Question 2

Joseph Froelicher

December 3, 2020

Contents

Part A

```
fit1 = lm(iq ~ expose, data = lead)
fit2 = lm(iq ~ expose + resdur, data = lead)
fit3 = lm(resdur ~ expose, data = lead)

summary(fit1)
```

```
##
## Call:
## lm(formula = iq ~ expose, data = lead)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.705 -10.127   1.295  10.295  46.295
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  102.705      1.767  58.121  < 2e-16 ***
## expose        -7.770      2.901  -2.678  0.00842 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.61 on 122 degrees of freedom
## Multiple R-squared:  0.05553,    Adjusted R-squared:  0.04779
## F-statistic: 7.173 on 1 and 122 DF,  p-value: 0.008421
```

```
summary(fit2)
```

```
##
## Call:
## lm(formula = iq ~ expose + resdur, data = lead)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.452 -10.663   0.750   9.328  52.946
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 108.0446     3.2034  33.728   <2e-16 ***
## expose       -7.6358     2.8676  -2.663   0.0088 **
## resdur       -0.7994     0.4021  -1.988   0.0491 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.42 on 121 degrees of freedom
## Multiple R-squared:  0.0854, Adjusted R-squared:  0.07029
## F-statistic: 5.649 on 2 and 121 DF,  p-value: 0.004512
```

```
summary(fit3)
```

```
##
## Call:
## lm(formula = resdur ~ expose, data = lead)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8478 -2.6795 -0.6795  2.3205  8.3205
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.6795     0.3932  16.989   <2e-16 ***
## expose        0.1683     0.6455   0.261    0.795
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.472 on 122 degrees of freedom
## Multiple R-squared:  0.0005571,  Adjusted R-squared:  -0.007635
## F-statistic: 0.06801 on 1 and 122 DF,  p-value: 0.7947
```

```
gamma_x = fit3$coefficients[2] # gamma_x
beta_m = fit2$coefficients[3] # beta_mediator
beta_adj = fit2$coefficients[2] # beta adjusted
beta_crude = fit1$coefficients[2] # beta crude

# greek = c("\gamma", "\beta")
#
# paste0("$$", greek[1], " = ", gamma_x, "$$")
```

γ_x	β_{adj}	β_{crude}	β_m
0.1683389	-0.7993818	-7.6357785	-7.7703456

Part B

IQ was mediated 1.7318029% by duration at residence.

Part C

```
se = sqrt(
  ( ( fit3$coefficients[2] ^ 2 ) *
  ( summary(fit2)$coefficients[3, 2] ^ 2 ) ) +
  ( ( fit2$coefficients[3] ^ 2 ) *
  ( summary(fit3)$coefficients[2, 2] ^ 2 ) )
)

z = (fit1$coefficients[2] - fit2$coefficients[2]) / se
p = 2 * pnorm(z)
alpha = 0.05

confint = c(
  ( ( fit1$coefficients[2] - fit2$coefficients[2] ) - ( qnorm(1 - (alpha / 2)) * se ) ) / fit1$coeffici
  ( ( fit1$coefficients[2] - fit2$coefficients[2] ) + ( qnorm(1 - (alpha / 2)) * se ) ) / fit1$coeffici
)
```

The 95% confidence interval for percent mediated by duration at residence using the normal approximation to estimate standard error is [-11.3952193, 14.858825] with p = 0.795966.

Part D

```
n <- dim(lead)[1]
b <- 10000

se_vec = vector("double", b)
fit_mat = matrix(NA, nrow = b, ncol = 6)

set.seed(8675309)

for (i in 1:b) {
  new = lead[sample(nrow(lead), n, replace = TRUE), ]

  fit1 = lm(iq ~ expose, data = new)
  fit2 = lm(iq ~ expose + resdur, data = new)
  fit3 = lm(resdur ~ expose, data = new)

  fit_mat[i, 1] = fit3$coefficients[2] # gamma_x
  fit_mat[i, 2] = summary(fit2)$coefficients[3, 2] # se of beta_mediator
  fit_mat[i, 3] = fit2$coefficients[3] # beta_mediator
  fit_mat[i, 4] = summary(fit3)$coefficients[2, 2] # se of gamma_x
  fit_mat[i, 5] = fit1$coefficients[2] - fit2$coefficients[2] # indirect effect
  fit_mat[i, 6] = fit1$coefficients[2] # total effect
}
```

```
se_boot = sqrt(
  ( mean(fit_mat[, 1]) ^ 2 ) *
  ( mean(fit_mat[, 2]) ^ 2 ) +
  ( mean(fit_mat[, 3]) ^ 2 ) *
  ( mean(fit_mat[, 4]) ^ 2 )
)

boot_confint = c(
  ( mean(fit_mat[, 5]) - ( qnorm(1 - (alpha / 2)) * se_boot ) ) / mean(fit_mat[, 6]) * 100,
  ( mean(fit_mat[, 5]) + ( qnorm(1 - (alpha / 2)) * se_boot ) ) / mean(fit_mat[, 6]) * 100
)

z_boot = mean(fit_mat[, 5]) / se_boot
p_boot = 2 * pnorm(z_boot)
```

The 95% bootstrap confidence interval for the percent mediated is [-11.1220797, 14.3338189], with p = 0.8046867. This is very similar to what we saw from part C, but as we would expect, it is slightly narrower.