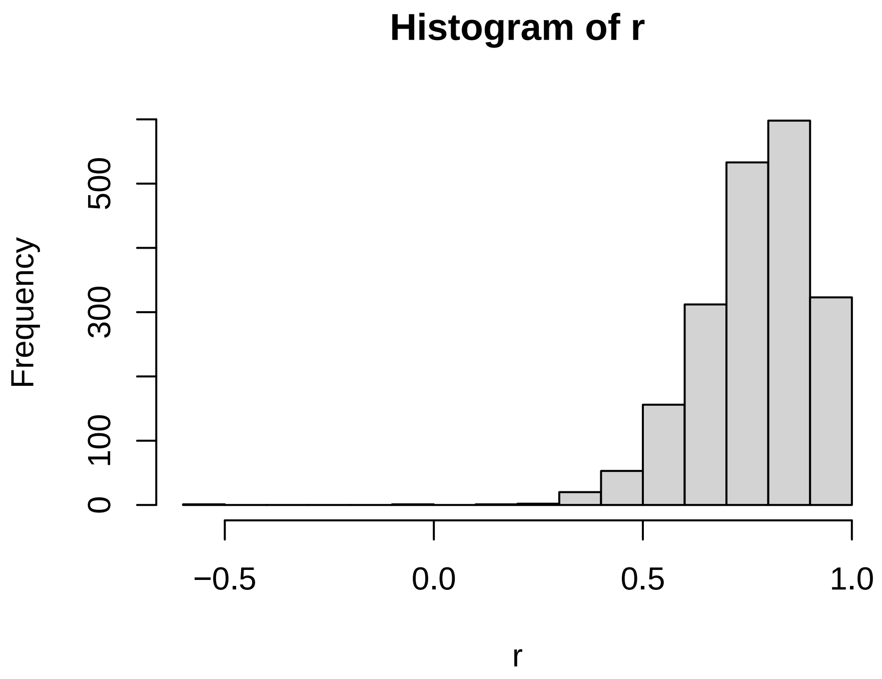
```
# set seed for reproducibility
set.seed(7)

r <-
  # Step 2: take the pearson correlation for each bootstrap sample
  sapply(
    # Step 1: 2000 bootstrap samples (with replacement) of size 15
    lapply(
      1:2000,
      function(x){ law_school_data[sample(law_school_data$School,
                                          15,
                                          replace = TRUE),
                                   c(2,3)] }
    ),
    function(x){cor(x$LSAT,x$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: 0.1369
ii. 95% confidence interval for ρ (the true population correlation): (0.451, 0.9625)
iii. a histogram showing the bootstrap distribution of the correlation r



1.3 Change maximum r_b^*

```
    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 0.9937 while the new one is 99.3656. Calculating the new $\widehat{se}(r)$ gives the value 2.209. This meant an increase of 1514.0138% compared to the original value.

2 Item 2

2.1 Fill in the table

```
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 $\tilde{se}_\alpha(r)$

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 $\tilde{se}_\alpha(r)$

α	95%	90%	85%
value	0.348	0.216	0.146

2.3 Change maximum r^*_b and recompute

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 $\tilde{se}_\alpha(r)$ with maximum r_b^* replaced

α	95%	90%	85%
value	0.348	0.216	0.146

3 Item 3

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

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

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}\right)^2 \tag{3.2}$$

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.

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

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 0.0372 while its usual estimate is 0.0371. The difference is very small (7.8474×10^{-5}).

```
FILENAME ../R/s02_i03_reg.R

    source("../R/s01_i03_load_data.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

B = 2000

# 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) / (nrow(X) - ncol(X))
#sigma2_squared_mle <- (1/n)*sum(resid^2)

# Step 2.a: Generate bootstrap samples
set.seed(7)
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 ==============================================================
# 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 first part.
Replace 2.b. with this: Obtain $\frac{\hat{\beta_1}}{\hat{\beta_3}}$ from the B OLS estimates for each b bootstrap dataset.
Step 3: Calculate the quantiles to get the 95% confidence interval.

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

```
  2.5%  97.5%
0.3105 2.1488
```

We are 95% confident that the true value of the ratio is between 0.3105 and 2.1488. 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.

```
FILENAME ../R/s04_i03_ratio.R

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

    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]
                    }
    )

    ci_ratio_boot <- c(quantile(ratio,.025),quantile(ratio,.975))
    print(ci_ratio_boot)
```

References

Fox, Jean-Paul, and Sukaesi Marianti. 2017. "Person-Fit Statistics for Joint Models for Accuracy and Speed." *Journal of Educational Measurement.*