BACS - HW 11   106073401

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):

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

```
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))
```

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

```
summary(with(cars_log,lm(log.mpg.~log.cylinders.+log.displacement.+log.horsepow
er.+log.weight.+log.acceleration.+model_year+factor(origin))))

                   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 *
```

Horsepower, weight, acceleration and model year.

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

Horsepower and acceleration have effects now.

iii. Which factors still have insignificant or opposite effect on mpg? Why might this be?

Displacement and cylinders still have insignificant effect.

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 = cars)
```

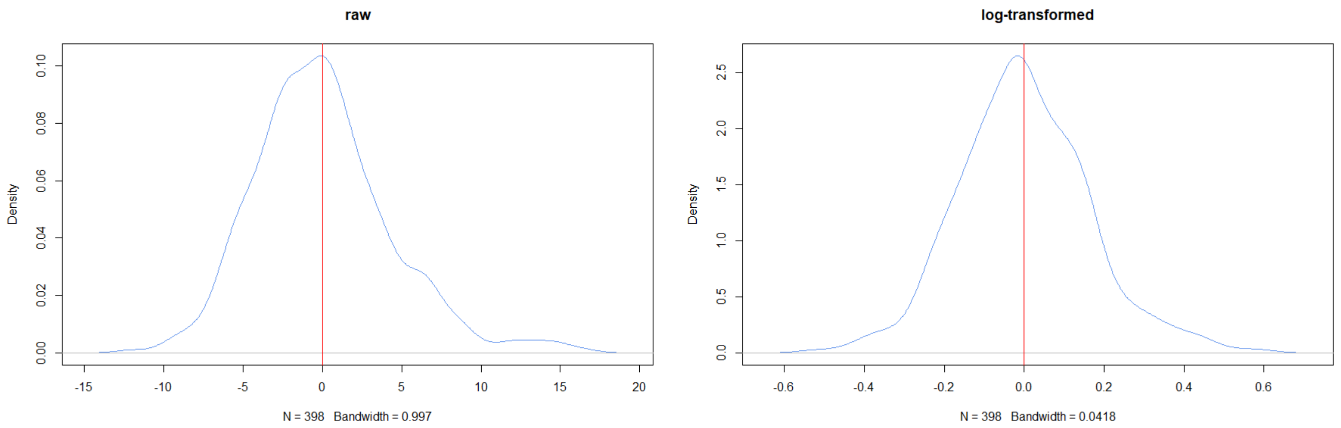
ii. Create a regression (call it `regr_wt_log`) of `log.mpg.` on `log.weight.` from `cars_log`

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

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

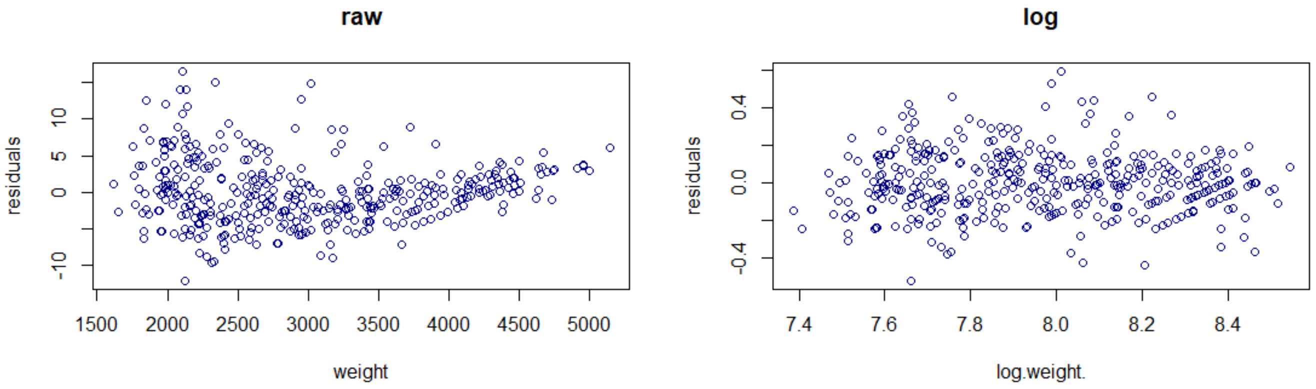
1. density plots of residuals

```
#Raw
plot(density(regr_wt$residuals),col = "cornflowerblue", main = "raw")
abline(v=mean(regr_wt$residuals),col = "red")

#Log-transformed
plot(density(regr_wt_log$residuals), col = "cornflowerblue", main = "log-transf
ormed")
abline(v=mean(regr_wt_log$residuals), col = "red")
```



2. scatterplot of `log.weight.` vs. residuals

```
#Raw
plot(cars$weight , regr_wt$residuals , col = "navy", main="raw", xlab="weight",
 ylab="residuals")

#Log-transformed
plot(density(regr_wt_log$residuals), col = "cornflowerblue", main = "log-transf
ormed")
abline(v=mean(regr_wt_log$residuals), col = "red")
```



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

The log regression seems better because it is more normal distirubed.

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

1% increase in weight leads to 1.0583% decrease in mpg.

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

i. Create a bootstrapped confidence interval

```
boot_regr <- function (log, dataset) {
     boot_sample <- sample(1:nrow(dataset), replace = TRUE)
     bootstrap_data <- dataset[boot_sample,]
     regression_bootstrap <- lm(log, data= bootstrap_data)
     regression_bootstrap$coefficients
 }
coef<-replicate(800,boot_regr(log(mpg)~log(weight),cars))
quantile(coef['log(weight)',],c(0.025,0.975))
      2.5%     97.5%
-1.114435 -1.010675
```

ii. Verify your results with a confidence interval using traditional statistics

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

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

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

```
> r2_log_wt=summary(regr_log)$r.squared
> vif_log_wt=1/(1-r2_log_wt)
> vif_log_wt
[1] 9.251547
```

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(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
```

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

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

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

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

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

iv. Report the final regression model and its summary statistics

```
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. Using stepwise VIF selection, have we lost any variables that were previously significant?

If so, did we hurt our explanation by dropping those variables?

We drop the horsepower variable.

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?

Nearly 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?

For 5 or higher, R^2 should be above 80%, and for 10 or higher, R^2 should be above 90%.

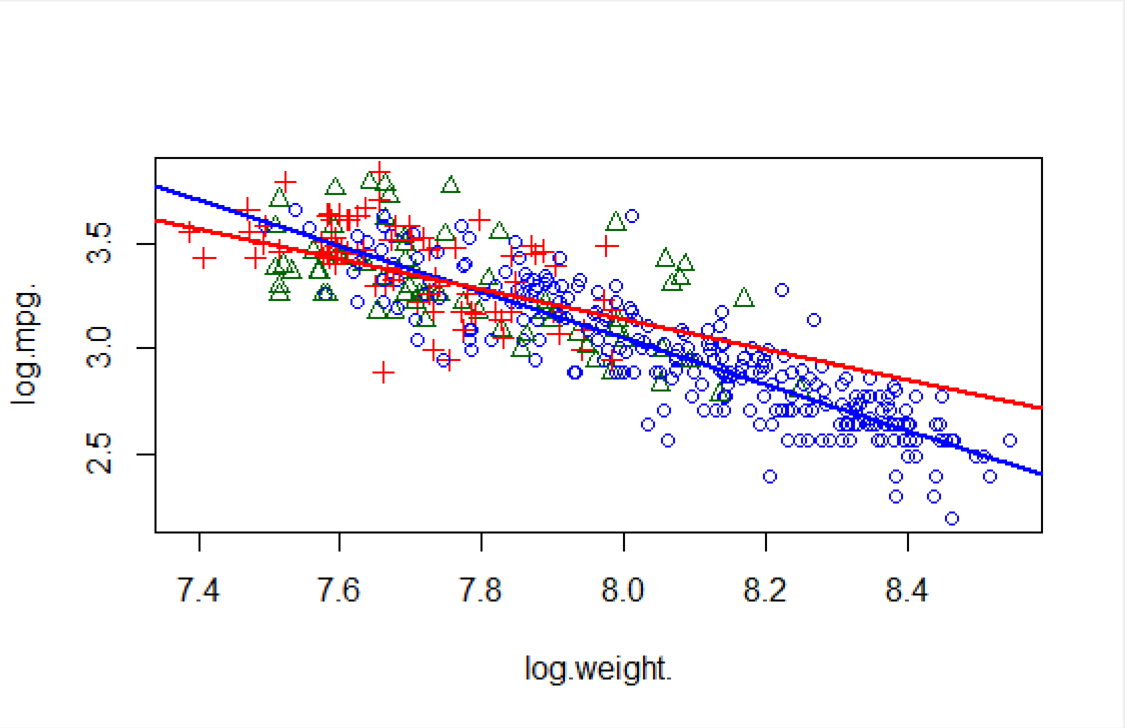
Question 3) Might the relationship of weight on mpg be different for cars from different origins?
Let’s try visualizing this. First, plot all the weights, using different colors and symbols for the three origins:

```
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:

```
#Extract cars from US, EU and Japan
cars_us <- subset(cars_log, origin==1)
cars_eu <- subset(cars_log, origin==2)
cars_jp <- subset(cars_log, origin==3)
```

```
regr_us <- lm(log.mpg. ~ log.weight., data=cars_us)
abline(regr_us, col=origin_colors[1], lwd=2)
regr_eu <- lm(log.mpg. ~ log.weight., data=cars_eu)
abline(regr_eu, col=origin_colors[2], lwd=2)
regr_jp <- lm(log.mpg. ~ log.weight., data=cars_jp)
abline(regr_eu, col=origin_colors[3], lwd=2)
```



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

Though it is no very clear, but there are differences between different origins.