bacs\_hw9

2024-04-21

110071010

**110034002** has done research and discussed with me how the Stepwise VIF Selection can be stimulated as some part of topic was left for the next lecture.

# Question 1

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

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))

## 1a

Run a new regression on the cars\_log dataset, with mpg.log. dependent on all other variables

### i

**Question**

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

**Answer**

Acceleration, weight, model year, factor(origin), and horsepower

log\_regr <- summary(lm(formula = log.mpg. ~ log.cylinders. + log.displacement. +   
 log.horsepower. + log.weight. + log.acceleration. + model\_year +   
 factor(origin), data = cars\_log, na.action = na.exclude))  
log\_regr

##   
## Call:  
## lm(formula = log.mpg. ~ log.cylinders. + log.displacement. +   
## log.horsepower. + log.weight. + log.acceleration. + model\_year +   
## factor(origin), data = cars\_log, na.action = na.exclude)  
##   
## 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  
## Multiple R-squared: 0.8919, Adjusted R-squared: 0.8897   
## F-statistic: 395 on 8 and 383 DF, p-value: < 2.2e-16

### ii

**Question**

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

**Answer**

Acceleration and horsepower are new factors. This may arise because the original data could be skewed, and now log-transformed to display the otherwise hidden patterns.

### iii

**Question**

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

**Answer**

Cylinders and displacement. This may arise due to the data’s multicollinearity.

## 1b

### i

Create a regression (call it regr\_wt) of mpg over weight from the original cars dataset

regr\_wt <- summary(lm(mpg ~ weight, data=cars))  
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 (call it regr\_wt\_log) of log.mpg. on log.weight. from cars\_log

regr\_wt\_log <- summary(lm(log.mpg. ~ log.weight., data=cars\_log, na.action=na.exclude))  
regr\_wt\_log

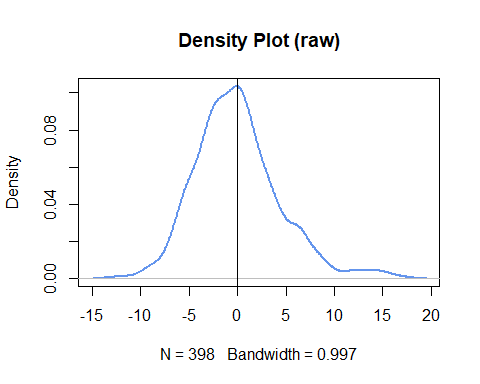
##   
## Call:  
## lm(formula = log.mpg. ~ log.weight., data = cars\_log, na.action = na.exclude)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.52408 -0.10441 -0.00805 0.10165 0.59384   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11.5219 0.2349 49.06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

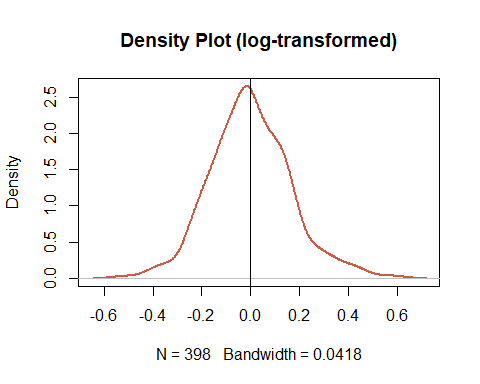
Visualize the residuals of both regression models (raw and log-transformed)

#### (1) density plot of residuals

plot(density(regr\_wt$residuals), lwd = 2, col = "cornflowerblue", main = "Density Plot (raw)")  
abline(v= mean(regr\_wt$residuals))

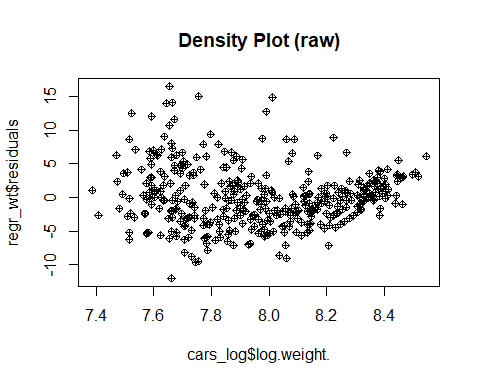


plot(density(regr\_wt\_log$residuals), lwd = 2, col = "coral3", main = "Density Plot (log-transformed)")  
abline(v= mean(regr\_wt\_log$residuals))

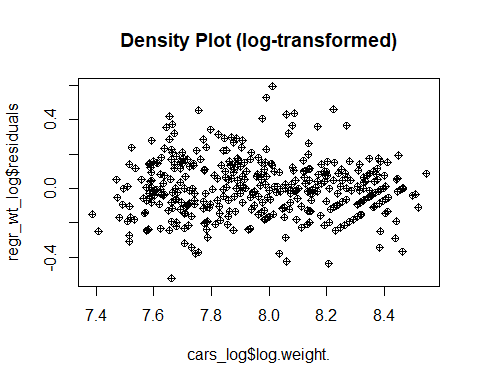


#### (2) scatterplot of log.weight. vs. residuals

plot(cars\_log$log.weight., regr\_wt$residuals, pch=10,main = "Density Plot (raw)")



plot(cars\_log$log.weight., regr\_wt\_log$residuals, pch=10, main = "Density Plot (log-transformed)")



### iv

**Question**

Which regression produces better distributed residuals for the assumptions of regression?

**Answer**

The log-transformed regression.

### v

**Question**

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

**Answer**

Based on the summary tables in (i) and (ii), it is clear that one percent change in log.weight. leads to -1.0583 percent change in log.mpg.

### vi

**Question**

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

**Answer**

regr\_wt\_log

##   
## Call:  
## lm(formula = log.mpg. ~ log.weight., data = cars\_log, na.action = na.exclude)  
##   
## 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

slope\_estimate <- regr\_wt\_log$coefficients["log.weight.", "Estimate"]  
slope\_se <- regr\_wt\_log$coefficients["log.weight.", "Std. Error"]  
  
CI\_lower <- slope\_estimate - 1.96 \* slope\_se  
CI\_upper <- slope\_estimate + 1.96 \* slope\_se  
  
cat("CI\_upperbound:", CI\_upper, "\n")  
cat("CI\_lowerbound:", CI\_lower)

## CI\_upperbound: -1.000448   
## CI\_lowerbound: -1.116088

# 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)

## 2a

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

logweight\_regr <- lm(log.weight.~log.cylinders.+log.displacement.+log.horsepower.+log.acceleration.+model\_year, data=cars\_log, na.action = na.exclude)  
r2\_logweight\_regr <- summary(logweight\_regr)$r.squared  
vif\_logweight <- 1 / (1 - r2\_logweight\_regr)  
vif\_logweight

## [1] 16.07917

## 2b

Let's try a procedure called Stepwise VIF Selection to remove highly collinear predictors.

### i

Use vif(regr\_log) to compute VIF of the all the independent variables

#install.packages('car')  
library('car')

## 載入需要的套件：carData

regr\_log <- lm(log.weight. ~ log.cylinders.+log.displacement.+log.horsepower.+log.acceleration.+model\_year,  
 data=cars\_log, na.action=na.exclude)  
vif(regr\_log)

## log.cylinders. log.displacement. log.horsepower. log.acceleration.   
## 9.748860 13.412802 7.013535 2.253283   
## model\_year   
## 1.198164

### ii

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

# log.displacement scores the largest VIF  
regr\_log <- lm(log.weight. ~ log.cylinders.+log.horsepower.+log.acceleration.+model\_year,  
 data=cars\_log, na.action=na.exclude)  
vif(regr\_log)

## log.cylinders. log.horsepower. log.acceleration. model\_year   
## 3.326803 5.208472 2.167932 1.190458

### iii

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

# Keep on to eliminate log.horsepower, whose VIF is at once the largest and greater than 5  
regr\_log <- lm(log.weight. ~ log.cylinders.+log.acceleration.+model\_year,  
 data=cars\_log, na.action=na.exclude)  
vif(regr\_log)

## log.cylinders. log.acceleration. model\_year   
## 1.412557 1.382461 1.165129

### iv

Report the final regression model and its summary statistics

summary(regr\_log)

##   
## Call:  
## lm(formula = log.weight. ~ log.cylinders. + log.acceleration. +   
## model\_year, data = cars\_log, na.action = na.exclude)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.35101 -0.09406 -0.00256 0.09311 0.41564   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.398532 0.198541 32.228 <2e-16 \*\*\*  
## log.cylinders. 0.835451 0.026327 31.734 <2e-16 \*\*\*  
## log.acceleration. 0.035708 0.043451 0.822 0.412   
## model\_year 0.001084 0.001950 0.556 0.579   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1331 on 394 degrees of freedom  
## Multiple R-squared: 0.7769, Adjusted R-squared: 0.7752   
## F-statistic: 457.3 on 3 and 394 DF, p-value: < 2.2e-16

## 2c

**Question**

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?

**Answer**

We eliminated displacement and horsepower, which used to seem significant. In dropping these variables, the explanation of the fit model can be hurt due to the R-squared change.

## 2d

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

### i

**Question**

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

**Answer**

By the VIF formula 1 / (1 - R-squared), no correlation means R-squared to be 0, and VIF in turn becomes 1.

### ii

**Question**

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**

Correlation would have to be above 0.894 to get VIF scores of 5 or higher.

Correlation would have to be above 0.948 to get VIF scores of 10 or higher.

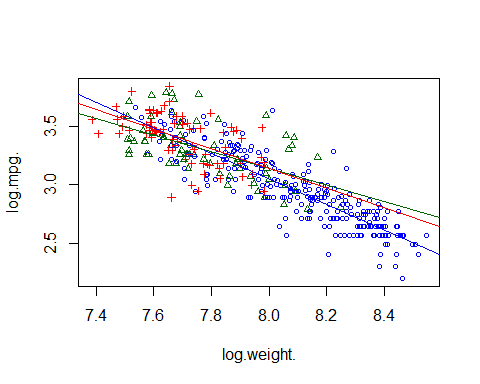
# Question 3

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

## 3a

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], cex = 0.7))  
  
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=1)  
  
cars\_jp <- subset(cars\_log, origin == 2)  
wt\_regr\_jp <- lm(log.mpg. ~ log.weight., data=cars\_jp)  
abline(wt\_regr\_jp, col=origin\_colors[2], lwd=1)  
  
cars\_eu <- subset(cars\_log, origin == 3)  
wt\_regr\_eu <- lm(log.mpg. ~ log.weight., data=cars\_eu)  
abline(wt\_regr\_eu, col=origin\_colors[3], lwd=1)



## 3b

**Question**

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

**Answer**

Yes, each of their data points seems clustered.