HW12

110077443

5/3/2022

Credit: 109077424

Please note that all code in this document is presented in a grey box and the output reflected below each box

- The below code allows lengthy lines of comments to display neatly within the grey box (wrapping it)

```
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)
```

Create a data.frame called cars_log with log-transformed columns (except model_year and origin)

```
# Importing data
auto <- read.table("auto-data.txt", header = FALSE, na.strings = "?")

# Renaming variables
names(auto) <- c("mpg", "cylinders", "displacement", "horsepower",
    "weight", "acceleration", "model_year", "origin", "car_name")
auto$car_name <- NULL  # Removing variable 'car_name'

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

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

a) Visualizing 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)

```
# Calculate mean weight
logweight_mean <- mean(cars_log$log.weight.)
logweight_mean  # Print
```

```
## [1] 7.95689
```

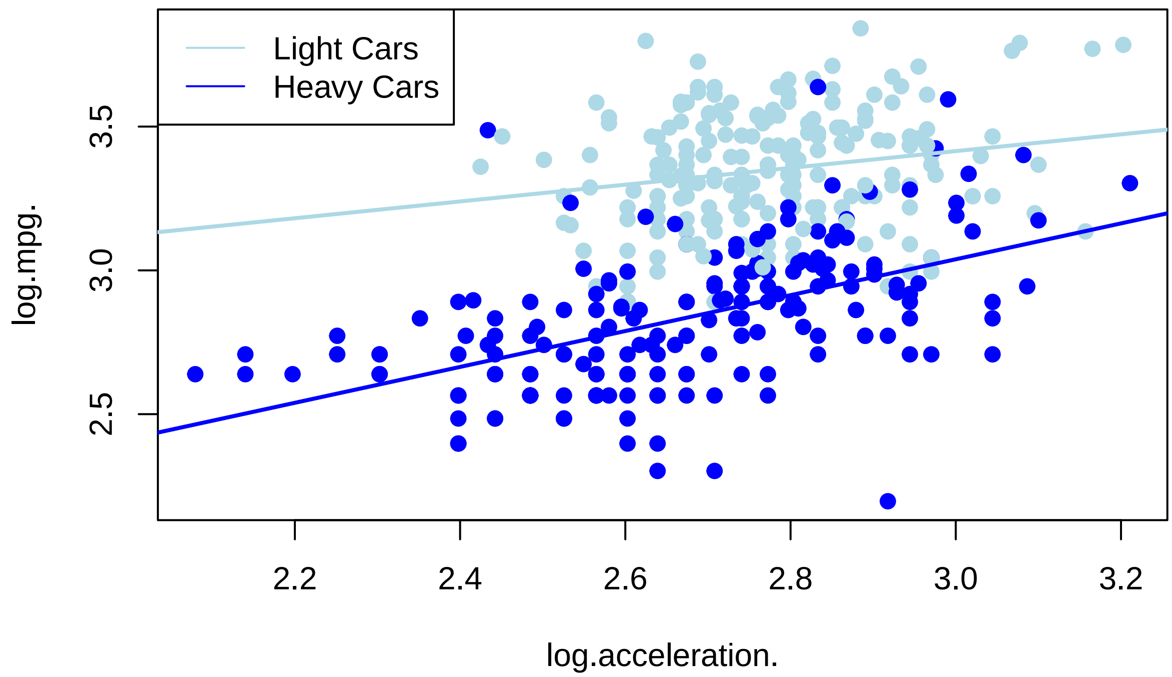
```r
lightweight <- subset(cars_log, log.weight. < logweight_mean)  # 205 obs.
heavyweight <- subset(cars_log, log.weight. >= logweight_mean)  # 193 obs.
```

ii) Create a single scatter plot of acceleration vs. mpg

iii) Draw two slopes of acceleration-vs-mpg over the scatter plot:

```r
# plot points colorized by weight of car
weight_cars <- ifelse(cars_log$log.weight. < logweight_mean,
    "lightblue", "blue")
weight_colors <- c("lightblue", "blue")
with(cars_log, plot(log.acceleration., log.mpg., pch = 19, col = weight_cars[],
    main = "Acceleration vs. MPG beased on Weight of Cars"))

# plot points colorized by weight of car (repeated code in
# same chunk is necessary for knitting)
with(cars_log, plot(log.acceleration., log.mpg., pch = 19, col = weight_cars[],
    main = "Acceleration vs. MPG beased on Weight of Cars"))

# Separate regressions of log.mpg ~ log.acceleration by
# wight of cars
acc_light_regr <- lm(log.mpg. ~ log.acceleration., data = lightweight)
acc_heavy_regr <- lm(log.mpg. ~ log.acceleration., data = heavyweight)

# plot separate regression lines colorized by weight of car
abline(acc_light_regr, col = weight_colors[1], lwd = 2)
abline(acc_heavy_regr, col = weight_colors[2], lwd = 2)

# Adding legend
legend("topleft", lty = 1, c("Light Cars", "Heavy Cars"), col = c("lightblue",
    "blue"))
```

Acceleration vs. MPG beased on Weight of Cars



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

```
# Lightweight cars regression
lm_light <- lm(log.mpg. ~ log.weight. + log.acceleration. + model_year +
    factor(origin), data = lightweight)
summary(lm_light)  # Print
```

```
>
> Call:
> lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
>     factor(origin), data = lightweight)
>
> 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
```

```r
# Heavyweight cars regression
lm_heavy <- lm(log.mpg. ~ log.weight. + log.acceleration. + model_year +
    factor(origin), data = heavyweight)
summary(lm_heavy)  # Print
```

```
>
> Call:
> lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
>     factor(origin), data = heavyweight)
>
> 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
```

c) (not graded) Using your intuition only: What do you observe about light versus heavy cars so far?

ANSWER >

- The regression line appears to fit better with the heavy cars data set
- However, there doesn't seem to be a significant difference between the two slopes

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

a) (not graded) Between weight and acceleration ability (in seconds), use your intuition and experience to state which variable might be a moderating versus independent variable, in affecting mileage.

ANSWER >

- The effect of an independent variable on mileage could be contingent on **acceleration**.

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

i) Report a regression without any interaction terms

```
# No interaction terms - full regression
regr <- lm(log.mpg. ~ log.weight. + log.acceleration. + model_year +
    factor(origin), data = cars_log)

summary(regr)  # Print
```

```
>
> 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

```
# Raw interaction
int_regr_wacc <- lm(log.mpg. ~ log.weight. + log.acceleration. +
    log.weight. * log.acceleration., data = cars_log)

summary(int_regr_wacc)  # Print
```

```
>
> Call:
> lm(formula = log.mpg. ~ log.weight. + log.acceleration. + log.weight. *
>     log.acceleration., 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)                    16.0249     3.6950   4.337 1.84e-05 ***
> log.weight.                    -1.6878     0.4578  -3.687 0.000259 ***
> log.acceleration.              -1.8252     1.3537  -1.348 0.178351
> log.weight.:log.acceleration.   0.2529     0.1681   1.505 0.133123
> ---
> 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
```

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

```
# Mean Centering
mc_log_weight <- scale(cars_log$log.weight., center = TRUE, scale = FALSE)
mc_log_acc <- scale(cars_log$log.acceleration., center = TRUE,
    scale = FALSE)
mc_log_mpg <- scale(cars_log$log.mpg., center = TRUE, scale = FALSE)

# Mean-centered Regression with Interaction
summary(lm(mc_log_mpg ~ mc_log_weight + mc_log_acc + mc_log_weight *
    mc_log_acc))  # Print
```

```
>
> Call:
> lm(formula = mc_log_mpg ~ mc_log_weight + mc_log_acc + mc_log_weight *
>     mc_log_acc)
>
> 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)               0.005447   0.008857   0.615 0.538884
> mc_log_weight            -0.997466   0.031930 -31.239  < 2e-16 ***
> mc_log_acc                0.187500   0.051862   3.615 0.000339 ***
> mc_log_weight:mc_log_acc  0.252948   0.168071   1.505 0.133123
> ---
> 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
```

iv) Report a regression with an orthogonalized interaction term

```r
# Orthogonalized Moderation
oregr <- lm((log.weight. * log.acceleration.) ~ log.weight. +
    log.acceleration., data = cars_log)

# Residuals of interaction's regression
wacc_oregr <- oregr$residuals

# Regression Model with Residual
summary(lm(log.mpg. ~ log.weight. + log.acceleration. + wacc_oregr,
    data = cars_log))  # Print
```

```
>
> Call:
> lm(formula = log.mpg. ~ log.weight. + log.acceleration. + wacc_oregr,
>     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 ***
> wacc_oregr         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?

```
# Raw - correlation
cor(cars_log$log.weight, cars_log$log.weight. * cars_log$log.acceleration)  # Print
```

```
ANSWER > [1] 0.1083055
```

```
cor(cars_log$log.acceleration., cars_log$log.weight. * cars_log$log.acceleration)  # Print
```

```
ANSWER > [1] 0.852881
```

```
# Mean-centered - correlation
cor(mc_log_weight, mc_log_weight * mc_log_acc)  # Print
```

```
ANSWER >            [,1]
ANSWER > [1,] -0.2026948
```

```
cor(mc_log_acc, mc_log_weight * mc_log_acc)  # Print
```

```
ANSWER >           [,1]
ANSWER > [1,] 0.3512271
```

```
# Orthogonalized - Correlation of residual
cor(wacc_oregr, cars_log$log.weight.)  # Print
```

```
ANSWER > [1] 2.468461e-17
```

```
cor(wacc_oregr, cars_log$log.acceleration.)  # Print
```

```
ANSWER > [1] -6.804111e-17
```

3) 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.*

a) Let's try computing the direct effects first

i) Model 1: Regress log.weight. over log.cylinders. only

```r
# Model 1
wt_cyl_regr <- lm(log.weight. ~ log.cylinders., data = cars_log)

require(stargazer)  # Helps understand results of regression analysis
stargazer(wt_cyl_regr, type = "text", main = "Model 1")  # Print (coefficients)
```

```
>
> ===============================================
>                         Dependent variable:
>                     ---------------------------
>                             log.weight.
> -----------------------------------------------
> log.cylinders.               0.820***
>                               (0.022)
>
> Constant                     6.604***
>                               (0.037)
>
> -----------------------------------------------
> Observations                    398
> R2                             0.776
> Adjusted R2                    0.776
> Residual Std. Error      0.133 (df = 396)
> F Statistic         1,373.624*** (df = 1; 396)
> ===============================================
> Note:               *p<0.1; **p<0.05; ***p<0.01
>
> =======
> Model 1
> -------
```

ANSWER >

- Number of cylinders **has a significant direct effect** on weight

ii) Model 2: Regress log.mpg. over log.weight. and all control variables

```r
# Model 2
wt_mpg_regr <- lm(log.mpg. ~ log.weight. + log.acceleration. +
    model_year + factor(origin), data = cars_log)
stargazer(wt_mpg_regr, type = "text", main = "Model 2")  # Print (coefficients)
```

```
>
> ===============================================
>                         Dependent variable:
>                     ---------------------------
>                              log.mpg.
> -----------------------------------------------
> log.weight.                  -0.877***
>                               (0.029)
```

```
>
> log.acceleration.              0.052
>                               (0.037)
>
> model_year                   0.033***
>                               (0.002)
>
> factor(origin)2              0.058***
>                               (0.018)
>
> factor(origin)3               0.032*
>                               (0.018)
>
> Constant                     7.431***
>                               (0.312)
>
> -----------------------------------------------
> Observations                    398
> R2                             0.886
> Adjusted R2                    0.884
> Residual Std. Error      0.116 (df = 392)
> F Statistic          606.774*** (df = 5; 392)
> ===============================================
> Note:               *p<0.1; **p<0.05; ***p<0.01
>
> =======
> Model 2
> -------
```

ANSWER >

- Weight **has a significant direct effect** on mpg with other variables statistically controlled.

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

```
wt_cyl_regr$coefficients[2] * wt_mpg_regr$coefficients[2]  # Print
```

```
ANSWER > log.cylinders.
ANSWER >     -0.7189275
```

ANSWER >

- The indirect effect of cylinders on mpg is **-0.7189275**.

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.?

```r
# Bootstrapped Test of Indirect Effects
boot_mediation <- 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])
}
set.seed(42)
indirect <- replicate(2000, boot_mediation(wt_cyl_regr, wt_mpg_regr,
    cars_log))

quantile(indirect, probs = c(0.025, 0.975))  # Print
```

```
ANSWER >       2.5%      97.5%
ANSWER > -0.7784044 -0.6610106
```

ii) Density plot of the distribution of the 95% CI of the indirect effect

```r
# Plot
plot(density(indirect), col = "green", main = "95% CI of the indirect effect")
abline(v = quantile(indirect, probs = c(0.025, 0.975)), lty = "dotted")  # 95% CI
```