HW11

109006206

Set Working Directories & Reading Files

```
setwd("/Users/olivia/Documents/Documents/Study/Semester 6/BACS/HW11")
library(ggplot2)
library(ggpubr)
library(car)
cars<-read.table("auto-data.txt", header=FALSE, na.strings = "?")
names(cars) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")
```

QUESTION 1

```
cars_log <- with(cars, data.frame(log(mpg), log(cylinders), log(displacement), log(horsepower), log(wei
```

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

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

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

summary(cars_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
```

Answer : The log-transformed factors have a significant effect on log mpg at 10% significance are horsepower, weight, acceleration, model_year, origin

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

Answer : Yes , horsepower and acceleration now have effects on mpg

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

Answer : log.cylinders , log.displacement

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

1) Create a regression (call it regr_wt) of mpg over weight from the original cars 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
```

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

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

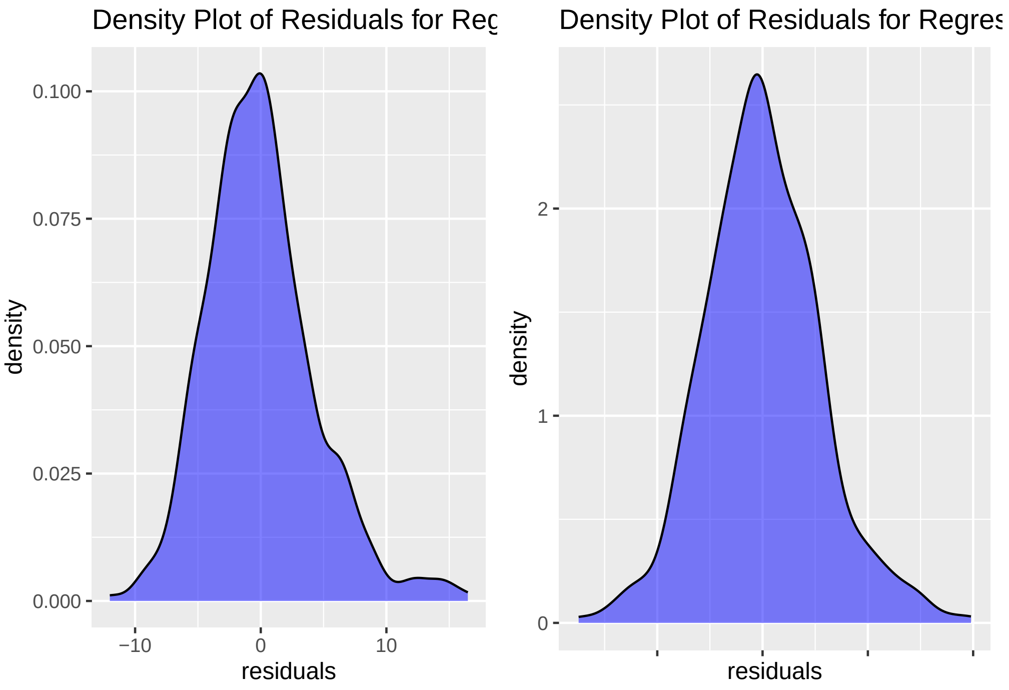
```
##
## Call:
## lm(formula = log.mpg. ~ log.weight., data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52408 -0.10441 -0.00805  0.10165  0.59384
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.5219     0.2349   49.06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

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

1. Density plots of residuals.

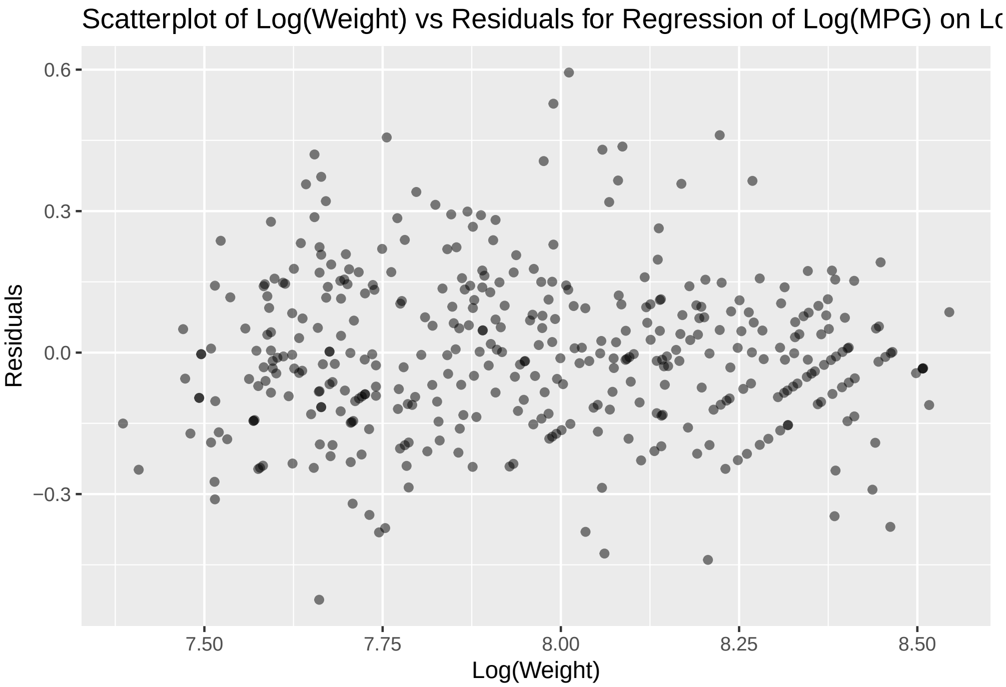
```
par(mfrow = c(1, 2))
A<- ggplot(data.frame(residuals = residuals(regr_wt)), aes(x = residuals)) +
  geom_density(fill = "blue", alpha = 0.5) +
  labs(title = "Density Plot of Residuals for Regression of MPG on Weight (Raw)")
B<- ggplot(data.frame(residuals = residuals(regr_wt_log)), aes(x = residuals)) +
  geom_density(fill = "blue", alpha = 0.5) +
  labs(title = "Density Plot of Residuals for Regression of MPG on Weight (Log-Transformed)")

ggarrange(A,B + rremove("x.text"),
          ncol = 2, nrow = 1)
```



2. Scatterplot of log.weight. vs. residuals

```
ggplot(cars_log, aes(x = log.weight., y = residuals(regr_wt_log))) +
  geom_point(alpha = 0.5) +
  labs(x = "Log(Weight)", y = "Residuals", title = "Scatterplot of Log(Weight) vs Residuals for Regressi
```



4) Which regression produces better distributed residuals for the assumptions of regression?
Answer : I think Log-Transformed regression produces better distributed residuals

5) How would you interpret the slope of log.weight. vs log.mpg. in simple words?
Answer : 1% increase in weight leads to 1.0583% decrease in mpg

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

```
confint(regr_wt_log, level = 0.95)
```

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

QUESTION 2

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

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

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

r2 <- summary(weight_regr)$r.squared
vif <- 1 / (1 - r2)
vif
```

```
## [1] 17.57512
```

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()

1) Use vif(regr__log) to compute VIF of the all the independent variables

```
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
```

2) 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.acceleration. + model_yea
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
```

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

```
regr_log <- lm(log.mpg. ~ log.cylinders. + log.weight. + log.acceleration. + model_year + factor(origin
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
```

```
regr_log <- lm(log.mpg. ~ log.horsepower. + log.weight. + log.acceleration. + model_year + factor(origi
vif(regr_log)
```

```
##                        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
```

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

4) Report the final regression model and its summary statistic

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

```
##
## 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?

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

```
## [1] 0.8855764
```

```
summary(final)$r.squared
```

```
## [1] 0.8855764
```

Answer : Yes, It is possible that we have lost some variables that were previously significant in the model. Dropping significant variables may hurt the model's explanatory power, as they may have been important predictors of the dependent variable.

Since the new model has a slight lower R-squared, that means we slightly hurt our explanation by dropping those variables.

D) From only the formula for VIF, try deducing/deriving the following:

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

Answer : If an independent variable has no correlation with other independent variables, it means that its coefficient of determination(R^2) will be zero

Therefore, the denominator in the VIF formula will be equal to 1, and the VIF score will be 1.

2) 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?

Answer :

1) To get VIF scores of 5 or higher :
   Correlation between X1 and X2 would need to be at least 0.8944
   R = sqrt(1 / (1/5 - 1)) = 0.8944

2) To get a VIF score of 10 or higher:
   Correlation between X1 and X2 would need to be at least 0.9487
   R = sqrt(1 / (1/10 - 1)) = 0.9487

QUESTION 3

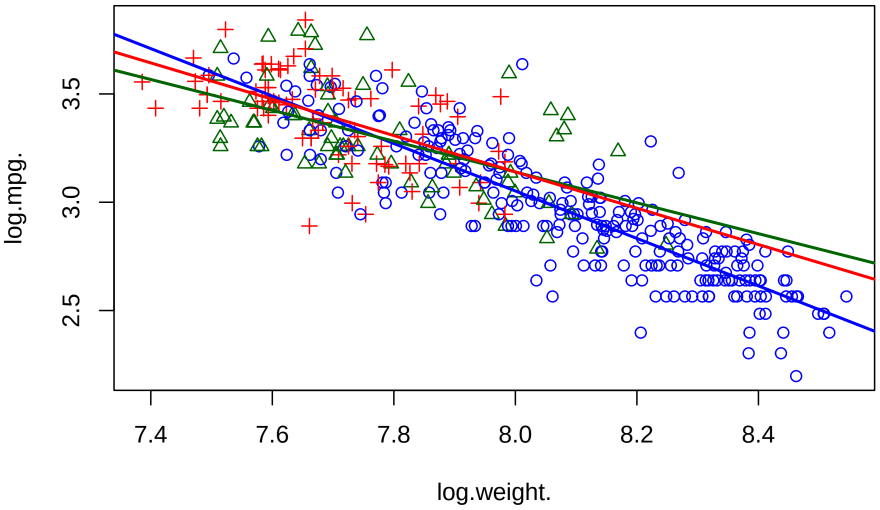
A) Let's add three separate regression lines on the scatterplot, one for each of the origins. Here's one for the US to get you started:

```
origin_colors = c("blue", "darkgreen", "red")
with(cars_log, plot(log.weight., log.mpg., pch=origin, 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_eur=subset(cars_log, origin==2)
wt_regr_eur=lm(log.mpg. ~ log.weight., data=cars_eur)
abline(wt_regr_eur, col=origin_colors[2], lwd=2)

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=2)
```



B) Do cars from different origins appear to have different weight vs. mpg relationships?
Answer : Yes, I think different origins have different relationships between weight and mpg.