HW 12

109078704

(who helps me: 109078702,109078706,109078510, 109078511)

Question 1) Let's deal with **nonlinearity** first. Create a new dataset that log-transforms several variables from our original dataset (called `cars` in this case):

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

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

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

```
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% significance?

- log.horsepower.
- log.weight.
- log.acceleration.
- model_year
- origin

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

- The assumption of regression was failed to be proved because there are non-linear correlated between horsepower and acceleration.

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

- I assume that because these two factors - "Cylinders" and "displacement" have shown the great multi-collinearity with other variables, both of them have insignificant effects on mpg.

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

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

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

```
Call:
lm(formula = mpg ~ weight, data = auto)

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 (call it `regr_wt_log`) of `log.mpg.` on `log.weight.` from `cars_log`

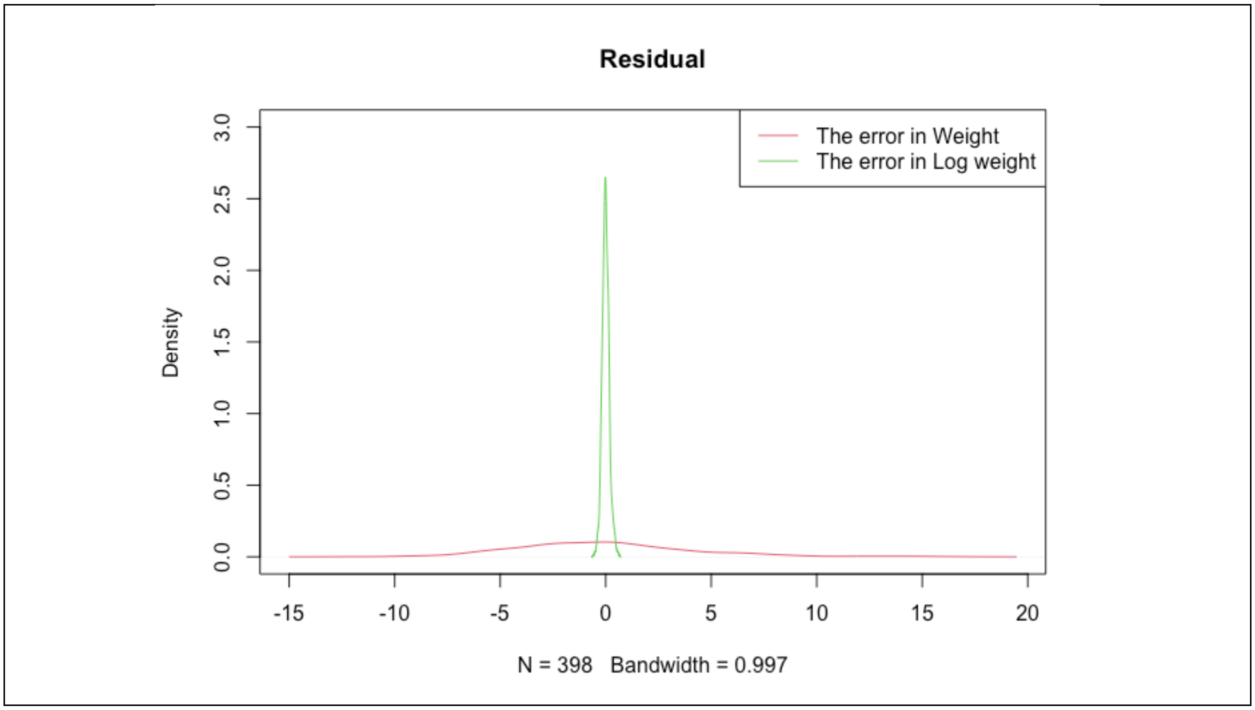
```
regr_wt_log <- lm(log.mpg. ~log.weight., data = cars_log)
summary(regr_wt_log)
```

```
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 <u>both regression models</u> (raw and log-transformed)

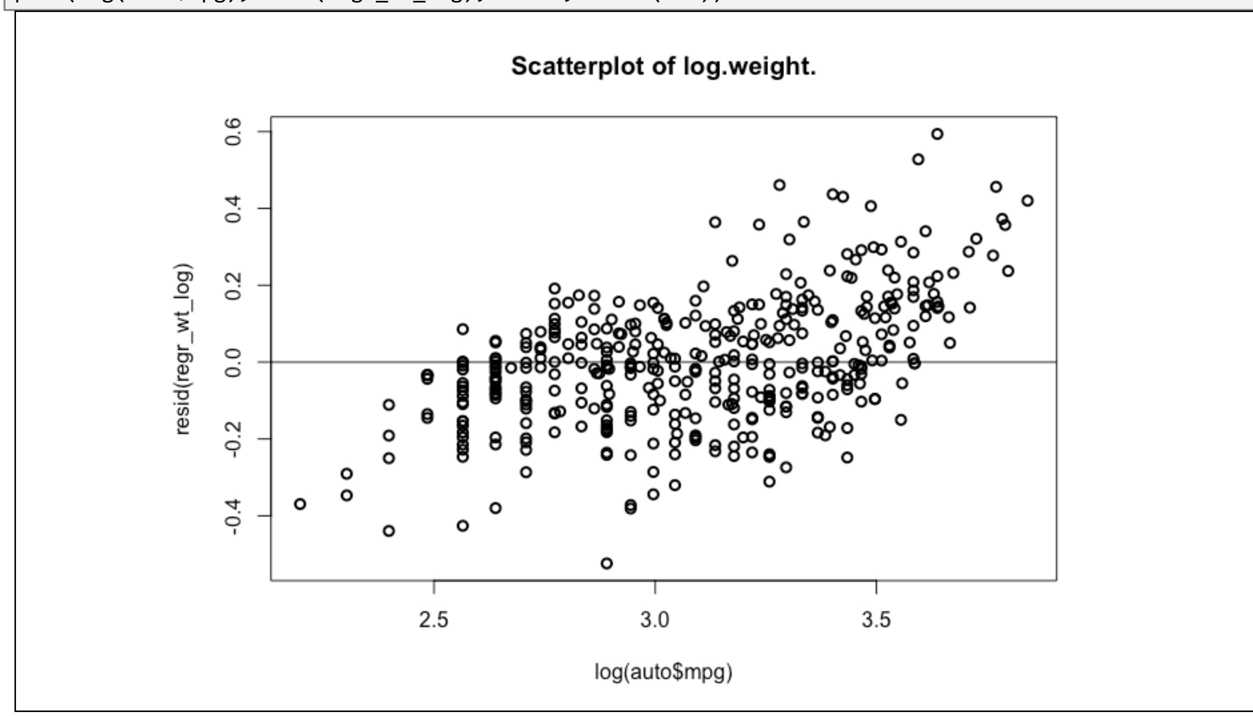
i. density plots of residuals

```
plot(density(regr_wt$residuals), main = "Residual", col = 2,ylim = c(0,3),)
lines(density(regr_wt_log$residuals),col = 3)
legend("topright",c("The error in Weight","The error in Log weight"),lty=c(1,1),col = c(2,3))
```

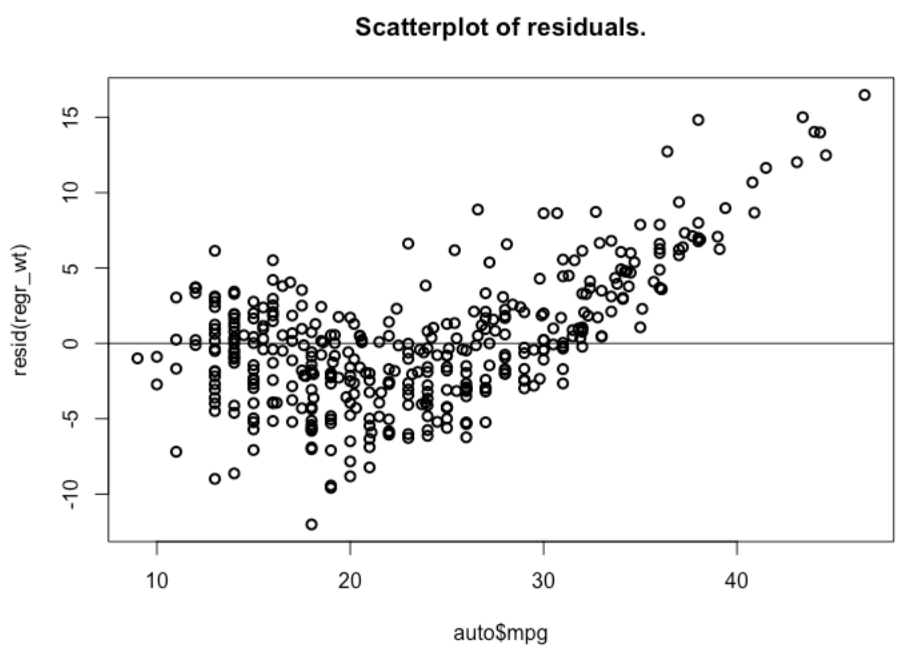


i. scatterplot of log.weight. vs. residuals

```
plot(log(auto$mpg),resid(regr_wt_log), lwd=2,abline(h=0))
```



```
plot(auto$mpg, resid(regr_wt),lwd=2,abline(h=0), main = "Scatterplot of residuals.")
```



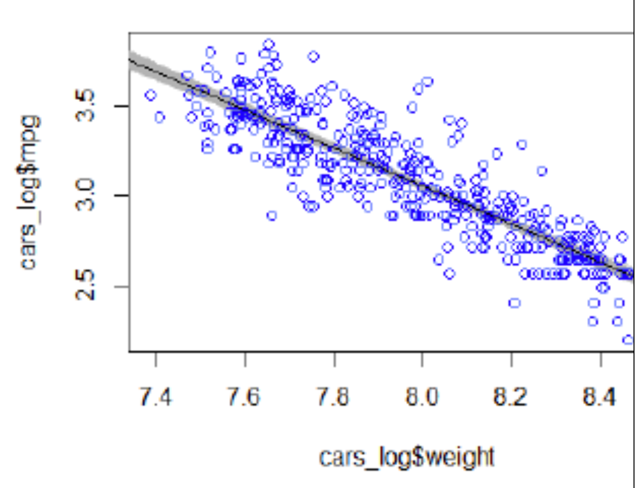
iv. Which regression produces better residuals for the assumptions of regression?
For me, log.weight. is better because its errors is more centralized.

v. How would you interpret the slope of `log.weight.` vs `log.mpg.` in simple words?
Weight might has positive change in around 1% resulting in negative change around -1% in mpg.

c. Let's examine the 95% confidence interval of the *slope* of `log.weight.` vs. `log.mpg.`

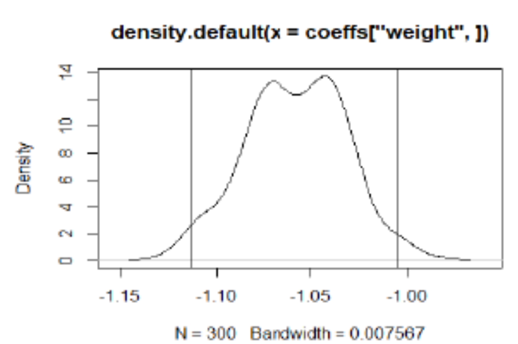
i.Create a *bootstrapped* confidence interval

```
plot(cars_log$weight,cars_log$mpg,col=NA,pch=19)
bootstrap_regr<-function(model,dataset){
  bootstrap_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
}

coeff<-
replicate(300,boot_regr(mpg~weight,cars_log))

points(cars_log$weight,cars_log$mpg,col="blue",pch=1)
abline(a=mean(coeffs["(Intercept)",]),

b=mean(coeffs["weight",]),c(0.025,0.975))
```



ii.Verify your results with a confidence interval using traditional statistics (i.e., estimate of coefficient and its standard error from `lm()` results)

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



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

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

```
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
```

a. Using regression and R^2, compute the VIF of `log.weight.` using the approach shown in class

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

r2_weight<-summary(log.weight_regr)$r.squared
```

```
vif_weight<-1/(1-r2_weight)

sqrt(vif_weight)
```

```
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

> log.weight_regr<-lm(log.weight.~log.cylinders.+log.displacement.+log.horsepower.+log.acceler
ation.+ model_year+factor(origin),data=cars_log,na.action = na.exclude)
> 
> r2_weight<-summary(log.weight_regr)$r.squared
> 
> vif_weight<-1/(1-r2_weight)
> 
> sqrt(vif_weight)
[1] 4.192269
> 
```

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!)

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

```
library(car)
vif<-vif(regr_log)
```

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

```
vif_no_displacement<-
vif(lm(log.mpg.~log.cylinders.+log.horsepower.+log.weight.+log.acceleration.+model_year+factor(origi
n),data=cars_log))
vif_no_displacement
```

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

```
vif_no_cylinders<vif(lm(log.mpg.~log.horsepower.+log.weight.+log.acceleration.+model_year+factor(ori
gin),data=cars_log))
vif_no_cylinders
```

	GVIF	Df	GVIF^(1/(2*Df))
log.horsepower.	12.102217	1	3.478824
log.weight.	8.022686	1	2.832435
log.acceleration.	3.202264	1	1.789487
model_year	1.257618	1	1.121436
factor(origin)	1.781513	2	1.155307

```
vif_no_horsepower<
vif(lm(log.mpg.~log.weight.+log.acceleration.+model_year+factor(origin),data=cars_ log))
vif_no_horsepower
```

```
                          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
```

iv. Report the final regression model and its summary statistics

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

```
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. 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)

```
summary(regr_log)$adj.r.squared
summary(last_regr)$adj.r.squared
```

```
> summary(regr_log)$adj.r.squared
[1] 0.8896522
> summary(last_regr)$adj.r.squared
[1] 0.8841169
```

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?

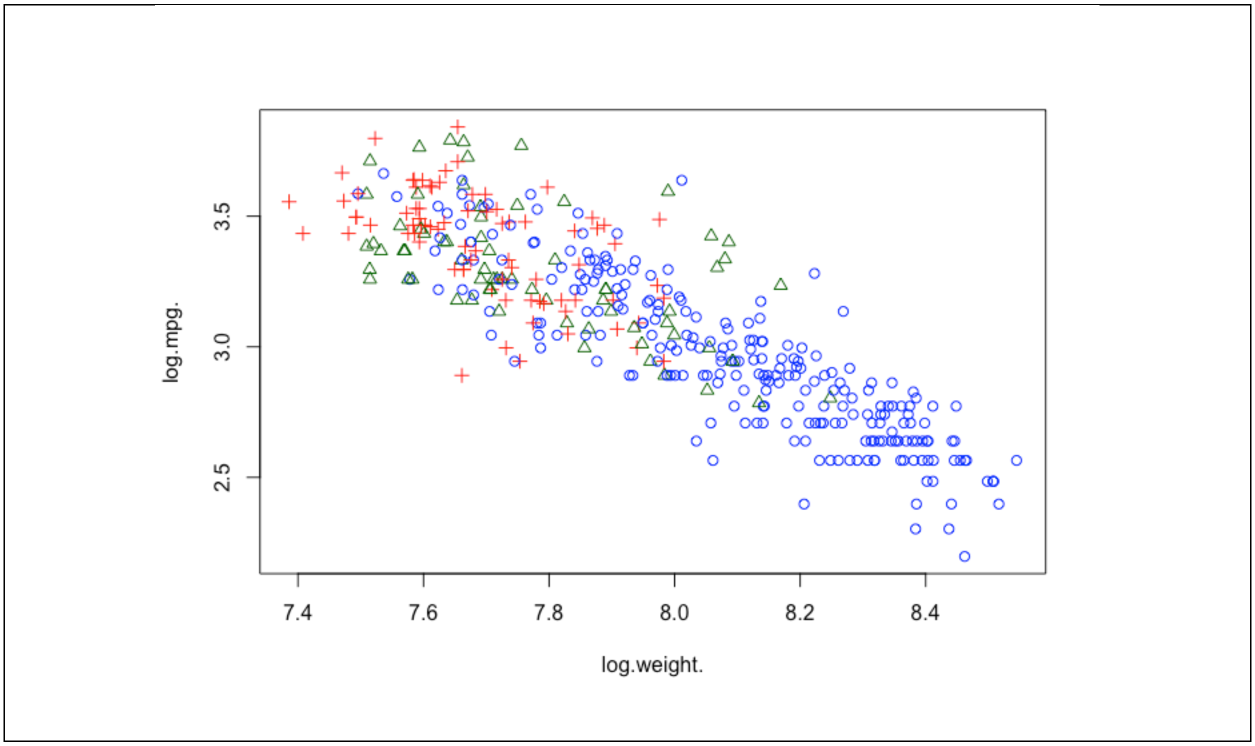
I think VIF will be equal to 1

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?

correlation > 0.894 → VIF scores ≧ 5 or higher.
correlation > 0.949 → VIF scores ≧ 10 or higher.

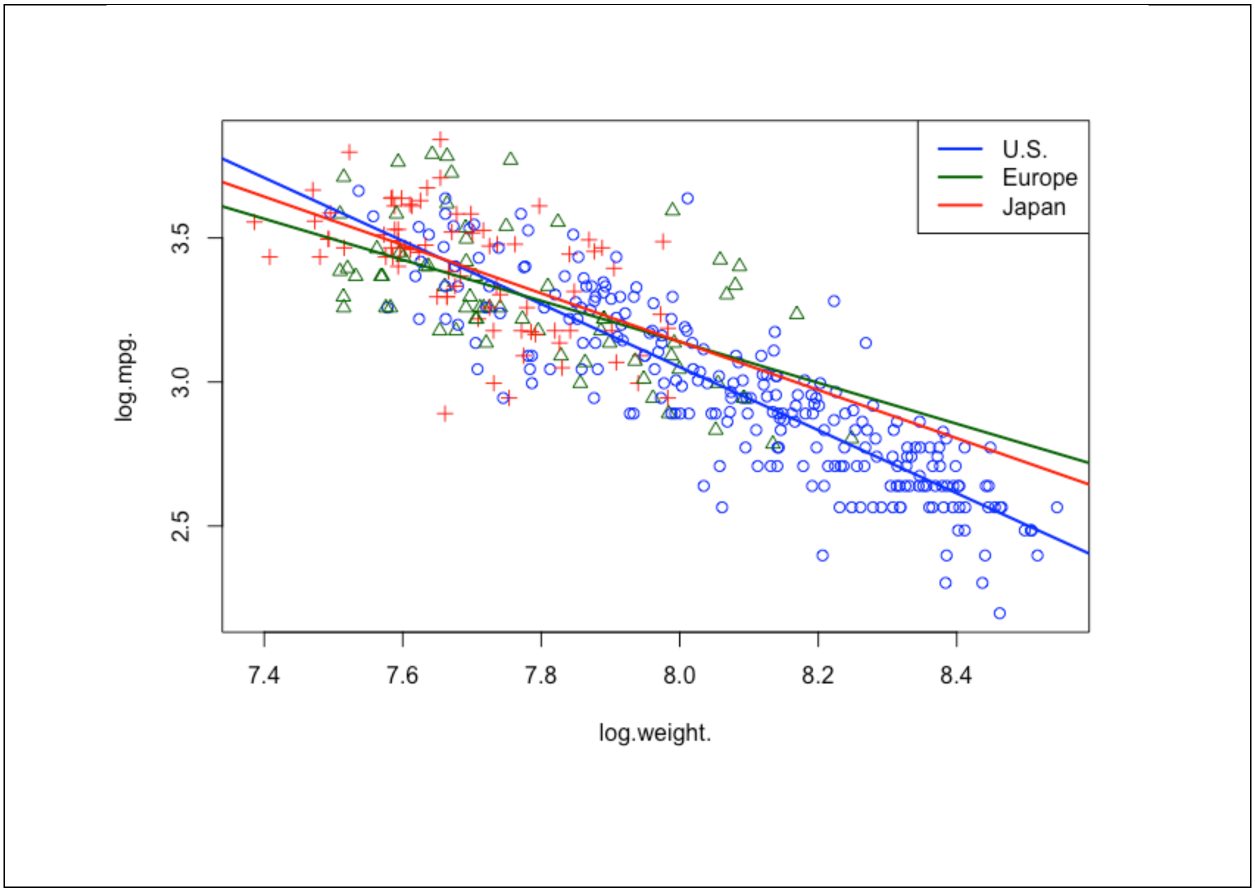
Question 3) Might the relationship of weight on mpg be different for cars from different origins? Let's try visualizing this. First, plot all the weights, using different colors and symbols for the three origins:

```
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:

```
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_europe<-subset(cars_log, origin==2)
wt_regr_europe<-lm(log.mpg.~log.weight.,data=cars_europe)

abline(wt_regr_europe,col=origin_colors[2],lwd=2)

cars_japan<-subset(cars_log,origin==3)
wt_regr_japan<- lm(log.mpg.~log.weight.,data=cars_japan)
abline(wt_regr_japan,col=origin_colors[3],lwd=2)

legend(x="topright",y="topright",legend=c("U.S.","Europe","Japan"),col=c("blue","darkgreen","red"),l
wd=c(2,2,2))
```



U.S.
Europe
Japan
log.mpg.
3.5
3.0
2.5
7.4
7.6
7.8
8.0
8.2
8.4
log.weight.