HW12

106022113

5/12/2021

Question 1. Deal with nonlinearity

```
cars <- read.table("auto-data.txt",header = FALSE, na.strings = "?")
names(cars)<- c("mpg","cylinders","displacement","horsepower","weight","acceleration","model_ye
ar","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 with cars_log dataset, with mpg dependent

```
regr <- lm(log.mpg. ~ log.cylinders.+log.displacement.+log.horsepower.+log.weight.+log.accelera
tion.+model_year+factor(origin), data = cars_log)
summary(regr)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.cylinders. + log.displacement. +
##     log.horsepower. + log.weight. + log.acceleration. + model_year +
##     factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39727 -0.06880  0.00450  0.06356  0.38542
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.301938   0.361777  20.184  < 2e-16 ***
## log.cylinders.    -0.081915   0.061116  -1.340  0.18094
## log.displacement.  0.020387   0.058369   0.349  0.72707
## log.horsepower.   -0.284751   0.057945  -4.914 1.32e-06 ***
## log.weight.       -0.592955   0.085165  -6.962 1.46e-11 ***
## log.acceleration. -0.169673   0.059649  -2.845  0.00469 **
## model_year         0.030239   0.001771  17.078  < 2e-16 ***
## factor(origin)2    0.050717   0.020920   2.424  0.01580 *
## factor(origin)3    0.047215   0.020622   2.290  0.02259 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.113 on 383 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.8919, Adjusted R-squared:  0.8897
## F-statistic:   395 on 8 and 383 DF,  p-value: < 2.2e-16
```

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

ANSWER : According to the summary, *horsepower, weight, acceleration, model_year, and origin* have a significant effect with p-value lower than 10%.

ii. Do some new factors have effect on mpg, why?

ANSWER: Comparing to our homework last week, we can discover that *horsepower, acceleration* has become significant after taken the log of it. It can be explained that they are non-linear variables, so they won't perform well in regression without preprocessing. After log-transforming, the results are nice.

iii. Which factors still have insignificant or opposite effect on mpg, why?

ANSWER: *Cylinders and displacement* are insignificant on mpg, it might because the distribution of cylinders is irregular and the displacement value shared a more disproportionate feature with mpg. Also, they may share high multicollinearity.

b. Take a look at weight

i.Create a regression of mpg on weight from the original dataset

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

```
##
## 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
```

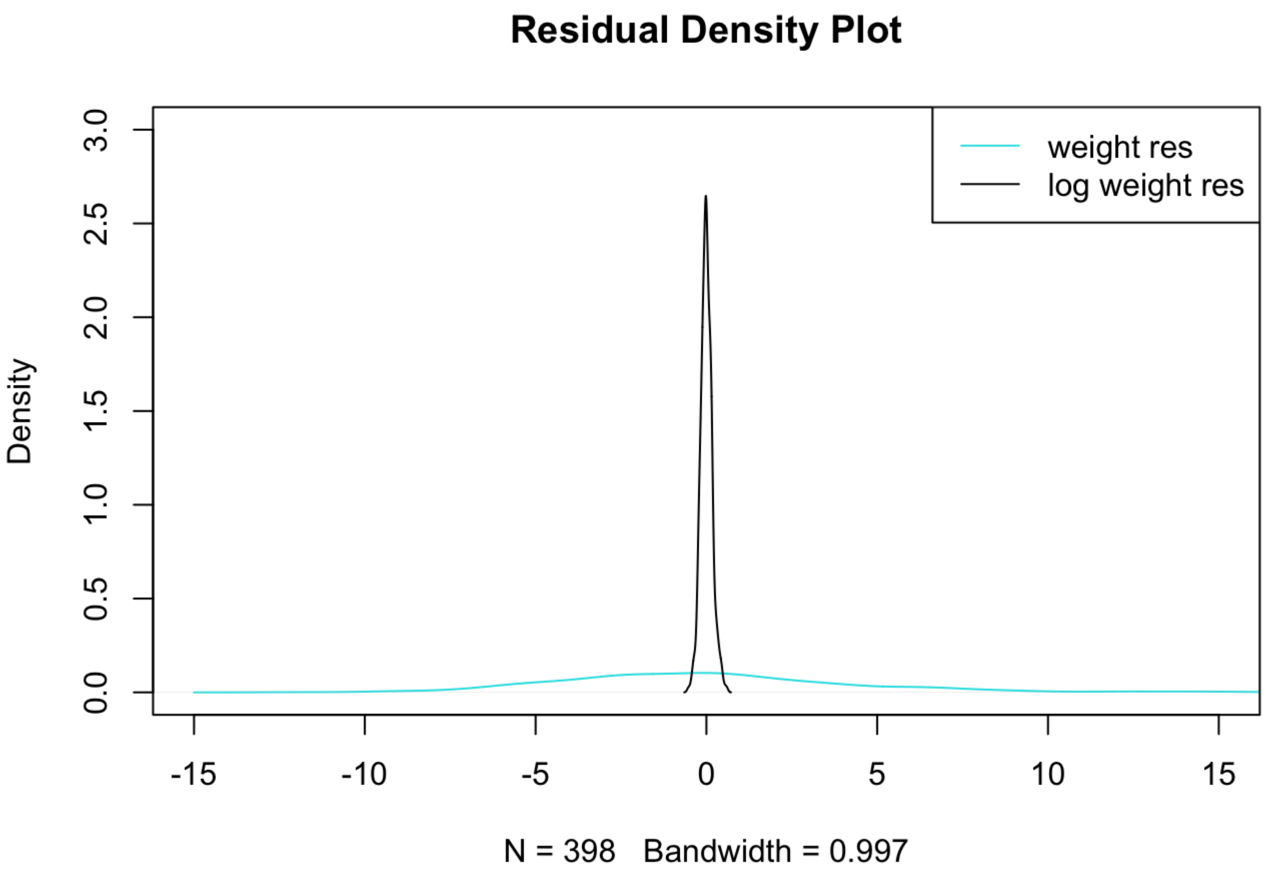
ii . Create a regression of log.mpg. on log.weight. from cars_log

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

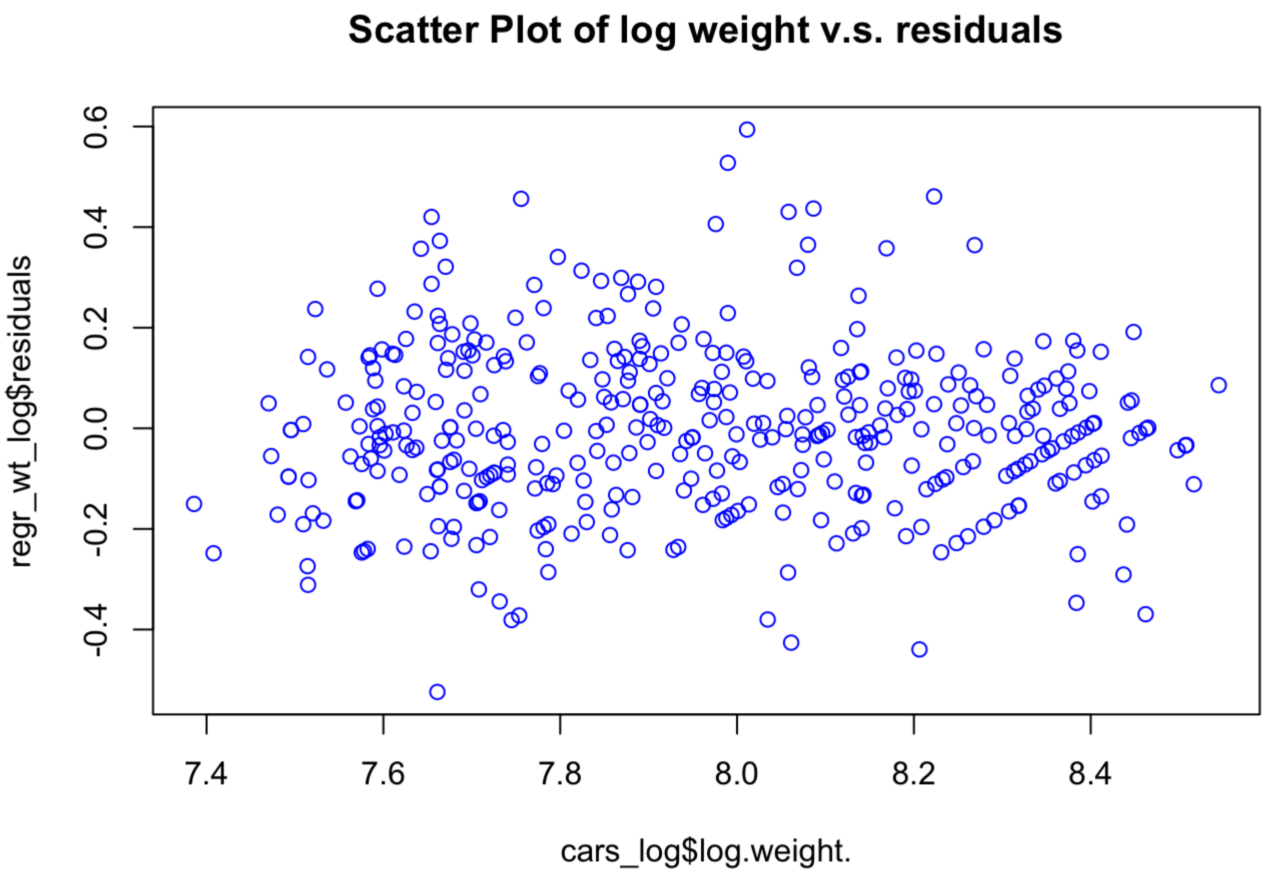
```
##
## Call:
## lm(formula = log.mpg. ~ log.weight., data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52408 -0.10441 -0.00805  0.10165  0.59384
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.5219     0.2349   49.06   <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 regressions

```
plot(density(regr_wt$residuals),ylim = c(0,3),xlim = c(-15,15),col = 5,main = "Residual Density Plot")
lines(density(regr_wt_log$residuals),col = 1)
legend("topright",c("weight res", "log weight res"),lty = c(1,1),col =c(5,1))
```




```
plot(cars_log$log.weight.,regr_wt_log$residuals,col = 'blue', main = "Scatter Plot of log weight v.s. residuals")
```



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

ANSWER: Observing the density plot of residuals before and after log-transformation, we can see that after taken the log the residuals are centralized better than the without log, hence produces better residuals for assumptions of regression.

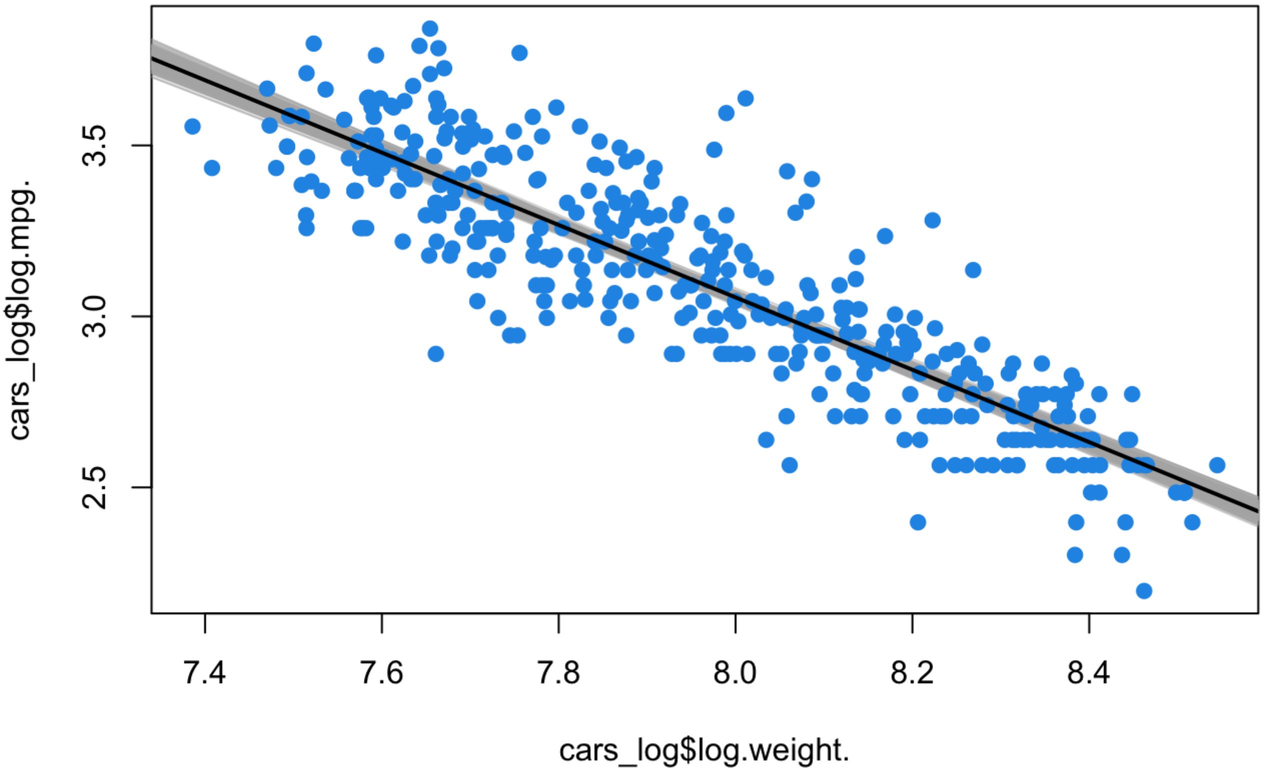
v. How would you interpret the slope of log.weight. vs log.mpg.?

ANSWER: We can acquire the slope by the above summary. Hence it can be interpreted as with 1 percent increase in weight causes -1.05 percent increase in mpg.

c. What is the 95% confidence interval of the slope of log.weight. vs log.mpg.?

i. Create a bootstrapped confidence interval

```
plot(cars_log$log.weight.,cars_log$log.mpg.,col = NA,pch = 19)
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)
  abline(regr_boot, lwd =1, col = rgb(0.7,0.7,0.7,0.5))
  regr_boot$coefficients
}
coeffs <- replicate(300,boot_regr(log.mpg.~log.weight.,cars_log))
points(cars_log$log.weight.,cars_log$log.mpg.,col = 4, pch = 19)
abline(a = mean(coeffs["(Intercept)",]), b = mean(coeffs["log.weight.",]),lwd = 2)
```



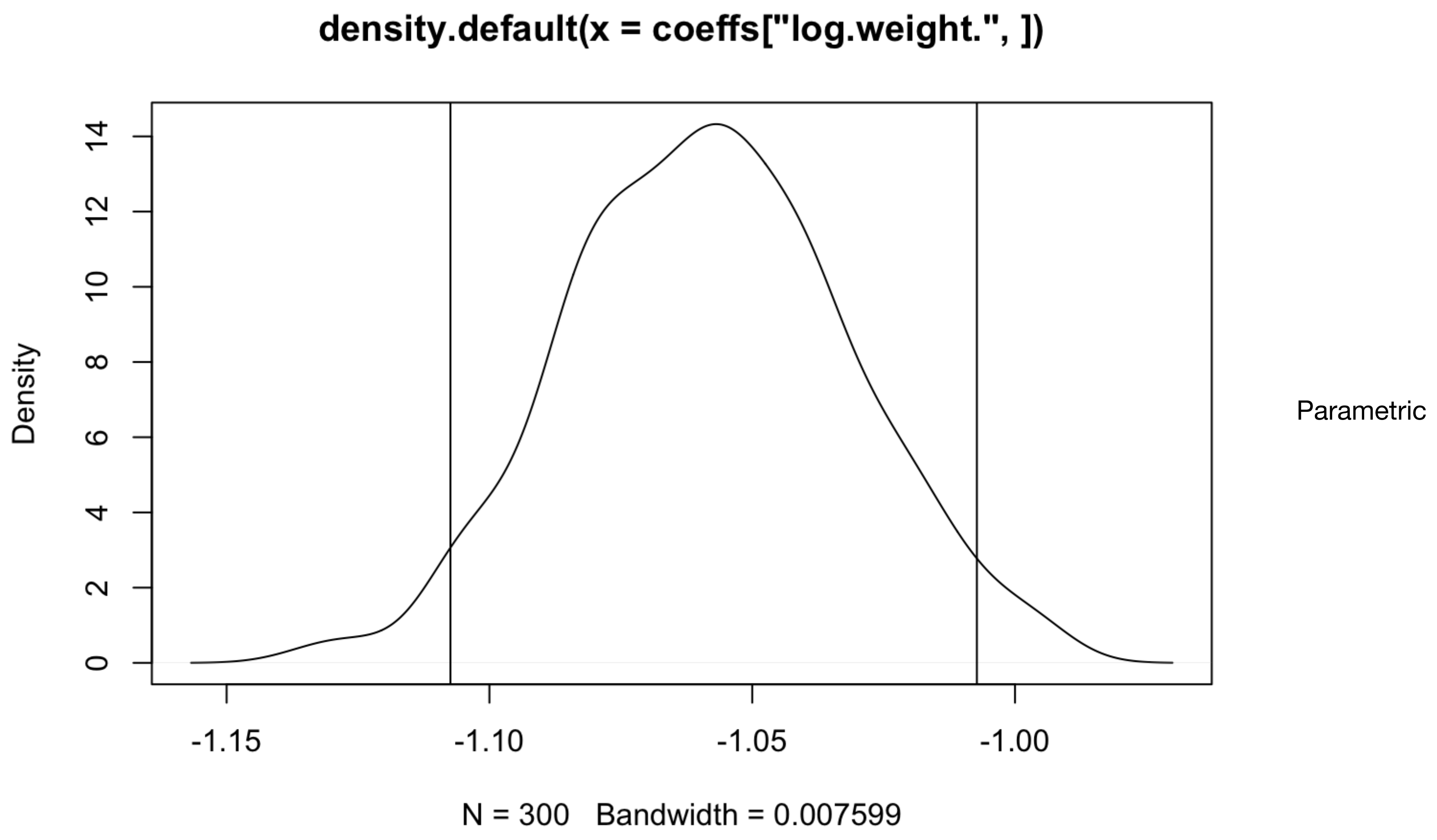
ii. Verify results with confidence interval using traditional methods

```
quantile(coeffs["log.weight.",],c(0.025,0.975))
```

```
##      2.5%     97.5% 
## -1.107420 -1.007253
```

Confidence Interval Plot

```
plot(density(coeffs["log.weight.",]))
abline(v =quantile(coeffs["log.weight.",],c(0.025,0.975)))
```



Confidence Intervals

```
hp_regr_log <- lm(log.mpg.~log.weight.,cars_log)
confint(hp_regr_log)
```

```
##                 2.5 %    97.5 %
## (Intercept) 11.060154 11.983659
## log.weight. -1.116264 -1.000272
```

Question 2. Tackle multicollinearity

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

a. Use regression and R2 and calculate the VIF of log.weight.

```
log_weight <- lm(log.weight. ~ log.cylinders. + log.displacement. + log.horsepower. + log.accel
eration. + model_year + factor(origin),  data=cars_log)
r2_weight <- summary(log_weight)$r.squared
vif_weight <- 1/(1-r2_weight)
paste("weight r2 :",r2_weight, "weight vif:",vif_weight)
```

```
## [1] "weight r2 : 0.943101375320313 weight vif: 17.57511724808"
```

b. Try Stepwise VIF selection to remove highly collinear variables

i. Compute VIF of all independent variables

```r
library(car)
```

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

```r
vif_df <- vif(regr_log)
vif_df
```

```
##                         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. Remove independent variable with largest VIF score greater than 5

```r
#Eliminate Displacement
regr_log1 <- lm(log.mpg. ~ log.cylinders. + log.horsepower. +
                              log.weight. + log.acceleration. + model_year +
                              factor(origin),  data=cars_log)
vif(regr_log1)
```

```
##                         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
```

iii. Repeat i, ii.

```r
#Eliminate horsepower
regr_log2 <- lm(log.mpg. ~ log.cylinders. +
                              log.weight. + log.acceleration. + model_year +
                              factor(origin),  data=cars_log)
vif(regr_log2)
```

```
##                        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
```

```r
##Eliminate cylinders
regr_log3 <- lm(log.mpg. ~    log.weight. + log.acceleration. + model_year +
                              factor(origin),  data=cars_log)
vif(regr_log3)
```

```
##                          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
```

Now only *weight*,*acceleration*,*model_year*, *origin* remains

iv. Report final regression model

```
regr_log3 <- lm(log.mpg. ~    log.weight. + log.acceleration. + model_year +
                              factor(origin),  data=cars_log)
summary(regr_log3)
```

```
##
## 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. Does stepwise VIF selection lost any significant variables?

ANSWER: Yes, stepwise VIF drops *horsepower and weight*. It is reasonable to drop these two variables because the r square value of the full model and the VIF seleciton model is approximately the same.

d. General quesions of VIF

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

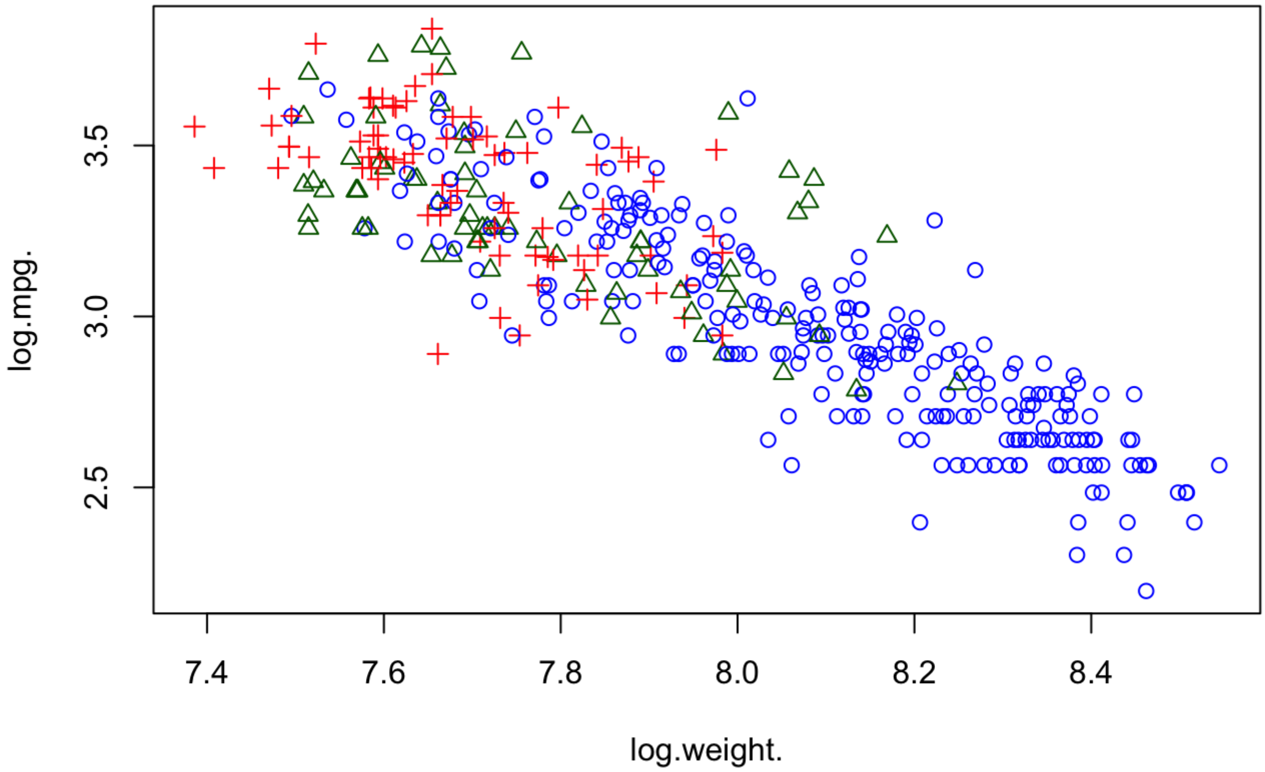
ANSWER: VIF is calculated using the R squared values. If there is no correlation within the variables, then the R sqaured values would be 0 and so it's VIF will be 1.

ii. Regression with 2 independent variables(X1,X2), how correlated would X1, X2 be to get VIF higher than 5, 10?

$$VIF = 1/(1 - r^2)$$
$$r = 0.894(VIF = 5)$$
$$r = 0.949(VIF = 10)$$

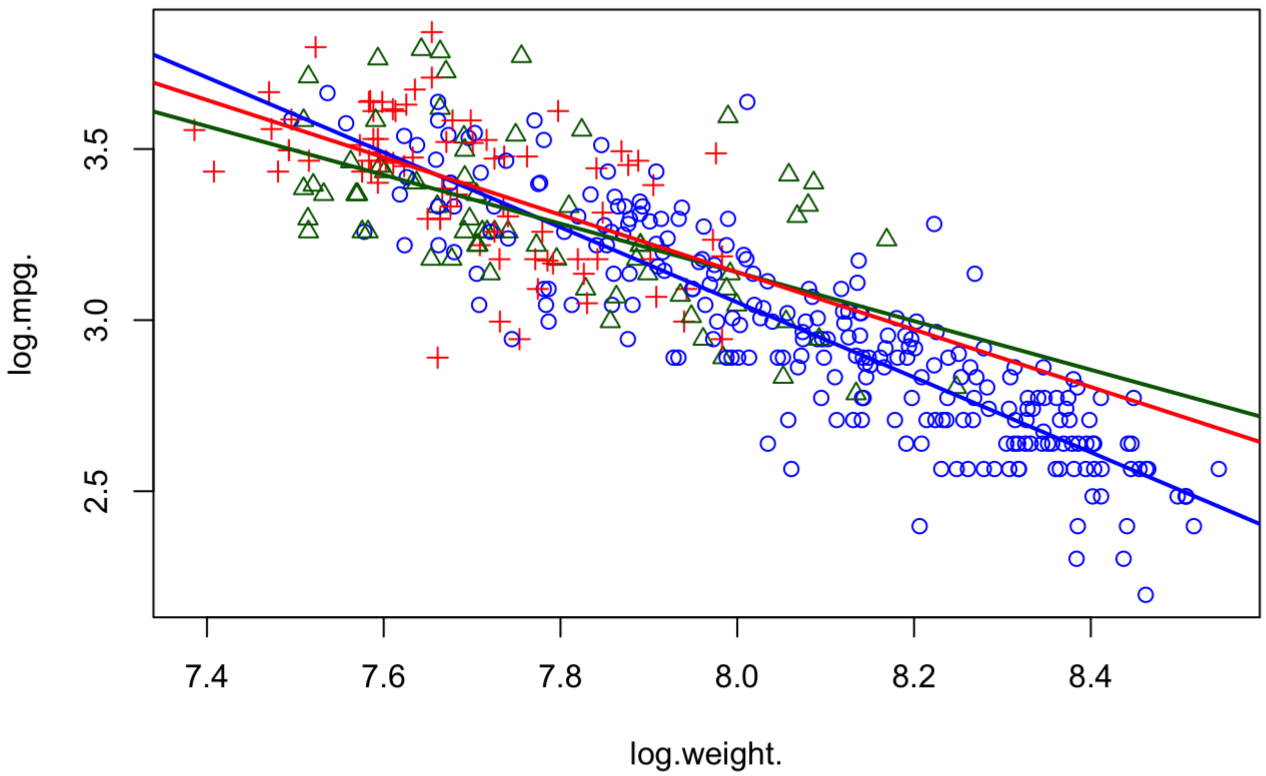
Question 3

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



a.

```
with(cars_log, plot(log.weight., log.mpg., pch=origin, col=origin_colors[origin]))
cars_us <- subset(cars_log, origin==1)
cars_eu <- subset(cars_log, origin==2)
cars_jp <- subset(cars_log, origin==3)
wt_regr_us <- lm(log.mpg. ~ log.weight., data=cars_us)
wt_regr_eu <- lm(log.mpg. ~ log.weight., data=cars_eu)
wt_regr_jp <- lm(log.mpg. ~ log.weight., data=cars_jp)
abline(wt_regr_us, col=origin_colors[1], lwd=2)
abline(wt_regr_eu, col=origin_colors[2], lwd=2)
abline(wt_regr_jp, col=origin_colors[3], lwd=2)
```



b. Do cars from different origins appear to have different weight vs mpg relationships?

ANSWER: The slope of the three regression lines are similar. Hence, the relationship between the two variables for the three countries are also similar. However, the number of data points vary between different countries, hence it may affect the results.