HW_Week12_108020033

Che-Wei, Chang

2023-05-04 Helped by 108020024

Question 1) Let's visualize how weight and acceleration are related to mpg.

```
# Import the data
cars <- read.table("auto-data.txt", header = FALSE, na.strings = "?")
names(cars) <- c("mpg",  "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")

# Create data frame
cars_log <- with(cars, data.frame(log(mpg), log(cylinders), log(displacement),
                                  log(horsepower), log(weight), log(acceleration),
                                  model_year, origin))
```

a. Let's visualize how weight might moderate the relationship between acceleration and mpg:

   i. Create two subsets of your data, one for light-weight cars (less than mean weight) and one for heavy cars (higher than the mean weight) HINT: consider carefully how you compare log weights to mean weight

   ii. Create a single scatter plot of acceleration vs. mpg, with different colors and/or shapes for light versus heavy cars

   iii. Draw two slopes of acceleration-vs-mpg over the scatter plot: one slope for light cars and one slope for heavy cars (distinguish them by appearance)

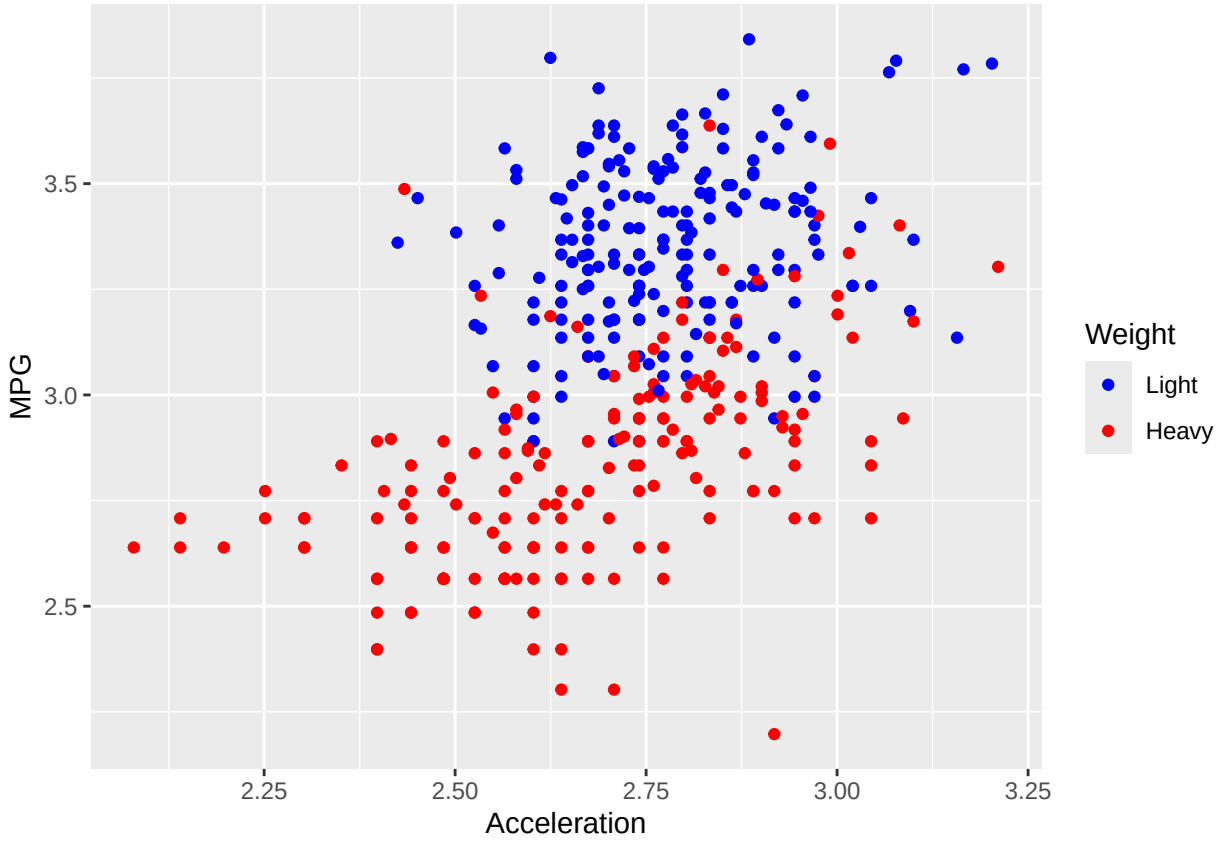
```
# Import the library
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("ggplot2")

# i. Create two subsets, one is light-weight, the other is heavy-weight
```

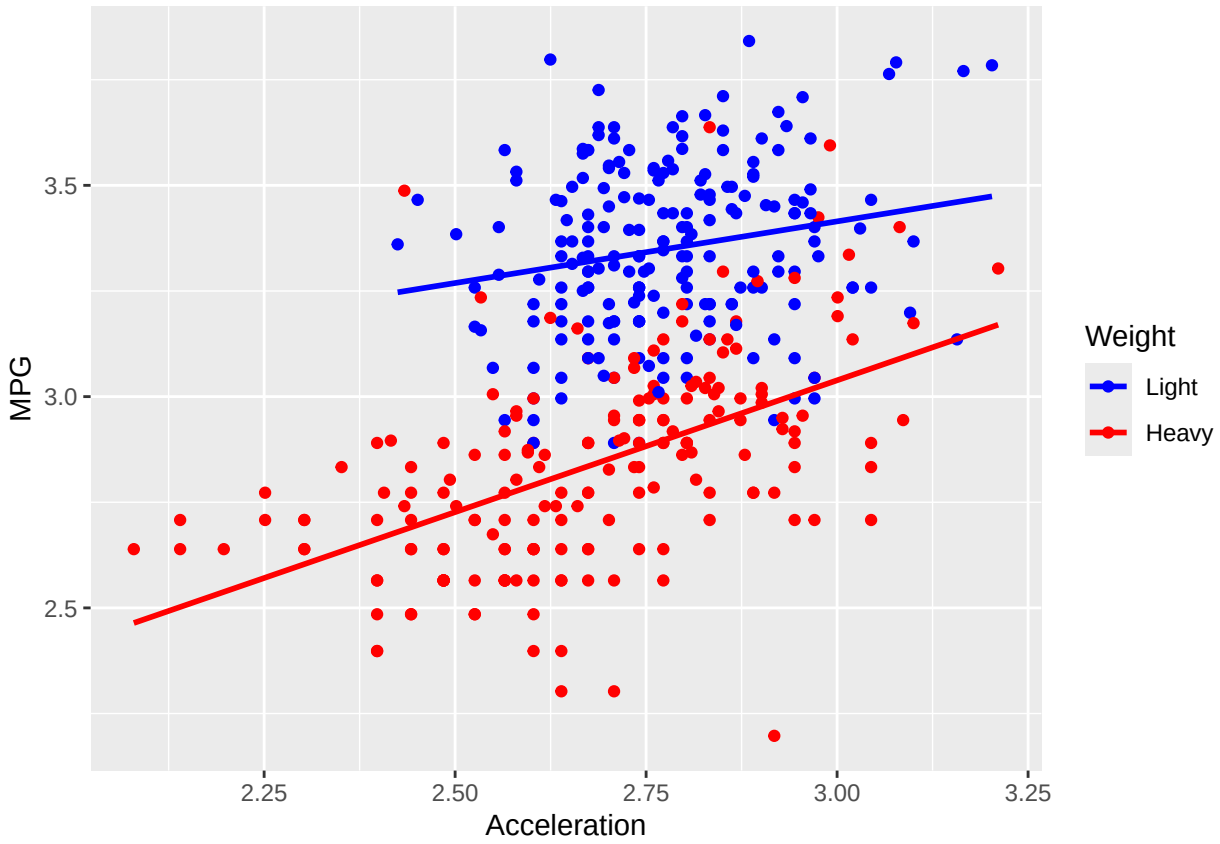
```r
light_cars <- cars_log %>% filter(log.weight. < mean(log.weight.))
heavy_cars <- cars_log %>% filter(log.weight. >= mean(log.weight.))


# ii. Create a scatter plot of acceleration vs. mpg with different colors for light and heavy cars
ggplot(cars_log, aes(x = log.acceleration., y = log.mpg.,
  color = factor(log.weight. >= mean(cars_log$log.weight.)))) + geom_point() +
  labs(x = "Acceleration", y = "MPG", color = "Weight") +
  scale_color_manual(values = c("blue", "red"), labels = c("Light", "Heavy"))
```

```
## Warning: Use of ‘cars_log$log.weight.‘ is discouraged.
## i Use ‘log.weight.‘ instead.
```




```r
# iii. Add separate regression lines for light and heavy cars
ggplot(cars_log, aes(x = log.acceleration., y = log.mpg.,
  color = factor(log.weight. >= mean(cars_log$log.weight.)))) + geom_point() +
  labs(x = "Acceleration", y = "MPG", color = "Weight") +
  scale_color_manual(values = c("blue", "red"), labels = c("Light", "Heavy")) +
  geom_smooth(method = "lm", formula = y ~ x, data = light_cars, se = FALSE) +
  geom_smooth(method = "lm", formula = y ~ x, data = heavy_cars, se = FALSE)
```

```
## Warning: Use of ‘cars_log$log.weight.‘ is discouraged.
## i Use ‘log.weight.‘ instead.
```



b. Report the full summaries of two separate regressions for light and heavy cars where log.mpg. is dependent on log.weight., log.acceleration., model_year and origin

```
# Create linear model for light cars and heavy cars
light_lm <- lm(log.mpg. ~ log.weight. + log.acceleration. +
                 model_year + factor(origin), data = light_cars)
heavy_lm <- lm(log.mpg. ~ log.weight. + log.acceleration. +
                 model_year + factor(origin), data = heavy_cars)

# Show the full summary for light-weight cars
summary(light_lm)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     factor(origin), data = light_cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36590 -0.06612  0.00637  0.06333  0.31513
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6.809014   0.598446  11.378   <2e-16 ***
## log.weight.       -0.821951   0.065769 -12.497   <2e-16 ***
## log.acceleration.  0.111137   0.058297   1.906   0.0580 .
```

```
## model_year         0.033344   0.002049  16.270   <2e-16 ***
## factor(origin)2    0.042309   0.020926   2.022   0.0445 *
## factor(origin)3    0.020923   0.019210   1.089   0.2774
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1102 on 199 degrees of freedom
## Multiple R-squared:  0.7093, Adjusted R-squared:  0.702
## F-statistic:  97.1 on 5 and 199 DF,  p-value: < 2.2e-16
```

```
# Show the full summary for heavy-weight cars
summary(heavy_lm)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     factor(origin), data = heavy_cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37099 -0.07224  0.00150  0.06704  0.42751
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.132892   0.677740  10.525  < 2e-16 ***
## log.weight.       -0.825517   0.068101 -12.122  < 2e-16 ***
## log.acceleration.  0.031221   0.055465   0.563  0.57418
## model_year         0.031735   0.003254   9.752  < 2e-16 ***
## factor(origin)2    0.099027   0.033840   2.926  0.00386 **
## factor(origin)3    0.063148   0.065535   0.964  0.33650
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1212 on 187 degrees of freedom
## Multiple R-squared:  0.7585, Adjusted R-squared:  0.752
## F-statistic: 117.4 on 5 and 187 DF,  p-value: < 2.2e-16
```

Question 2) Use the transformed dataset from above (cars_log), to test whether we have moderation.

b. Use various regression models to model the possible moderation on log.mpg.: (use log.weight., log.acceleration., model_year and origin as independent variables)

i. Report a regression without any interaction terms

```
# Create linear model without any interaction terms
lm1 <- lm(log.mpg. ~ log.weight. + log.acceleration. + model_year + factor(origin), data = cars_log)

# Show the summary of linear model without any interaction terms
summary(lm1)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38275 -0.07032  0.00491  0.06470  0.39913
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.431155   0.312248  23.799  < 2e-16 ***
## log.weight.       -0.876608   0.028697 -30.547  < 2e-16 ***
## log.acceleration.  0.051508   0.036652   1.405  0.16072
## model_year         0.032734   0.001696  19.306  < 2e-16 ***
## factor(origin)2    0.057991   0.017885   3.242  0.00129 **
## factor(origin)3    0.032333   0.018279   1.769  0.07770 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1156 on 392 degrees of freedom
## Multiple R-squared:  0.8856, Adjusted R-squared:  0.8841
## F-statistic: 606.8 on 5 and 392 DF,  p-value: < 2.2e-16
```

ii. Report a regression with an interaction between weight and acceleration

```
# Create linear model with an interaction between weight and acceleration
lm2 <- lm(log.mpg. ~ log.weight. * log.acceleration. + model_year + factor(origin), data = cars_log)

# Show the summary of linear model with an interaction between weight and acceleration
summary(lm2)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. * log.acceleration. + model_year +
##     factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.37807 -0.06868  0.00463  0.06891  0.39857
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    1.089642   2.752872   0.396  0.69245
## log.weight.                   -0.096632   0.337637  -0.286  0.77488
## log.acceleration.              2.357574   0.995349   2.369  0.01834 *
## model_year                     0.033685   0.001735  19.411  < 2e-16 ***
## factor(origin)2                0.058737   0.017789   3.302  0.00105 **
## factor(origin)3                0.028179   0.018266   1.543  0.12370
## log.weight.:log.acceleration. -0.287170   0.123866  -2.318  0.02094 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.115 on 391 degrees of freedom
## Multiple R-squared:  0.8871, Adjusted R-squared:  0.8854
## F-statistic: 512.2 on 6 and 391 DF,  p-value: < 2.2e-16
```

iii. Report a regression with a mean-centered interaction term

```r
# Do mean centering
weight_mc <- scale(cars_log$log.weight., center = TRUE, scale = FALSE)
acceleration_mc <- scale(cars_log$log.acceleration., center = TRUE, scale = FALSE)

# Create linear model
lm3 <- lm(log.mpg. ~ weight_mc * acceleration_mc +
            model_year + factor(origin), data = cars_log)

# Show the summary of linear model
summary(lm3)
```

```
##
## Call:
## lm(formula = log.mpg. ~ weight_mc * acceleration_mc + model_year +
##     factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37807 -0.06868  0.00463  0.06891  0.39857
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.518882   0.132944   3.903 0.000112 ***
## weight_mc                  -0.880393   0.028585 -30.799  < 2e-16 ***
## acceleration_mc             0.072596   0.037567   1.932 0.054031 .
## model_year                  0.033685   0.001735  19.411  < 2e-16 ***
## factor(origin)2             0.058737   0.017789   3.302 0.001049 **
## factor(origin)3             0.028179   0.018266   1.543 0.123704
## weight_mc:acceleration_mc -0.287170   0.123866  -2.318 0.020943 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.115 on 391 degrees of freedom
```

```
## Multiple R-squared:  0.8871, Adjusted R-squared:  0.8854
## F-statistic: 512.2 on 6 and 391 DF,  p-value: < 2.2e-16
```

iv. Report a regression with an orthogonalized interaction term

```
# Do residuals of interaction's regression
log_weight_acceleration <- cars_log$log.weight. * cars_log$log.acceleration.
interaction_regr <- lm(log_weight_acceleration ~ cars_log$log.weight. + cars_log$log.acceleration.)
interaction_ortho <- interaction_regr$residuals

# Create linear model
lm4 <- lm(log.mpg. ~ log.weight. + log.acceleration. + interaction_ortho, data = cars_log)

# Show the summary of linear model
summary(lm4)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + interaction_ortho,
##     data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49728 -0.10145 -0.01102  0.09665  0.56416
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       10.48669    0.33430  31.369  < 2e-16 ***
## log.weight.       -1.00048    0.03187 -31.395  < 2e-16 ***
## log.acceleration.  0.21084    0.04949   4.260 2.56e-05 ***
## interaction_ortho  0.25295    0.16807   1.505    0.133
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1613 on 394 degrees of freedom
## Multiple R-squared:  0.7763, Adjusted R-squared:  0.7746
## F-statistic: 455.7 on 3 and 394 DF,  p-value: < 2.2e-16
```

c. For each of the interaction term strategies above (raw, mean-centered, orthogonalized) what is the correlation between that interaction term and the two variables that you multiplied together?

```
w_raw <- cars_log$log.weight.
a_raw <- cars_log$log.acceleration.
# For raw
cor(data.frame(w_raw, a_raw, w_raw * a_raw))
```

```
##                    w_raw      a_raw w_raw...a_raw
## w_raw          1.0000000 -0.4256194     0.1083055
## a_raw         -0.4256194  1.0000000     0.8528810
## w_raw...a_raw  0.1083055  0.8528810     1.0000000
```

```r
# For mean-centered
cor(data.frame(weight_mc, acceleration_mc, weight_mc * acceleration_mc) )
```

```
##                               weight_mc acceleration_mc
## weight_mc                     1.0000000      -0.4256194
## acceleration_mc              -0.4256194       1.0000000
## weight_mc...acceleration_mc -0.2026948       0.3512271
##                              weight_mc...acceleration_mc
## weight_mc                                      -0.2026948
## acceleration_mc                                 0.3512271
## weight_mc...acceleration_mc                     1.0000000
```

```r
# For orthogonalized
cor(data.frame(w_raw, a_raw, interaction_ortho))
```

```
##                            w_raw         a_raw interaction_ortho
## w_raw              1.000000e+00 -4.256194e-01      3.668639e-17
## a_raw             -4.256194e-01  1.000000e+00     -1.132144e-17
## interaction_ortho  3.668639e-17 -1.132144e-17      1.000000e+00
```

Question 3) We saw earlier that the number of cylinders does not seem to directly influence mpg when car weight is also considered. But might cylinders have an indirect relationship with mpg through its weight?

Let's check whether weight mediates the relationship between cylinders and mpg, even when other factors are controlled for. Use log.mpg., log.weight., and log.cylinders as your main variables, and keep log.acceleration., model_year, and origin as control variables (see gray variables in diagram).

a. Let's try computing the direct effects first:

   i. Model 1: Regress log.weight. over log.cylinders. only (check whether number of cylinders has a significant direct effect on weight)

```r
# Create model that regress log.weight. over log.cylinder. only
m1 <- lm(log.weight. ~ log.cylinders., data = cars_log)

# Show the summary
summary(m1)
```

```
##
## Call:
## lm(formula = log.weight. ~ log.cylinders., data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35473 -0.09076 -0.00147  0.09316  0.40374
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.60365    0.03712  177.92   <2e-16 ***
## log.cylinders.  0.82012    0.02213   37.06   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1329 on 396 degrees of freedom
## Multiple R-squared:  0.7762, Adjusted R-squared:  0.7757
## F-statistic:  1374 on 1 and 396 DF,  p-value: < 2.2e-16
```

ii. Model 2: Regress log.mpg. over log.weight. and all control variables (check whether weight has a significant direct effect on mpg with other variables statistically controlled)

```
# Create the model that regress log.mpg. over log.weight. and all control variables
m2 <- lm(log.mpg. ~ log.weight. + log.cylinders. + log.acceleration. +
           model_year + factor(origin), data = cars_log)

# Show the summary
summary(m2)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.cylinders. + log.acceleration. +
##     model_year + factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39866 -0.06888  0.00227  0.06718  0.40603
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.25316    0.34818  20.831   <2e-16 ***
## log.weight.       -0.83628    0.04523 -18.491   <2e-16 ***
## log.cylinders.    -0.05119    0.04438  -1.153   0.2495
## log.acceleration.  0.03997    0.03798   1.053   0.2932
## model_year         0.03240    0.00172  18.838   <2e-16 ***
## factor(origin)2    0.05298    0.01840   2.880   0.0042 **
## factor(origin)3    0.02984    0.01840   1.622   0.1057
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1156 on 391 degrees of freedom
## Multiple R-squared:  0.886,  Adjusted R-squared:  0.8842
## F-statistic: 506.3 on 6 and 391 DF,  p-value: < 2.2e-16
```

b. What is the indirect effect of cylinders on mpg? (use the product of slopes between Models 1 & 2)

```
# Calculate the indirect effect
indir_effect <- m1$coefficients[2] * m2$coefficients[2]

# Show the result
cat("indirect effect: ", indir_effect, "\n")
```

```
## indirect effect:  -0.6858539
```

c. Let's bootstrap for the confidence interval of the indirect effect of cylinders on mpg

i. Bootstrap regression models 1 & 2, and compute the indirect effect each time: What is its 95% CI of the indirect effect of log.cylinders. on log.mpg.?

```
# Create bootstrap function
boot_mpg_cylinder <- function(model1, model2, dataset)
{
  boot_index <- sample(1:nrow(dataset), replace=TRUE)
  data_boot <- dataset[boot_index, ]
  regr1 <- lm(model1, data_boot)
  regr2 <- lm(model2, data_boot)
  return(regr1$coefficients[2] * regr2$coefficients[2])
}

# Do bootstrapping
set.seed(42)
indirect <- replicate(2000, boot_mpg_cylinder(m1, m2, cars_log))

# Show the 95% of the indirect effect of log.cylinders. on log.mpg.
quantile(indirect, probs = c(0.025, 0.975))
```

```
##       2.5%      97.5%
## -0.7607807 -0.6046015
```

ii. Show a density plot of the distribution of the 95% CI of the indirect effect

```
# Show the plot of indirect and add the lines on it
plot(density(indirect))
abline(v=quantile(indirect, probs=c(0.025, 0.975)), col = "blue", lty = 2)
```