HW12

106022103

2021/5/15

Assist

- `106000199`
  - Remind me using GVIF to do the VIF operation.
  - How to fix the plot BUG in Q3.a

Set up

import libary

```
library(ggplot2)
require(qqplotr)
library(plyr)
library(gridExtra)
library(ggcorrplot)
library(magrittr)
library(ggpubr)
library(car)
```

Read file

```
cars <- read.table("data/auto-data.txt", header=FALSE, na.strings = "?")
names(cars) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")
cars <- cars[complete.cases(cars), ] # remove missing value
cars[,'origin']<-factor(cars[,'origin']) # convert to factor
cars[,'car_name']<-factor(cars[,'car_name']) # convert to factor
cars_value <- cars[,-9] # drop the class data
cars_log <- with(cars_value, data.frame(log(mpg), log(cylinders), log(displacement), log(horsepower), lo

corr <- round(cor(cars_value[,-8]), 2)
p.mat <- cor_pmat(cars_value[,-8])

log.corr <- round(cor(cars_log[,-8]), 2)
log.p.mat <- cor_pmat(cars_log[,-8])
```

Q1.

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

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

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

```
##
## Call:
## lm(formula = log.mpg. ~ ., 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 ***
## origin2             0.050717   0.020920   2.424  0.01580 *
## origin3             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
## Multiple R-squared:  0.8919, Adjusted R-squared:  0.8897
## F-statistic:   395 on 8 and 383 DF,  p-value: < 2.2e-16
```

ANSWER: `log.horsepower.`, `log.weight.`, `log.acceleration.`, `model_year` and `origin` have a significant effect on log.mpg. at 10% significance.

```r
summary(lm(mpg~., data = cars_value))
```

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

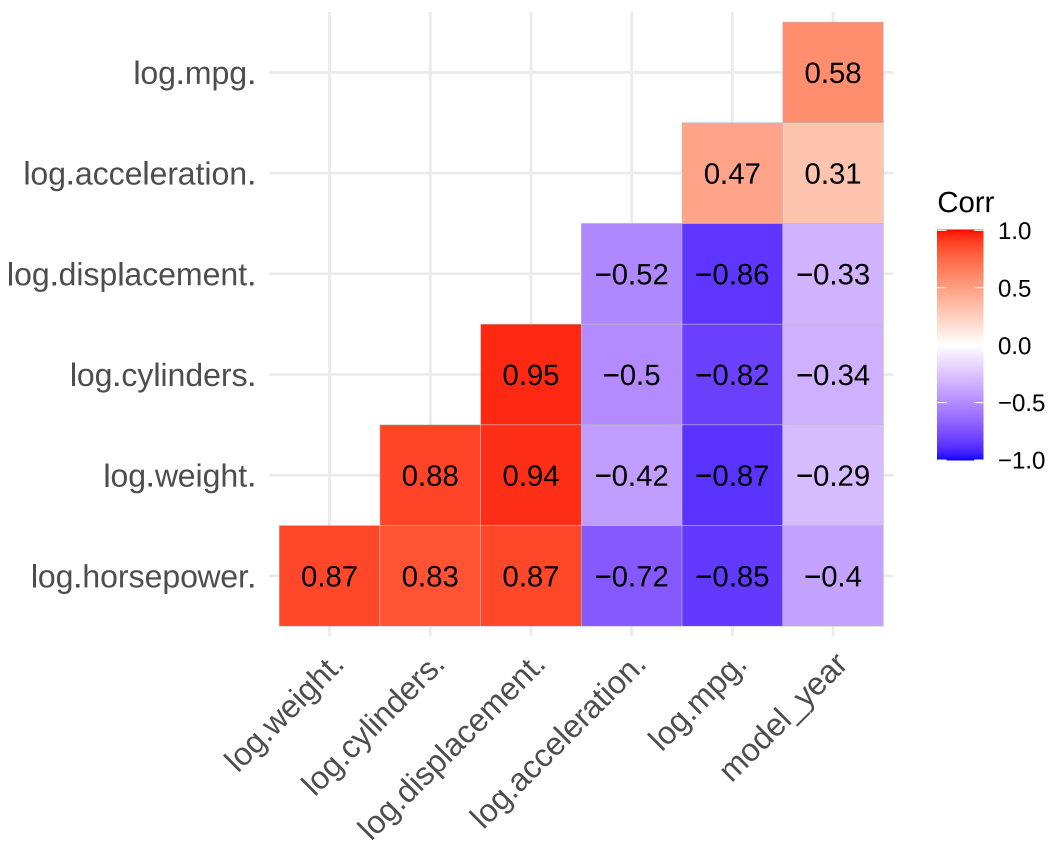
```
##
## Call:
## lm(formula = mpg ~ ., data = cars_value)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0095 -2.0785 -0.0982  1.9856 13.3608
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.795e+01  4.677e+00  -3.839 0.000145 ***
## cylinders    -4.897e-01  3.212e-01  -1.524 0.128215
## displacement  2.398e-02  7.653e-03   3.133 0.001863 **
## horsepower   -1.818e-02  1.371e-02  -1.326 0.185488
## weight       -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## acceleration  7.910e-02  9.822e-02   0.805 0.421101
```

```
## model_year    7.770e-01  5.178e-02  15.005  < 2e-16 ***
## origin2       2.630e+00  5.664e-01   4.643 4.72e-06 ***
## origin3       2.853e+00  5.527e-01   5.162 3.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.307 on 383 degrees of freedom
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

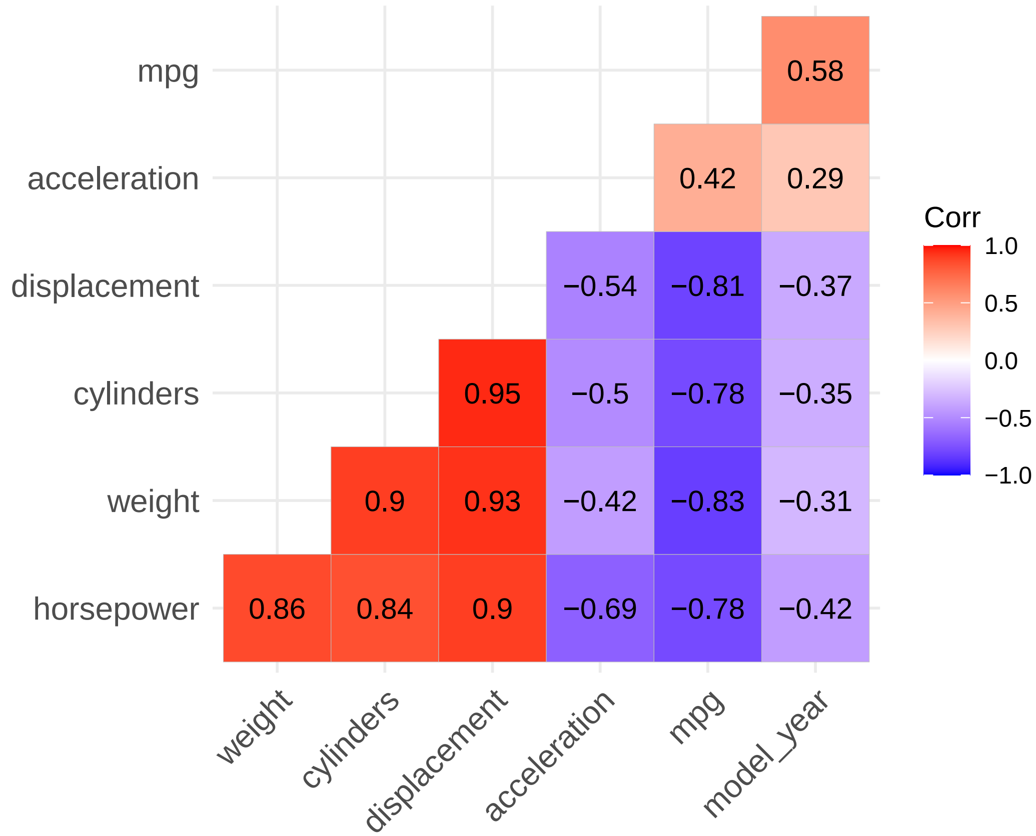
ANSWER: Compared two results, `horsepower` and `acceleration` will have effects after we take the log operation. The reason may be because `lm()` can only respond to linear relationships. Other non-linear relationships do not obtain a very high level of significance.

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

```
ggcorrplot(log.corr, hc.order = TRUE,
  type = "lower", p.mat = log.p.mat, lab = TRUE)
```




```
ggcorrplot(corr, hc.order = TRUE,
  type = "lower", p.mat = p.mat, lab = TRUE)
```



ANSWER: The `acceleration` still has insignificant effects on `mpg`, and it probably because this factor has not so much relation with mpg. The `displacement`,`cylinders`, `weight`, `horsepower` has still opposite effects on `mpg`, and either linear or logarithmic may have opposite relationships.

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

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

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

```
##
## Call:
## lm(formula = mpg ~ weight, data = cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9736  -2.7556  -0.3358   2.1379  16.5194
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.216524   0.798673   57.87   <2e-16 ***
## weight      -0.007647   0.000258  -29.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 4.333 on 390 degrees of freedom
## Multiple R-squared:  0.6926, Adjusted R-squared:  0.6918
## F-statistic: 878.8 on 1 and 390 DF,  p-value: < 2.2e-16
```

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

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

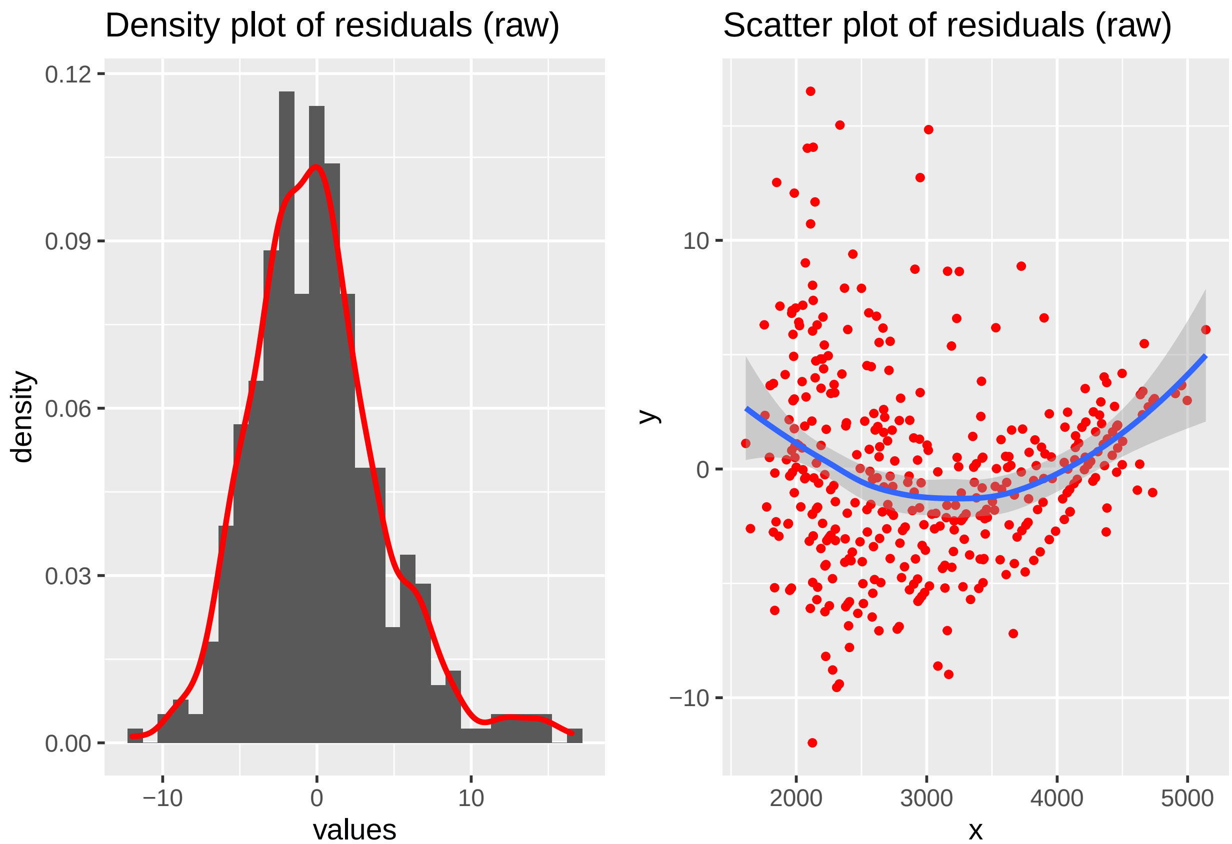
```
##
## Call:
## lm(formula = log.mpg. ~ log.weight., data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52321 -0.10446 -0.00772  0.10124  0.59445
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.5152     0.2365   48.69   <2e-16 ***
## log.weight.  -1.0575     0.0297  -35.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1651 on 390 degrees of freedom
## Multiple R-squared:  0.7648, Adjusted R-squared:  0.7642
## F-statistic:  1268 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
density_hist_plot <- function(values,title=""){
  p <- ggplot(mapping = aes(values)) +
    geom_histogram(mapping = aes(y = stat(density))) +
    geom_density(color = "red", size = 1) +
    labs(title = paste("Density plot of",title))
  p
}

scatter_plot <- function(x, y, title = ""){
  p <- ggplot(mapping = aes(x=x, y=y)) +
    geom_point(color = "red", size = 1) +
    geom_smooth() +
    labs(title = paste("Scatter plot of",title))
  p

}


# combine two plots
density_qq_plot <- function(values){
  text <- substitute(values)
  p1 <- norm_qq_ggplot(values)
  p2 <- density_hist_plot(values)
  figure <- ggarrange(p1,p2)
```

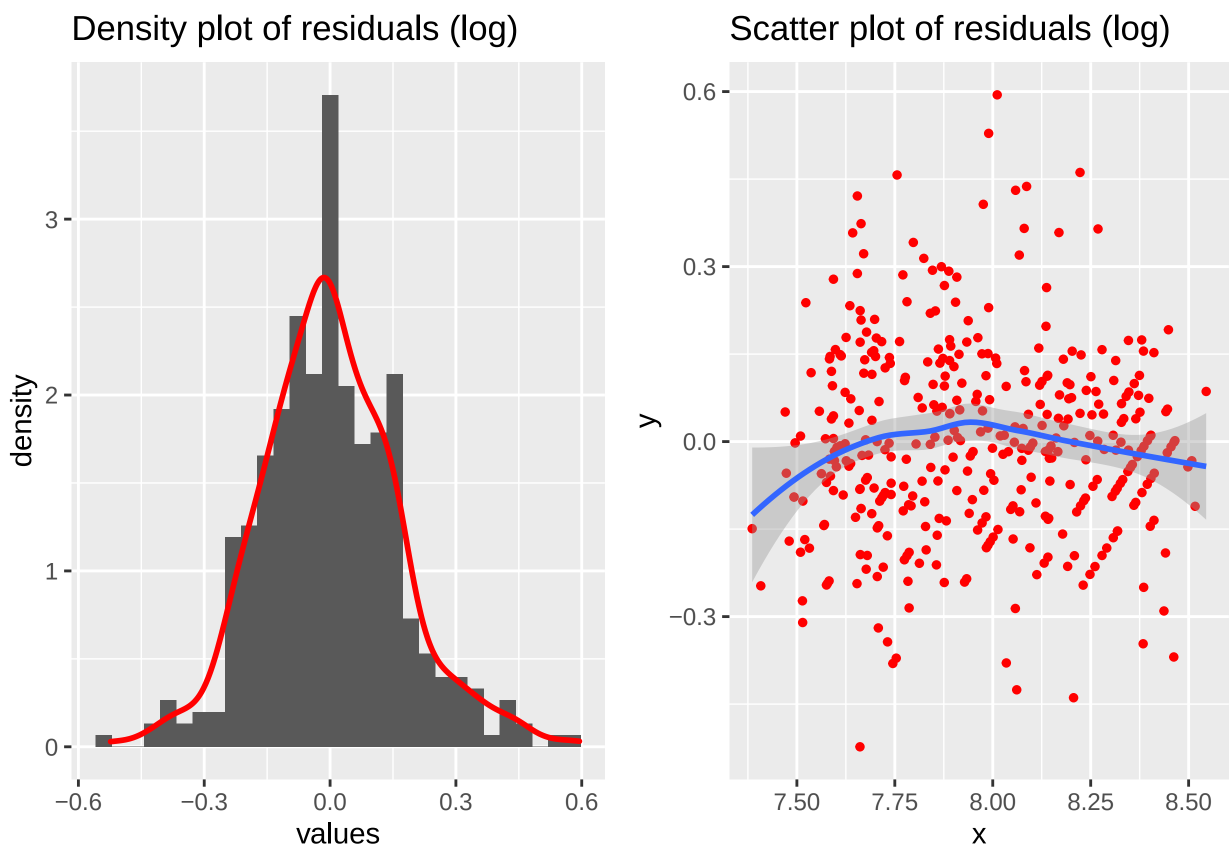
```r
  annotate_figure(figure,top = text_grob(text, color = "red", face = "bold", size = 14))
  # grid.arrange(p1,p2, nrow=1,ncol=2)
}
```

```r
p1 <- density_hist_plot(regr_wt$residuals,"residuals (raw)")
p2 <- scatter_plot(cars$weight, resid(regr_wt),"residuals (raw)")
ggarrange(p1,p2)
```

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




```r
p3 <- density_hist_plot(regr_wt_log$residuals,"residuals (log)")
p4 <- scatter_plot(cars_log$log.weight., resid(regr_wt_log),"residuals (log)")
ggarrange(p3,p4)
```



iv. which regression produces better residuals for the assumptions of regression? ANSWER: From the results above, the log regression seems has better regression.

```
regr_wt_log$coefficients[2]
```

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

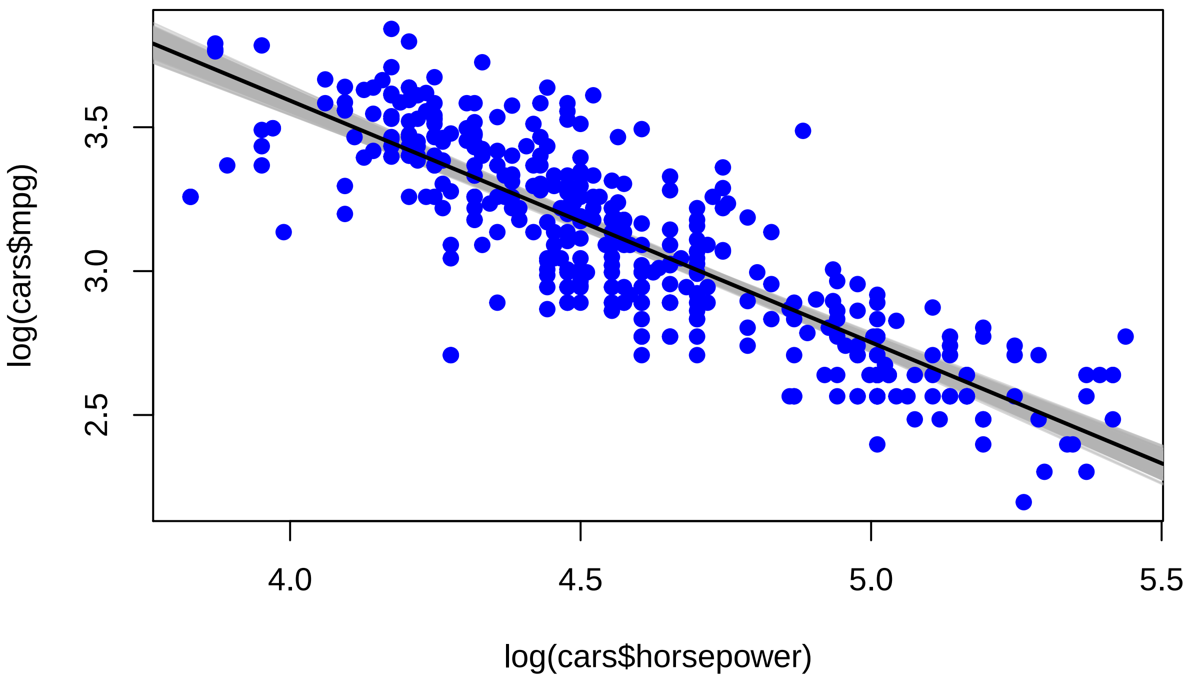
```
## log.weight.
##   -1.057506
```

ANSWER: It means each 1% change in `log.weight.` leads to -1.05% change in `log.mpg.`.

c.

```
# Empty plot canvas
plot(log(cars$horsepower), log(cars$mpg), col=NA, pch=19)
# 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)
abline(regr_boot, lwd=1, col=rgb(0.7, 0.7, 0.7, 0.5))
regr_boot$coefficients
}
# Bootstrapping for confidence interval
```

```
coeffs <- replicate(300, boot_regr(log(mpg) ~ log(horsepower), cars))
# Plot points and regression line
points(log(cars$horsepower), log(cars$mpg), col="blue", pch=19)
abline(a=mean(coeffs["(Intercept)",]),
b=mean(coeffs["log(horsepower)",]), lwd=2)
```



i.

```
# Confidence interval values
quantile(coeffs["log(horsepower)",], c(0.025, 0.975))
```

```
##       2.5%      97.5%
## -0.8960636 -0.7866247
```

```
hp_regr_log <- lm(log(mpg) ~ log(horsepower), cars)
confint(hp_regr_log)
```

ii.

```
##                      2.5 %     97.5 %
## (Intercept)      6.7217993  7.1994991
## log(horsepower) -0.8937626 -0.7899313
```

ANSWER: The two results are same.

Q2 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)
```

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

```
r2_weight <- summary(regr_wt_log)$r.squared
vif_weight <- 1 / (1 - r2_weight)
sqrt(vif_weight)
```

```
## [1] 2.061832
```

b. Stepwise VIF Selection

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

```
##                        GVIF Df GVIF^(1/(2*Df))
## log.cylinders.    5.427610  1        2.329723
## log.weight.       4.871730  1        2.207200
## log.acceleration. 1.401202  1        1.183724
## model_year        1.206351  1        1.098340
## factor(origin)    1.821167  2        1.161682
```

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

```
##                        GVIF Df GVIF^(1/(2*Df))
```

```
## log.weight.       1.933208  1        1.390398
## log.acceleration. 1.304761  1        1.142261
## model_year        1.175545  1        1.084225
## factor(origin)    1.710178  2        1.143564
```

ANSWER: The `log.displacement.`, `log.horsepower.`, `log.cylinders.` are removed in order.

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

```
regr_log_vif <- lm(log.mpg. ~ log.weight. + log.acceleration. + model_year + factor(origin),  data=cars_
e1 <- summary(regr_log_vif)$r.squared
regr_log=lm(log.mpg.~.,data=cars_log)
e2 <- summary(regr_log)$r.squared
sprintf("There are %.4f explanation loss in VIF ", (e2-e1))
```

```
## [1] "There are 0.0074 explanation loss in VIF "
```

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? **ANSWER:** The VIF of any independent variable should be 1.

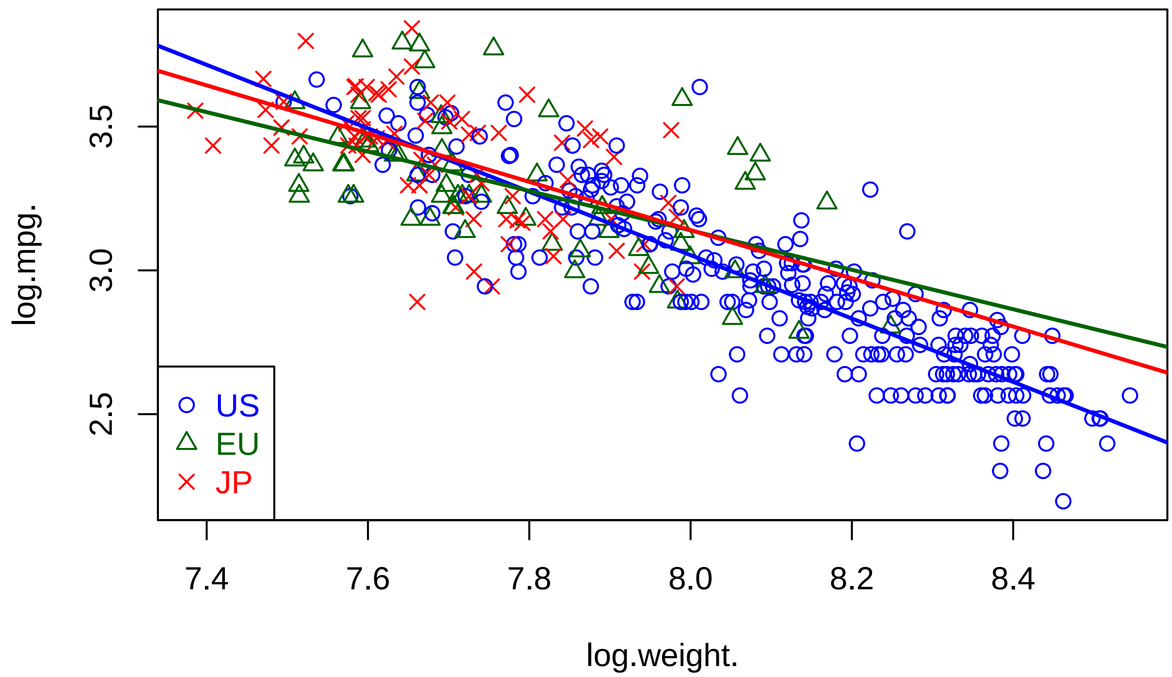
ii. Given a regression with only two independent variables (X1 and X2), how correlated would X1 and X2 have to be, to get VIF scores of 5 or higher? To get VIF scores of 10 or higher? ANSWER:

- Since $VIF_j = \frac{1}{1-R_j^2}$
  - $VIF_j = 5 \rightarrow R_j^2 = 0.8$
  - $VIF_j = 10 \rightarrow R_j^2 = 0.9$

Q3 Might the relationship of weight on mpg be different for cars from different origins?

a.

```
origin_pch = c(1,2,4)
origin_colors = c("blue", "darkgreen", "red")
with(cars_log, plot(log.weight., log.mpg., pch=origin_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_eu <- subset(cars_log, origin==2)
wt_regr_eu <- lm(log.mpg. ~ log.weight., data=cars_eu)
abline(wt_regr_eu, 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)

legend("bottomleft", legend = c("US", "EU", "JP"),
 pch = origin_pch,
 col = origin_colors, text.col = origin_colors)
```



b.(not graded)Do cars from different origins appear to have different weight vs. mpg relationships?

ANSWER: Different regions have different relationships with cars, and the U.S. is more different to Japan and Europe.

Reference Link

- ggplot2 scatter plots
- Concatenate Strings in R
- Variance inflation factor
- R visulize
- Visualization of a correlation matrix using ggplot2