BACS HW11

Student ID: 110065508

Due: 2022/05/01

Special thanks to 108071001

Question 1. Read the dataset with the following code:

```
# Question 1
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) Using the `lm()` function,

```
# Question 1 (a)
regr <- lm(log.mpg. ~ log.cylinders. + log.displacement. + log.horsepower. +
                      log.weight. + log.acceleration. + model_year +
                      factor(origin),
                      data=cars_log,
                      na.action=na.exclude)
```

(i) Use the command

`> summary(regr)`

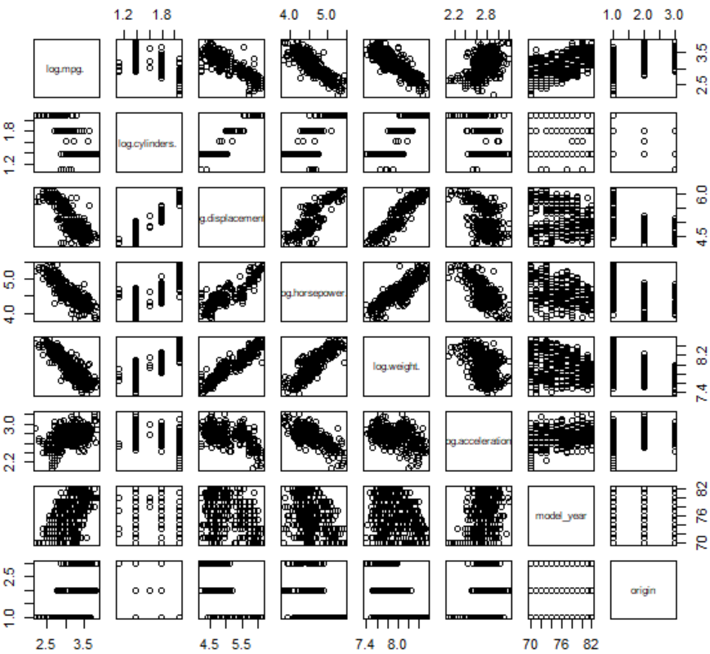
The following log-transformed factors have a significant effect on log.mpg. at 10% significance:

- `log.horsepower.`
- `log.weight.`
- `log.acceleration`
- `model_year`
- `factor(origin)2`
- `factor(origin)3`

(ii) Using the results from Question 3(b), Homework 10, the dependent variables `horsepower` and `acceleration` are now having effects after taking log transform. That's because the data becomes more linear.

(iii) Based on the scatter plot generated by

`> plot(cars_log)`

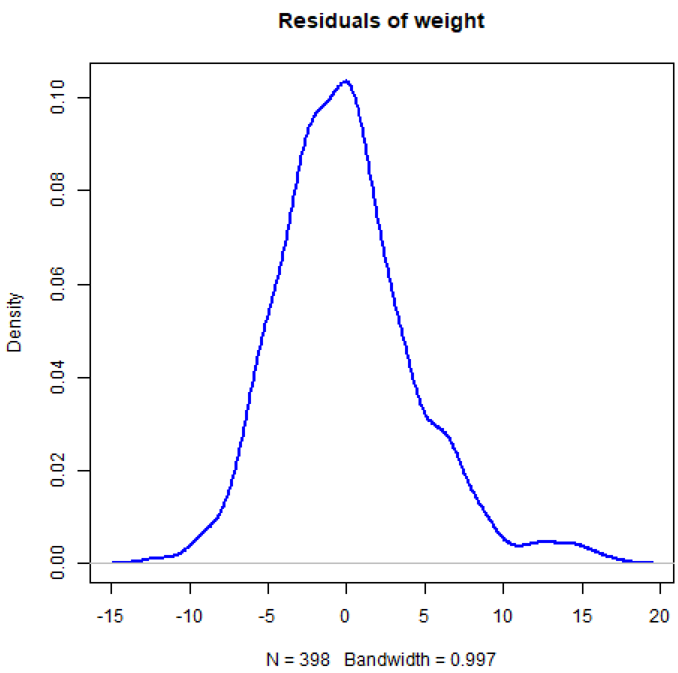


The dependent variables `log.cylinders.` is still insignificant. From correlation, `log.horsepower.` and `log.weight.` have opposite effects to `log.acceleration` and `model_year`. This may be a domain knowledge problem and I'm not familiar to cars. However I think the heavier a car is, the tougher for it to accelerate.

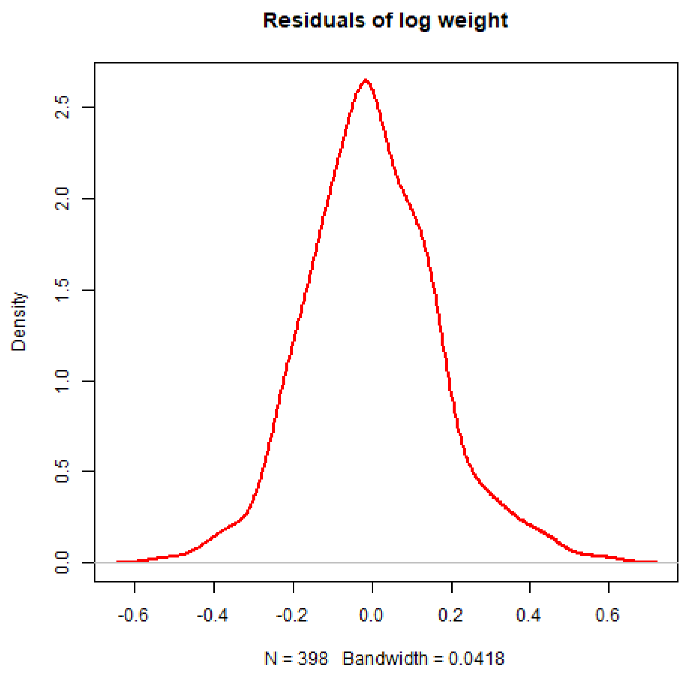
(b) Create two regression models and plots.

```r
# Question 1 (b)
regr_wt = lm(mpg ~ weight, data=cars, na.action=na.exclude)
regr_wt_log = lm(log.mpg. ~ log.weight., data=cars_log, na.action=na.exclude)

png(filename = "1b-1.png")
plot(density(regr_wt$residuals),
     main="Residuals of weight",
     col="blue", lwd=2)
dev.off()

png(filename = "1b-2.png")
plot(density(regr_wt_log$residuals),
     main="Residuals of log weight",
     col="red", lwd=2)
dev.off()

png(filename = "1b-3.png")
plot(cars$mpg, resid(regr_wt),
     col="blue", main="Residuals vs weight", lwd=2)
abline(h=0)
dev.off()

png(filename = "1b-4.png")
plot(cars_log$log.mpg., resid(regr_wt_log),
     col="red", main="Residuals of log weight", lwd=2)
abline(h=0)
dev.off()
```

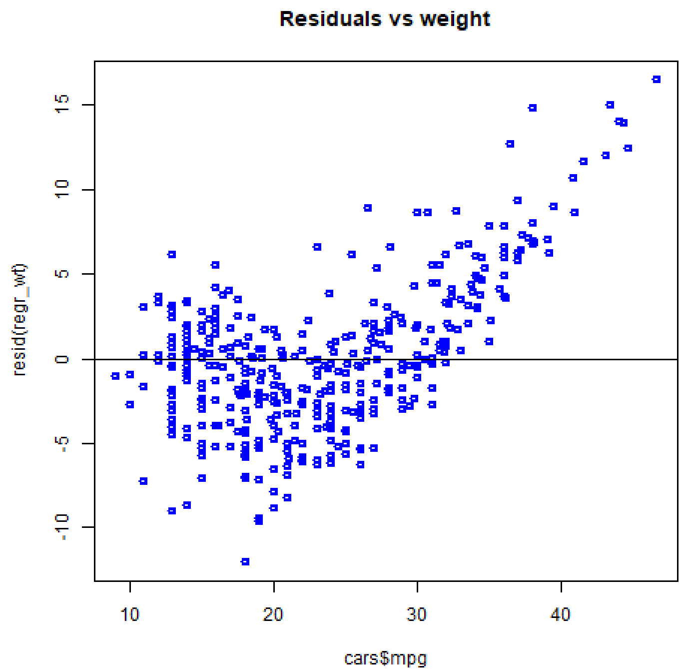
(iii) The density plots of residuals:



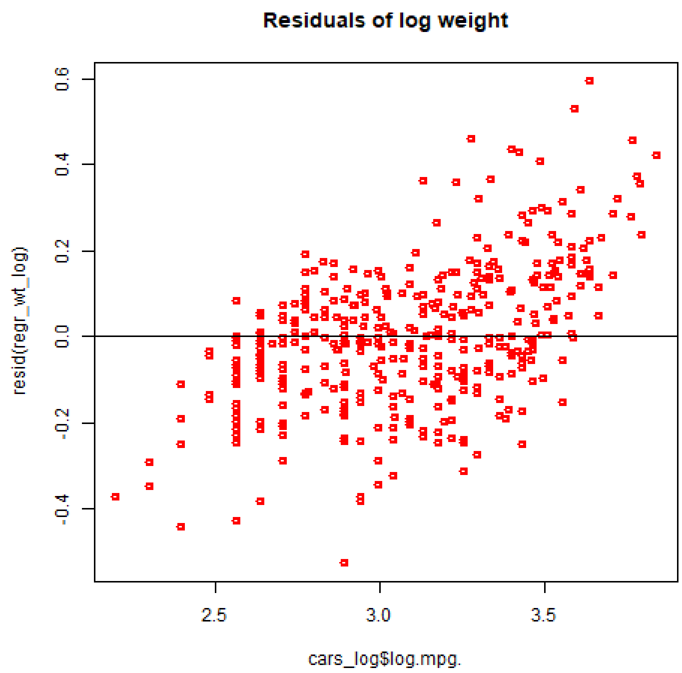
The density plots of log residuals:



The scatter plot of `weight.` vs. residuals:



The scatter plot of `log.weight.` vs. residuals:

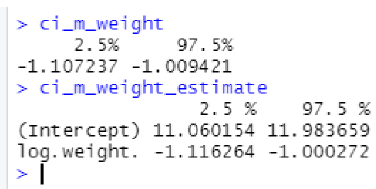


(iv) The regression of `log.mpg.` on `log.weight.` (`regr_wt_log`) produces better distributed residuals since it is more symmetric and closed to a mormal distribution.

(v) Using the command `summary(regr_wt_log)`, I found that the slope is -1.0583. That means 1% change in weight leads to 0.84 decreasing in mpg.

(c) Bootstrapped codes:

```
# Question 1 (c)
# Function for single resampled regression line
boot_regr <- function(model, dataset) {
  boot_index <- sample(1:nrow(dataset), replace=TRUE)
  data_boot <- dataset[boot_index,]
  regr_boot <- lm(model, data=data_boot)
  regr_boot$coefficients
}
# Bootstrapping for confidence interval
coeffs <- replicate(300, boot_regr(log.mpg. ~ log.weight., cars_log))

# Confidence interval values
ci_m_weight <- quantile(coeffs["log.weight.",], c(0.025, 0.975))

# estimate of coefficient and its standard error
ci_m_weight_estimate <- confint(regr_wt_log)
```



```
> ci_m_weight
     2.5%     97.5%
-1.107237 -1.009421
> ci_m_weight_estimate
                2.5 %    97.5 %
(Intercept) 11.060154 11.983659
log.weight.  -1.116264 -1.000272
>
```

I found that `ci_m_weight` and `ci_m_weight_estimate` is quite closed. They would not be identical since bootstrapping has some randomness. However I can deduct that the result is correct. ■

Question 2. (a) Build the required model first.

```
# Question 2 (a)
regr_weight <- lm(log.weight. ~ log.cylinders. + log.displacement. +
                    log.horsepower. + log.acceleration. + model_year +
```

```r
                   factor(origin), data=cars_log)
r2_weight <- summary(regr_weight)$r.squared
vif_weight <- 1 / (1 - r2_weight)
```

The VIF of `log.weight.` is 17.57512.

(b) I'll show the process of removing independent variables.

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

```
> vif(regr_log)
                       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
> 
> vif(regr_log)
                       GVIF Df GVIF^(1/(2*Df))
log.cylinders.     5.433107  1        2.330903
log.horsepower.   12.114475  1        3.480585
log.weight.       11.239741  1        3.352572
log.acceleration.  3.327967  1        1.824272
model_year         1.291741  1        1.136548
factor(origin)     1.897608  2        1.173685
> 
> vif(regr_log)
                      GVIF Df GVIF^(1/(2*Df))
log.cylinders.    5.321090  1        2.306749
log.weight.       4.788498  1        2.188264
log.acceleration. 1.400111  1        1.183263
model_year        1.201815  1        1.096273
factor(origin)    1.792784  2        1.157130
> 
> vif(regr_log)
                      GVIF Df GVIF^(1/(2*Df))
log.weight.       1.926377  1        1.387940
log.acceleration. 1.303005  1        1.141493
model_year        1.167241  1        1.080389
factor(origin)    1.692320  2        1.140567
> 
```

```
> summary(regr_log)

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

> |
```

(c) The variable `log.horsepower.` is lost. The R^2 drops from 0.8919 (by results of 1(a)) to 0.8856 (by viewing the suuamry of `regr_log`).

(d) Recall that the formula of VIF_j is

$$\text{VIF}_j = \frac{1}{1 - R_j^2}.$$

(i) If an independent variable X_j has no correlation with other independent variables, then $R_j = 0$. Then $\text{VIF}_j = 1$.

(ii) Since

$$\text{Cor}(X, Y) = \sqrt{R_j^2} = \sqrt{1 - \frac{1}{\text{VIF}}},$$

when $\text{VIF}_j > 5$,

$$\text{Cor}(X, Y) > \sqrt{1 - \frac{1}{5}} \approx 0.8944.$$

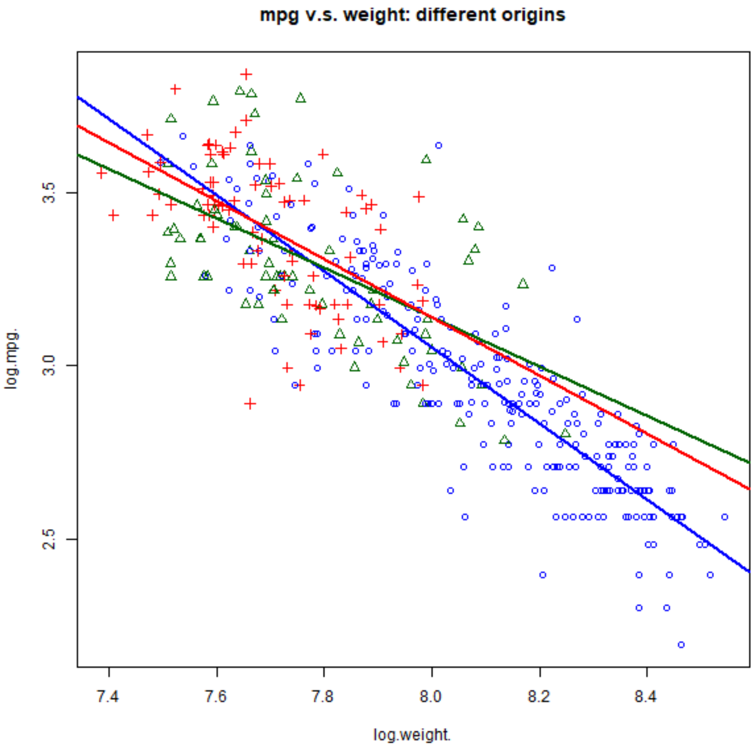
If $\text{VIF}_j > 10$,

$$\text{Cor}(X, Y) > \sqrt{1 - \frac{1}{10}} \approx 0.9486.$$

■

Question 3. Plot all the weights and add three separate regression lines on the scatter plot, one for each of the origins.

```r
# Question 3
png(filename = "3.png", width = 600, height = 600) # Subplots
origin_colors = c("blue", "darkgreen", "red")
with(cars_log, plot(log.weight.,
                    log.mpg.,
                    pch=origin,
                    main = "mpg v.s. weight: different origins",
                    col=origin_colors[origin]))

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)
cars_us_2 <- subset(cars_log, origin==2)
wt_regr_us <- lm(log.mpg. ~ log.weight., data=cars_us_2)
abline(wt_regr_us, col=origin_colors[2], lwd=2)
cars_us_3 <- subset(cars_log, origin==3)
wt_regr_us <- lm(log.mpg. ~ log.weight., data=cars_us_3)
abline(wt_regr_us, col=origin_colors[3], lwd=2)
dev.off()
```



So it looks like cars from different origins are having different weight vs. mpg relationships.

■