HW9

113078506

2025-04-19

- Gen AI Usage: I use Gen AI to refine my grammar, verify my reasoning and adjust to cleaner code.
- Students who helped: 113078505, 113078502, and 113078514, helped with conclusion reasoning.

Question 1

Set up

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

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

a. Run a new regression on the cars_log dataset, with mpg.log. dependent on all other variables

```
model <- lm(log.mpg. ~ log.cylinders. + log.displacement. +
               log.horsepower. + log.weight. +
               log.acceleration. + model_year + origin, data = cars_log)

summary(model)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.cylinders. + log.displacement. +
##     log.horsepower. + log.weight. + log.acceleration. + model_year +
##     origin, data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41449 -0.06967  0.00040  0.06035  0.39298
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.252158   0.363468  19.953  < 2e-16 ***
```

```
## log.cylinders.    -0.074879   0.061060  -1.226  0.22083
## log.displacement. -0.008015   0.055532  -0.144  0.88532
## log.horsepower.   -0.296585   0.057548  -5.154 4.09e-07 ***
## log.weight.       -0.554906   0.081716  -6.791 4.26e-11 ***
## log.acceleration. -0.182062   0.059222  -3.074  0.00226 **
## model_year         0.029608   0.001726  17.149  < 2e-16 ***
## origin             0.022419   0.010301   2.176  0.03014 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1132 on 384 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.8912, Adjusted R-squared:  0.8892
## F-statistic: 449.5 on 7 and 384 DF,  p-value: < 2.2e-16
```

i. Which log-transformed factors have a significant effect on log.mpg. at 10% significance?

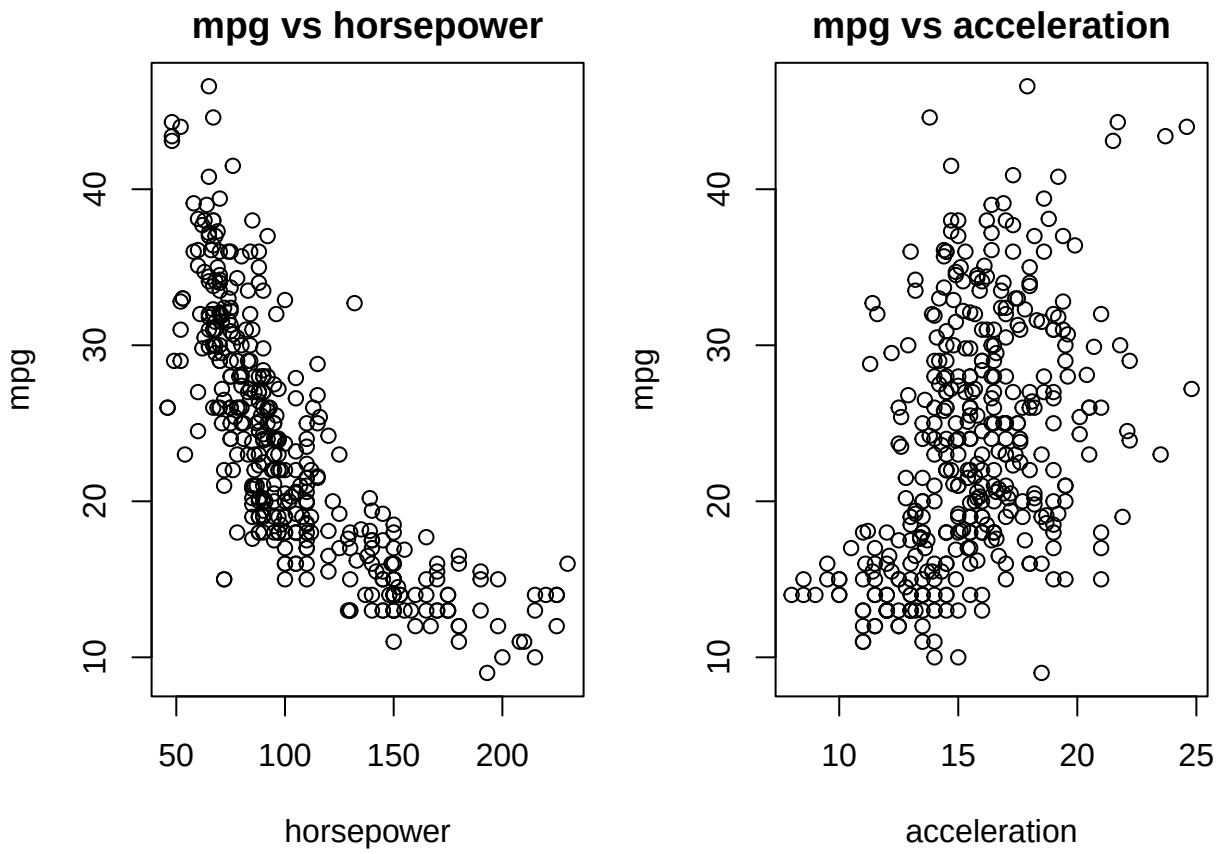
- Log-transformed factors which have a significant effect on log.mpg.: horsepower, weight and acceleration.

ii. Do some new factors now have effects on mpg, and why might this be?

- Compared to the original regression model, the log-transformed variables acceleration and horsepower show a significant effect on log(mpg). This improvement may be due to the fact that log transformation helps the regression model better capture nonlinear relationships. In the original scatter plot, horsepower exhibits a curved relationship with mpg. However, after log transformation, this relationship appears more linear. A similar effect likely occurs with acceleration, although it is less visually apparent in the scatter plots.
- Original scatter plot

```r
# original scatter plot
vars <- c("horsepower",  "acceleration")

par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))
for (v in vars) {
  plot(cars[[v]], cars$mpg, xlab = v, ylab = "mpg", main = paste("mpg vs", v))
}
```

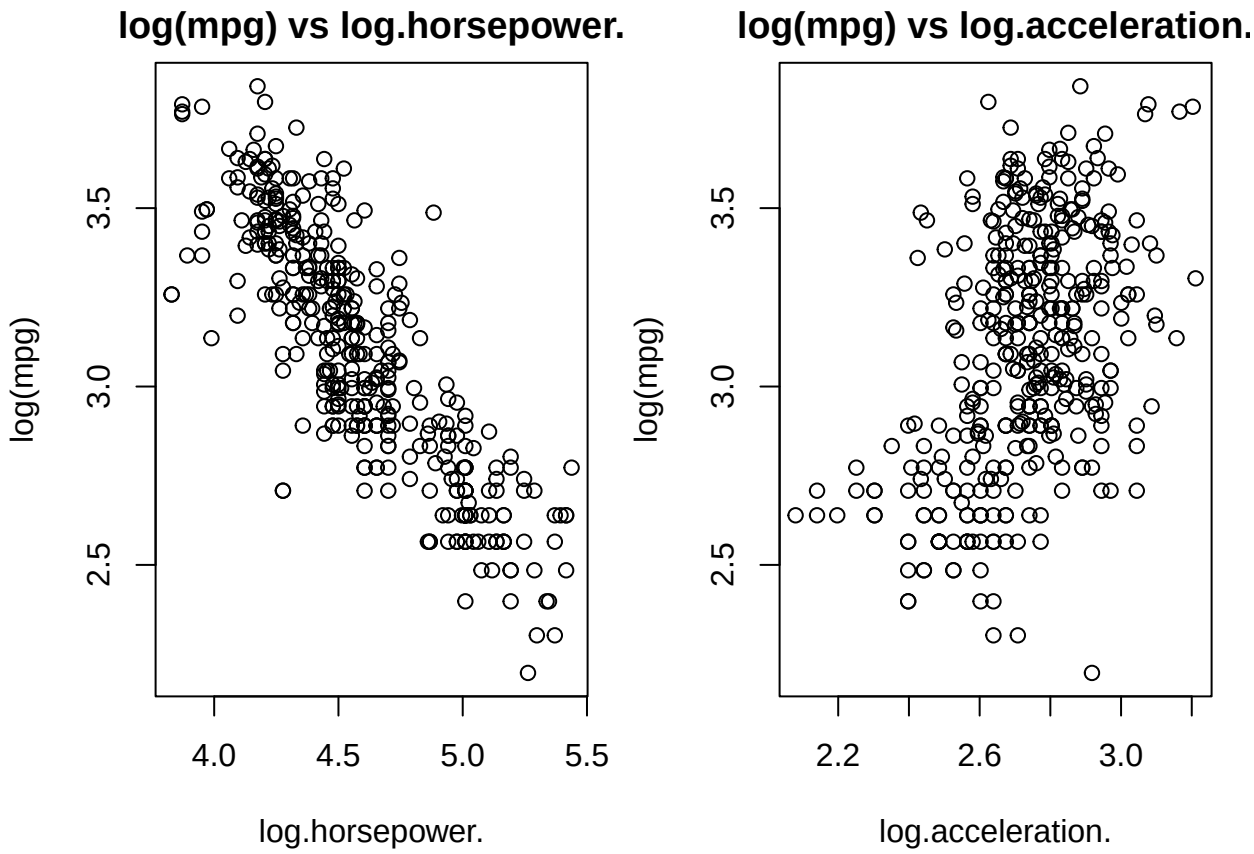



```
par(mfrow = c(1, 1))
```

- Log transformed scatter plot

```
# log transformed scatter plot
vars <- c("log.horsepower.", "log.acceleration.")

par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))
for (v in vars) {
  plot(cars_log[[v]], cars_log$log.mpg., xlab = v, ylab = "log(mpg)", main = paste("log(mpg) vs", v))
}
```




```
par(mfrow = c(1, 1))
```

iii. Which factors still have insignificant or opposite (from correlation) effects on mpg? Why might this be?

- Log transformed Cylinders and Displacement still have insignificant effects on mpg. Although the R^2 of both factors increase, they still can't reach the significant threshold. This might due to the multi-collinearity, if the log transformed variable is highly correlated with other predictors (e.g., log(horsepower) and log(weight)), this can cause instability in the regression coefficients. As a result, the standard errors increase, leading to larger p-values and making the variable appear statistically insignificant.

```
#original r-squared
r2_cylinders <- summary(lm(mpg ~ cylinders, data = cars))$r.squared
r2_displace <- summary(lm(mpg ~ displacement, data = cars))$r.squared

#log transformation r-squared
r2_log_cylinders <- summary(lm(log.mpg. ~ log.cylinders., data = cars_log))$r.squared
r2_log_displace <- summary(lm(log.mpg. ~ log.displacement., data = cars_log))$r.squared

data.frame(
  Variable = c("Cylinders", "Displacement"),
  Original_R2 = c(r2_cylinders, r2_displace),
  LogTransformed_R2 = c(r2_log_cylinders, r2_log_displace)
)
```

```
##       Variable Original_R2 LogTransformed_R2
## 1    Cylinders   0.6012394         0.6731353
## 2 Displacement   0.6467422         0.7405963
```

```
cor_table <- cor(cars[, 1:8], use = "pairwise.complete.obs")
cor_table
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7753963   -0.8042028 -0.7784268 -0.8317409
## cylinders    -0.7753963  1.0000000    0.9507214  0.8429834  0.8960168
## displacement -0.8042028  0.9507214    1.0000000  0.8972570  0.9328241
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8317409  0.8960168    0.9328241  0.8645377  1.0000000
## acceleration  0.4202889 -0.5054195   -0.5436841 -0.6891955 -0.4174573
## model_year    0.5792671 -0.3487458   -0.3701642 -0.4163615 -0.3065643
## origin        0.5634504 -0.5625433   -0.6094094 -0.4551715 -0.5810239
##              acceleration model_year     origin
## mpg             0.4202889  0.5792671  0.5634504
## cylinders      -0.5054195 -0.3487458 -0.5625433
## displacement   -0.5436841 -0.3701642 -0.6094094
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4174573 -0.3065643 -0.5810239
## acceleration    1.0000000  0.2881370  0.2058730
## model_year      0.2881370  1.0000000  0.1806622
## origin          0.2058730  0.1806622  1.0000000
```

b. Let's take a closer look at weight, because it seems to be a major explanation of mpg

```
regr_wt <- lm(mpg ~ weight, data = cars)
summary(regr_wt)
```

i. Create a regression (call it regr_wt) of mpg over weight from the original cars dataset

```
##
## Call:
## lm(formula = mpg ~ weight, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.012  -2.801  -0.351   2.114  16.480
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.3173644  0.7952452   58.24   <2e-16 ***
## weight      -0.0076766  0.0002575  -29.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.345 on 396 degrees of freedom
## Multiple R-squared:  0.6918, Adjusted R-squared:  0.691
## F-statistic: 888.9 on 1 and 396 DF,  p-value: < 2.2e-16
```

```
regr_wt_log <- lm(log.mpg. ~ log.weight., data = cars_log)
summary(regr_wt_log)
```

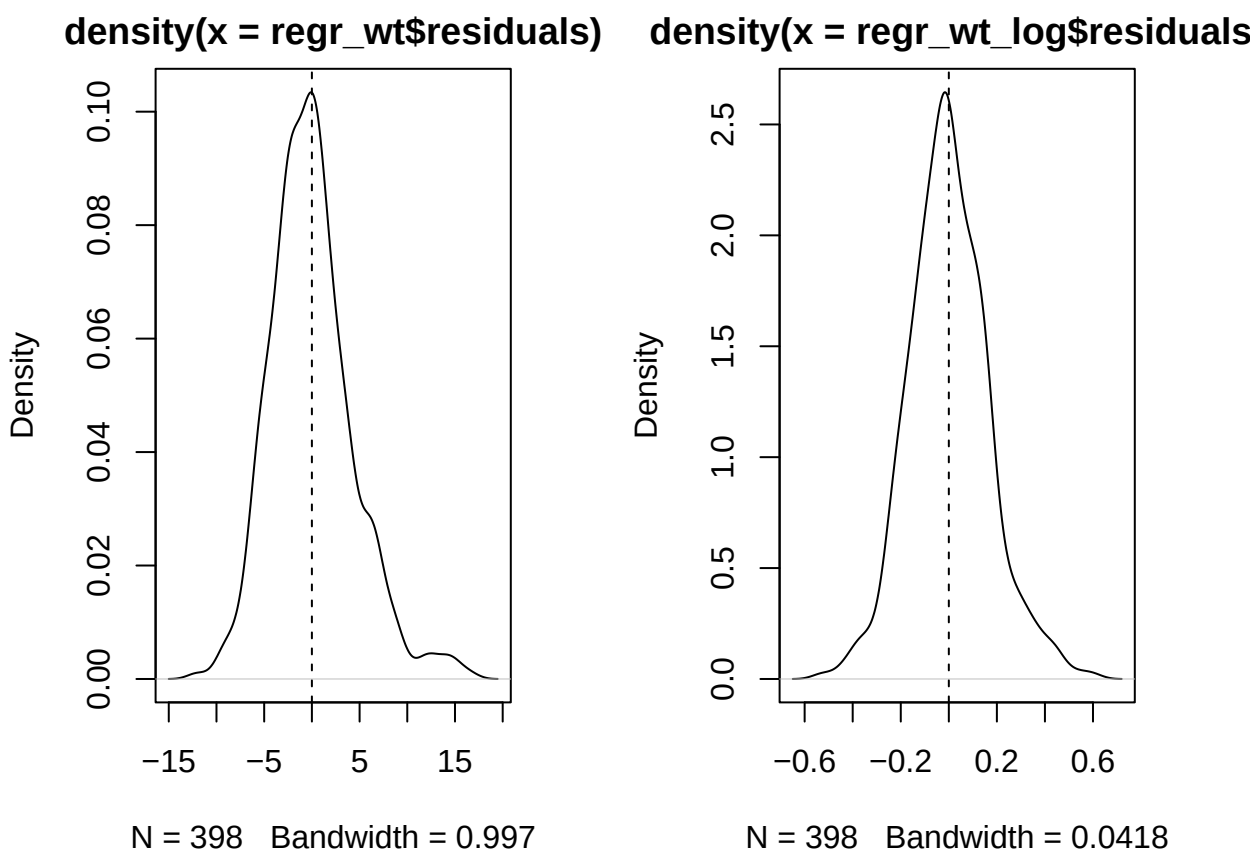
ii. Create a regression (call it regr_wt_log) of log.mpg. on log.weight. from cars_log

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight., data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52408 -0.10441 -0.00805  0.10165  0.59384
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.5219     0.2349   49.06   <2e-16 ***
## log.weight.  -1.0583     0.0295  -35.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.165 on 396 degrees of freedom
## Multiple R-squared:  0.7647, Adjusted R-squared:  0.7641
## F-statistic:  1287 on 1 and 396 DF,  p-value: < 2.2e-16
```

iii. Visualize the residuals of both regression models (raw and log-transformed):

- density plots of residuals

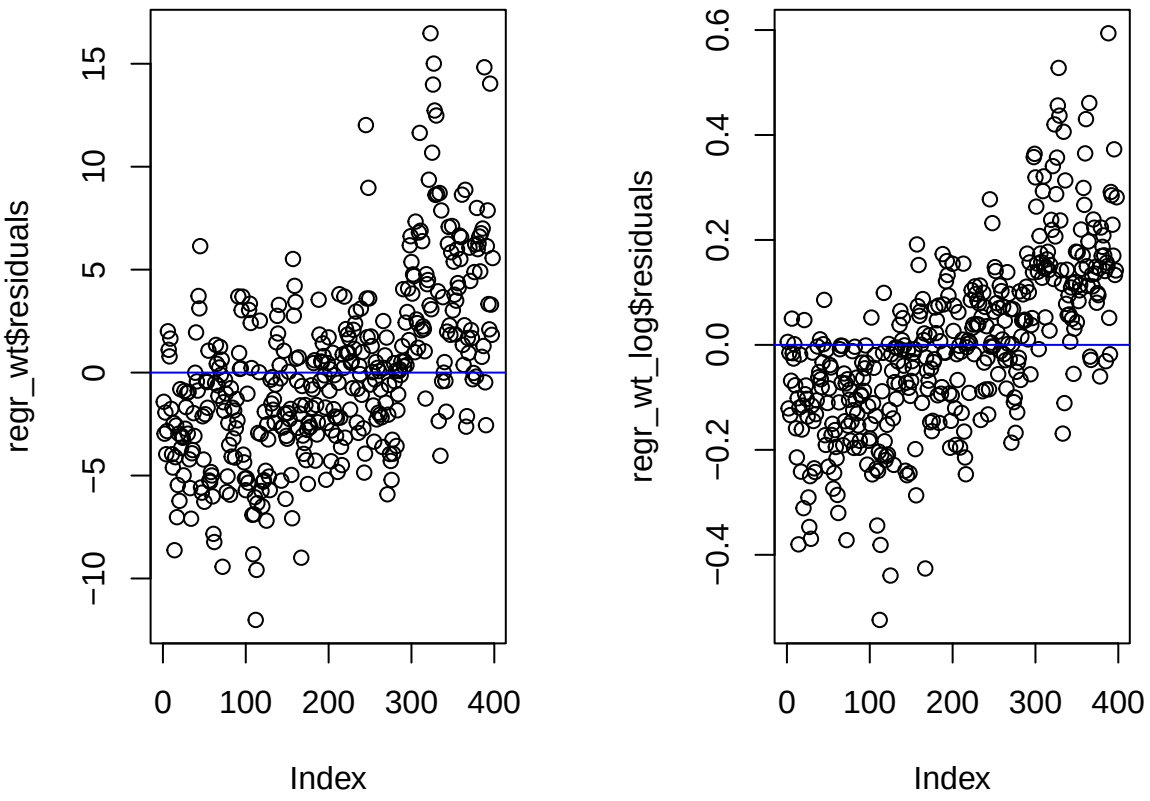
```
par(mfrow = c(1, 2),  mar = c(4, 4, 2, 3))
plot(density(regr_wt$residuals))
abline(v=0, lty="dashed")
plot(density(regr_wt_log$residuals))
abline(v=0, lty="dashed")
```




```
par(mfrow = c(1, 1))
```

- scatter plots of log.weight. vs. residuals

```
par(mfrow = c(1, 2),  mar = c(4, 4, 2, 3))
plot(regr_wt$residuals)
abline(h=0, col = "blue")
plot(regr_wt_log$residuals)
abline(h=0, col = "blue")
```




```
par(mfrow = c(1, 1))
```

iv. Which regression produces better distributed residuals for the assumptions of regression?

- The log transformed regression produces better distributed residuals as the original regression residual density plot shows a little bit of positive skew.

v.How would you interpret the slope of log.weight. vs log.mpg. in simple words?

- The slope is about -1.0583, which indicates that for every 1% increase in vehicle weight, the fuel efficiency (measured in mpg) is expected to decrease by approximately 1.06%, holding other factors constant.

vi.From its standard error, what is the 95% confidence interval of the slope of log.weight. vs log.mpg.?

- The 95% confidence interval of the slope is [-1.1161 , -1.0004], our slope value, -1.0538, has lied in this interval.

```
# Extract slope and standard error
slope <- coef(regr_wt_log)["log.weight."]
se <- summary(regr_wt_log)$coefficients["log.weight.", "Std. Error"]
```

```r
# Compute 95% CI
lower <- slope - 1.96 * se
upper <- slope + 1.96 * se
cat("95% CI for the slope:", round(lower, 4), "to", round(upper, 4))
```

```
## 95% CI for the slope: -1.1161 to -1.0004
```

Question 2 Let's tackle multicollinearity next. Consider the regression model:

a. Using regression and R2, compute the VIF of log.weight. using the approach shown in class

```r
weight_regr <- lm(log.weight. ~ log.cylinders. + log.displacement. + log.horsepower. + log.acceleration
+ model_year + origin, data = cars_log)

r2_weight <- summary(weight_regr)$r.squared

vif_weight <- 1 / (1 - r2_weight)

vif_weight
```

```
## [1] 16.12112
```

```r
sqrt(vif_weight)
```

```
## [1] 4.015111
```

b. Let's try a procedure called Stepwise VIF Selection to remove highly collinear predictors. Start by Installing the 'car' package in RStudio – it has a function called vif() (note: CAR package stands for "Companion to Applied Regression" – it isn't about cars!)

```r
library(car)
```

```
## Loading required package: carData
```

```r
regr_log_origin <- lm(log.mpg. ~ log.cylinders. + log.displacement. + log.horsepower. +
                              log.weight. + log.acceleration. + model_year +
                              factor(origin), data=cars_log)
```

i. Use vif(regr_log) to compute VIF of the all the independent variables

```r
vif(regr_log_origin)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## log.cylinders.    10.456738  1        3.233688
## log.displacement. 29.625732  1        5.442952
## log.horsepower.   12.132057  1        3.483110
## log.weight.       17.575117  1        4.192269
## log.acceleration.  3.570357  1        1.889539
## model_year         1.303738  1        1.141814
## factor(origin)     2.656795  2        1.276702
```

ii. Eliminate from your model the single independent variable with the largest VIF score that is also greater than 5

- log.displacement. will be eliminated due to 5.442952 VIF score.

```
regr_log <- lm(log.mpg. ~ log.cylinders. + log.horsepower. +
                              log.weight. + log.acceleration. + model_year +
                              factor(origin), data=cars_log)
```

iii. Repeat steps (i) and (ii) until no more independent variables have VIF scores above 5

- After log.displacement. is eliminated, there's no more independent variables have VIF scores above 5.

```
vif(regr_log_origin)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## log.cylinders.    10.456738  1        3.233688
## log.displacement. 29.625732  1        5.442952
## log.horsepower.   12.132057  1        3.483110
## log.weight.       17.575117  1        4.192269
## log.acceleration.  3.570357  1        1.889539
## model_year         1.303738  1        1.141814
## factor(origin)     2.656795  2        1.276702
```

iv. Report the final regression model and its summary statistics

```
regr_log <- lm(log.mpg. ~ log.cylinders. + log.horsepower. +
                              log.weight. + log.acceleration. + model_year +
                              factor(origin), data=cars_log)

summary(regr_log)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.cylinders. + log.horsepower. + log.weight. +
##     log.acceleration. + model_year + factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40059 -0.06820  0.00484  0.06208  0.39096
```

```
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.26400    0.34469  21.074  < 2e-16 ***
## log.cylinders.    -0.06712    0.04400  -1.525   0.1280
## log.horsepower.   -0.28552    0.05784  -4.937 1.19e-06 ***
## log.weight.       -0.57510    0.06803  -8.454 5.94e-16 ***
## log.acceleration. -0.17510    0.05752  -3.044   0.0025 **
## model_year         0.03018    0.00176  17.143  < 2e-16 ***
## factor(origin)2    0.04717    0.01826   2.582   0.0102 *
## factor(origin)3    0.04394    0.01834   2.396   0.0171 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1128 on 384 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.8919, Adjusted R-squared:  0.8899
## F-statistic: 452.5 on 7 and 384 DF,  p-value: < 2.2e-16
```

c. Using stepwise VIF selection, have we lost any variables that were previously significant? If so, how much did we hurt our explanation by dropping those variables? (hint: look at model fit)

- We lost displacement by using VIF selection, this variable was significant in original regression model.

```
cat("Explanation hurt by dropping displacement:",summary(regr_log_origin)$r.squared - summary(regr_log)$
```

```
## Explanation hurt by dropping displacement: 3.443087e-05
```

d. From only the formula for VIF, try deducing/deriving the following:

i. If an independent variable has no correlation with other independent variables, what would its VIF score be?

- If an independent variable has no correlation with other independent variables, its VIF score is 1. This is because the R^2 is 0 since other independent variable can explain this independent variable, and the VIF score will be:

$$\Rightarrow \text{VIF}_j = \frac{1}{1-0} = 1$$

ii. Given a regression with only two independent variables (X1 and X2), how correlated would X1 and X2 have to be, to get VIF scores of 5 or higher? To get VIF scores of 10 or higher?

- Since there are only two predictors, the squared correlation r^2 between X_1 and X_2 is equal to R^2 from regressing one on the other. Therefore:

$$\text{VIF} = \frac{1}{1-r^2} \quad \Rightarrow \quad r = \sqrt{1 - \frac{1}{\text{VIF}}}$$

- To achieve a VIF of 5 or higher:

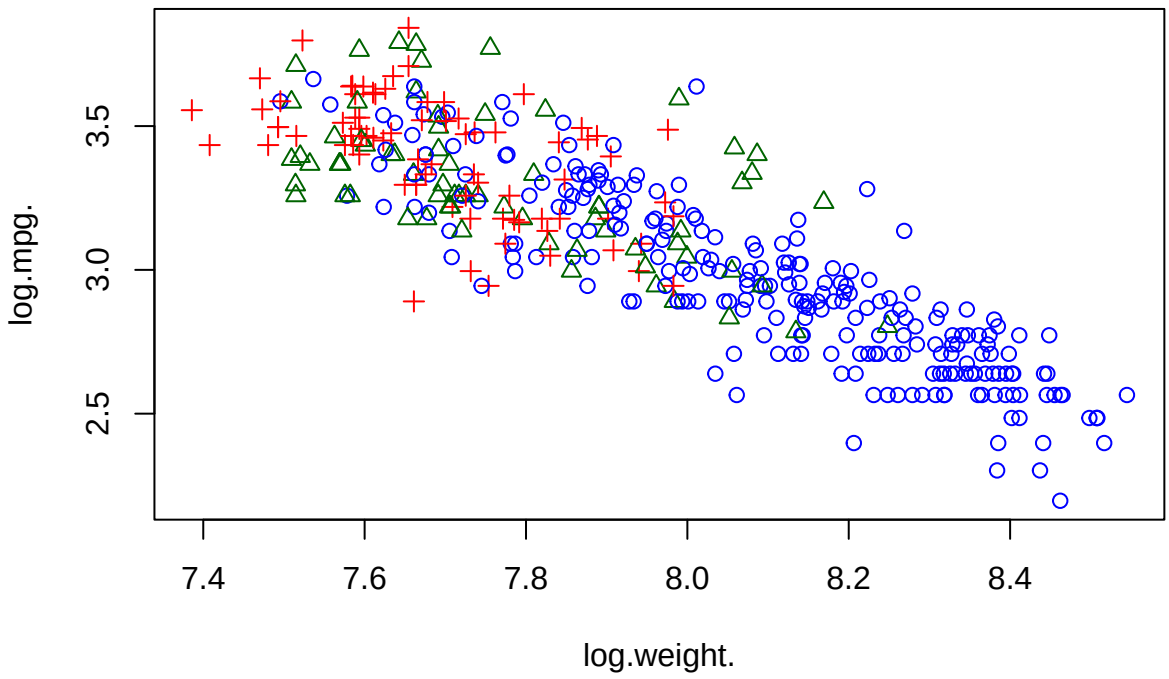
$$r = \sqrt{1 - \frac{1}{5}} = \sqrt{0.8} \approx 0.894$$

- To achieve a VIF of 10 or higher:

$$r = \sqrt{1 - \frac{1}{10}} = \sqrt{0.9} \approx 0.949$$

Question 3 Might the relationship of weight on mpg be different for cars from different origins? Let's try visualizing this. First, plot all the weights, using different colors and symbols for the three origins:

```
origin_colors = c("blue", "darkgreen", "red")
with(cars_log, plot(log.weight., log.mpg., pch=origin, col=origin_colors[origin]))
```



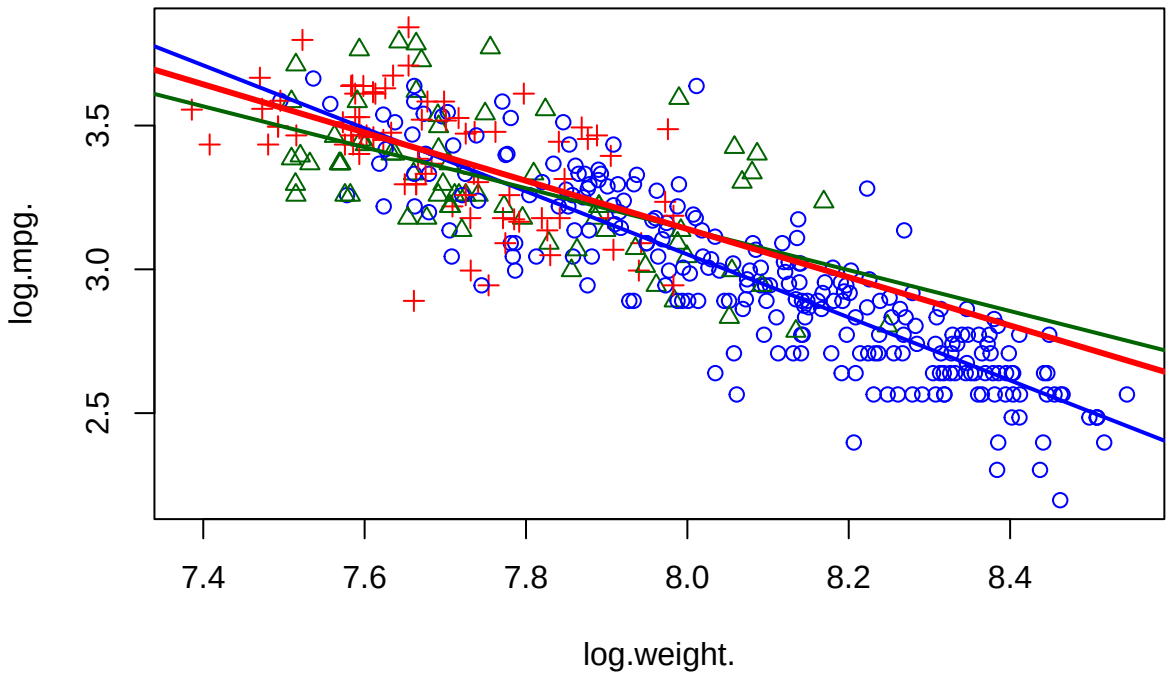
a. Let's add three separate regression lines on the scatterplot, one for each of the origins. Here's one for the US to get you started:

```r
origin_colors = c("blue", "darkgreen", "red")
with(cars_log, plot(log.weight., log.mpg., pch=origin, col=origin_colors[origin]))

# US cars
cars_us <- subset(cars_log, origin==1)
wt_regr_us <- lm(log.mpg. ~ log.weight., data=cars_us)
abline(wt_regr_us, col=origin_colors[1], lwd=2)

# EU cars
cars_eu <- subset(cars_log, origin == 2)
wt_regr_eu <- lm(log.mpg. ~ log.weight., data = cars_eu)
abline(wt_regr_eu, col = origin_colors[2], lwd = 2)

# JP cars
cars_jp <- subset(cars_log, origin == 3)
wt_regr_jp <- lm(log.mpg. ~ log.weight., data = cars_jp)
abline(wt_regr_jp, col = origin_colors[3], lwd = 3)
```



b. [not graded] Do cars from different origins appear to have different weight vs. mpg relationships?

- Yes, cars from different origins do appear to have different weight vs. mpg relationships, as suggested by the distinct regression lines:

- (1) US cars (blue) show a steeper negative slope, indicating that weight has a stronger negative effect on fuel efficiency (log.mpg) for US-made vehicles.
- (2) European cars (green) have a shallower slope, suggesting that weight plays a less pronounced role in determining mpg for these models.
- (3) Japanese cars (red) fall somewhere in between, with a moderate negative slope.
- This implies that vehicle design philosophies across regions (e.g., engine tuning, aerodynamics, vehicle size) may influence how weight impacts fuel efficiency. The separation of trend lines also visually supports this distinction.