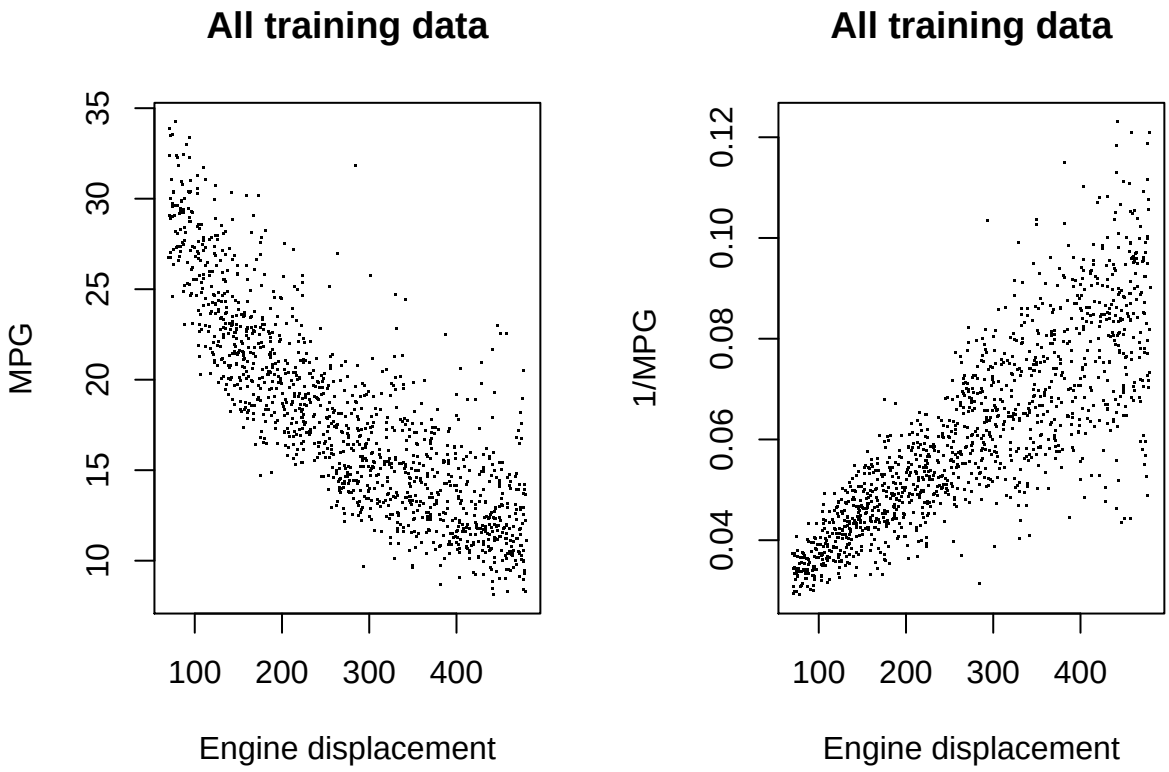
---
title: "36-402 Homework 6"
author: "Jacky Liu"
date: "2/12/21"
output: pdf_document

```
load("engine.Rdata")
```

Problem 1

Problem 1 (a)

```
par(mfrow=c(1,2))
plot(engine.xtrain, engine.ytrain, xlab="Engine displacement", pch=".", ylab = "MPG", main = "All traini
plot(engine.xtrain, 1/engine.ytrain, xlab="Engine displacement", pch=".", ylab = "1/MPG", main = "All t
```



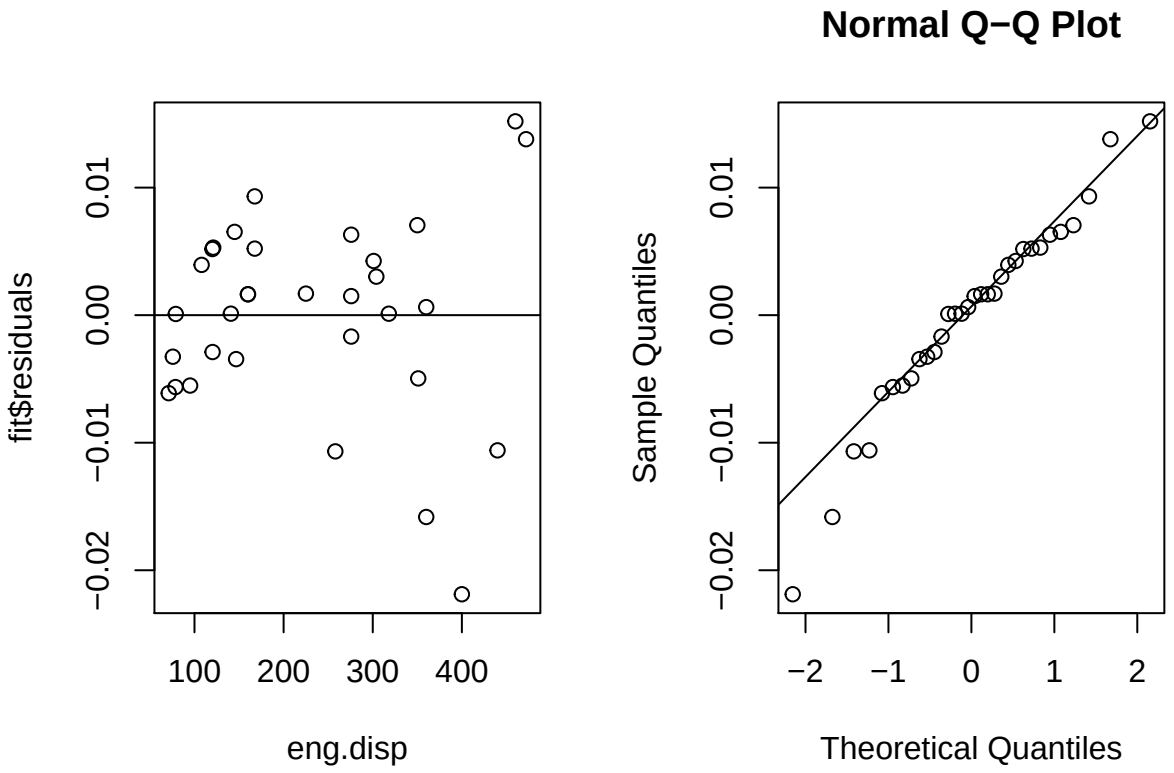
Just by looking, the 1/mpg plot seems slightly more linear than mpg since the mpg plot's points looks like it curves more as we move right. The 1/mpg plot seems to have a violation of constant variance as points seem to spread out more as we move right. A method that could be used to address this problem might be to make our model more complex, such as adding interaction terms, or transforming our data.

Problem 1 (b)

```
eng.disp = engine.xtrain[,1]
mpg = engine.ytrain[,1]
fit = lm((1/mpg) ~ eng.disp)
summary(fit)
```

```
##
## Call:
## lm(formula = (1/mpg) ~ eng.disp)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.021881 -0.003825  0.001061  0.005191  0.015194
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0273265  0.0030013   9.105 3.88e-10 ***
## eng.disp    0.0001166  0.0000115  10.139 3.32e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007936 on 30 degrees of freedom
## Multiple R-squared:  0.7741, Adjusted R-squared:  0.7666
## F-statistic: 102.8 on 1 and 30 DF,  p-value: 3.316e-11
```

```
par(mfrow=c(1,2))
plot(eng.disp, fit$residuals); abline(h=0)
qqnorm(fit$residuals); qqline(fit$residuals)
```



The assumptions of the linear model (iid, linearity, constant variance, Gaussian noise) seem relatively but not perfectly plausible. Although the Q-Q line seems to fit the points very well, there may be signs of heteroskedasticity in the residuals plot. Overall the data show some modest signs of betraying the linear model assumptions, but not too extreme.

Problem 1 (c)

```
set.seed(69)
n = length(engine.xtrain[,1])
B = 10000 ## number of bootstraps
results = numeric(B) ## vector to hold results
for(b in 1:B){
  i = sample(x = 1:n, size = n, replace = TRUE) ## sample indices
  x = engine.xtrain[,i] ## get data
  y = engine.ytrain[,i]
  fit = lm(1/y ~ x)
  slope = fit$coefficients[2]
  results[b] = slope
}
mean(results)
```

```
## [1] 0.0001107592
```

```r
sd(results)
```

```
## [1] 0.0003257189
```

The sample mean of bootstrapped slope parameter estimates is 0.0001107592 and standard deviation is 0.0003257189.

```r
abs(0.0003257189 - 0.0000115)/0.0003257189
```

```
## [1] 0.9646935
```

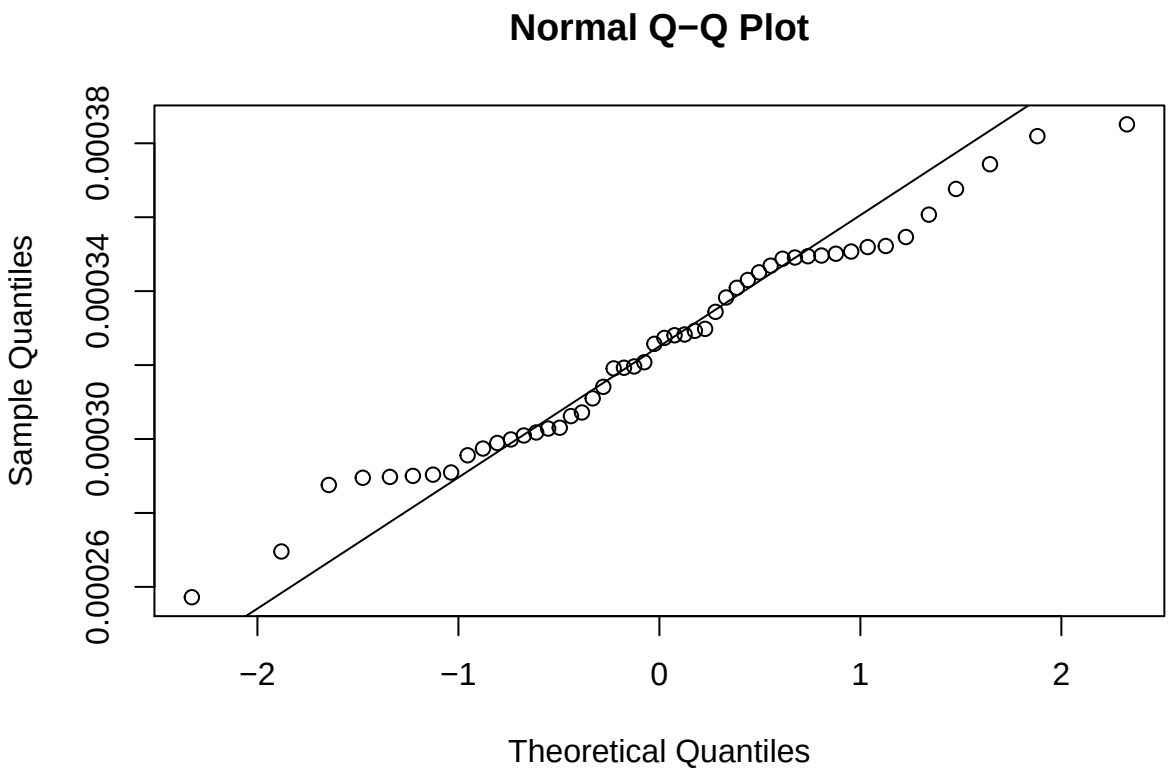
The percent change was 0.9646935.

Problem 1 (d)

```r
# create a 50x200 matrix where each row is a single replication of B=200 bootstrap samples
samples = matrix(results, nrow=50)
deviations = apply(samples, 1, sd)
t.test(deviations, mu=0.0000115)
```

```
##
##  One Sample t-test
##
## data:  deviations
## t = 76.114, df = 49, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 1.15e-05
## 95 percent confidence interval:
##  0.0003162552 0.0003327840
## sample estimates:
##    mean of x
## 0.0003245196
```

```r
qqnorm(deviations); qqline(deviations)
```



The 50 bootstrap errors mostly fit the line except for the two tails, which means this mostly looks like a plausible normal sample. It makes sense to check this because bootstrapped standard deviations (which is inherently biased) should still form a normal distribution.

Problem 1 (e)

One of the 50 sample stdevs calculated is a measure of how different the subset of slopes are from each other.

The difference is that if you do 10 reps of 1000 instead, you will get less data that's more precise, or in other words you get more bias and less variance.

Problem 2

Problem 2 (a)

```
library(MASS)
data(cats)
model = lm(Hwt ~ 0 + Bwt:Sex, data=cats)
summary(model)
```

```
##
## Call:
## lm(formula = Hwt ~ 0 + Bwt:Sex, data = cats)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4841 -0.9929 -0.1036  0.9879  5.2330
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## Bwt:SexF  3.88345    0.08925   43.51   <2e-16 ***
## Bwt:SexM  3.91461    0.05024   77.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.453 on 142 degrees of freedom
## Multiple R-squared:  0.9825, Adjusted R-squared:  0.9822
## F-statistic:  3982 on 2 and 142 DF,  p-value: < 2.2e-16
```

Forcing the intercept to zero is a reasonable thing to do because we don't always assume homosedasticity of estimated residuals, which means that we may not want to always assume an intercept term, and we want to insure that the residual term is zero mean and normally distributed.

Problem 2 (b)

$$H_0 : B_{1Sex=Female} = B_{1Sex=Male}$$
$$H_a : B_{1Sex=Female} \neq B_{1Sex=Male}$$

```
anova(lm(Hwt ~ 0 + Bwt * Sex, data=cats))
```

```
## Analysis of Variance Table
##
## Response: Hwt
##            Df  Sum Sq Mean Sq   F value  Pr(>F)
## Bwt         1 16820.8 16820.8 8091.1766 < 2e-16 ***
## Sex         2     0.7     0.4    0.1719 0.84225
## Bwt:Sex     1     8.3     8.3    4.0077 0.04722 *
## Residuals 140   291.0     2.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that in the anova above, the p-value under the interaction of Bwt by Sex is 0.04722. That means we reject our null hypothesis under an alpha=0.05 significance level and conclude that the slope coefficient for Bwt between female and male cats most likely differ.

Problem 2 (c)

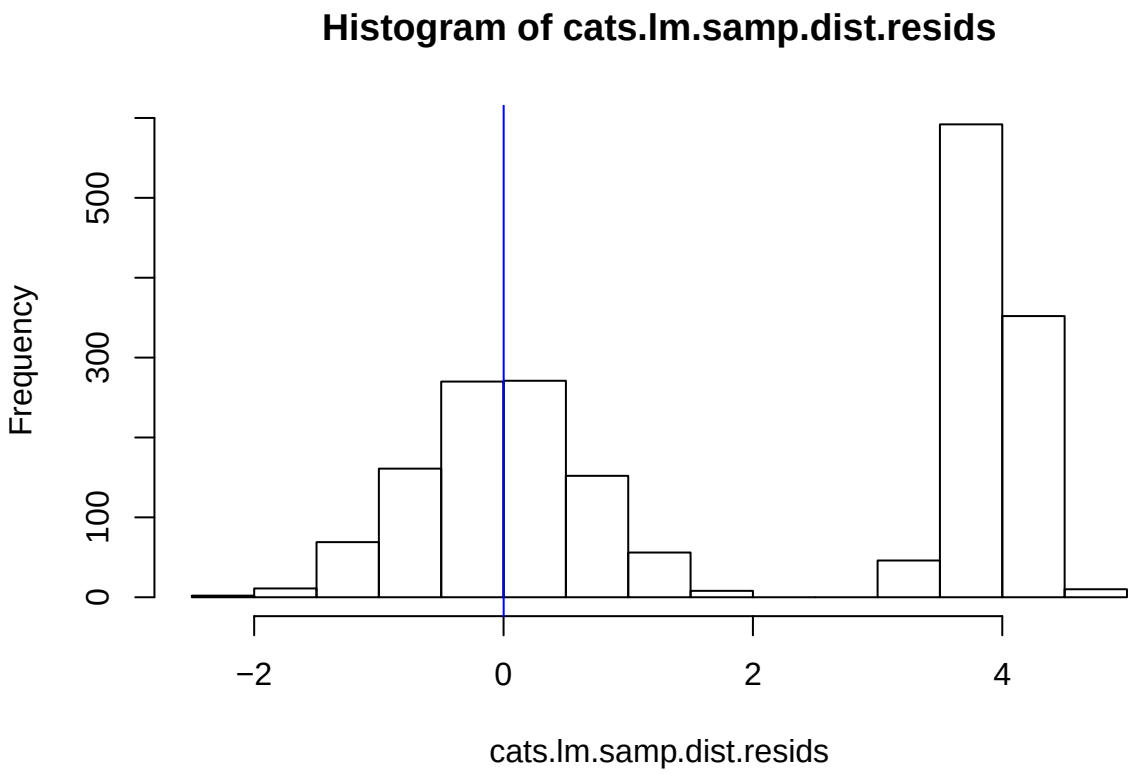
```
t = (3.88345 - 3.91461)^2
t
```

```
## [1] 0.0009709456
```

The value of our test statistic is 0.0009709456.

Problem 2 (d)

```
B = 1000
femalecats = cats[which(cats$Sex=="F"),]
female.lm = lm(Hwt ~ 0 + Bwt, data=femalecats)
malecats = cats[which(cats$Sex=="M"),]
male.lm = lm(Hwt ~ 0 + Bwt, data=malecats)

resample <- function(x) {
  return(sample(x, size=length(x), replace=TRUE))
}

sim.cats.resids <- function() {
  new.femalecats = femalecats
  new.malecats = malecats

  noise.female = resample(residuals(female.lm))
  noise.male = resample(residuals(male.lm))

  new.femalecats$Hwt = fitted(female.lm) + noise.female
  new.malecats$Hwt = fitted(male.lm) + noise.male

  new.cats = rbind(new.femalecats, new.malecats)
  return(new.cats)
}

coefs.cats.lm <- function(df) {
  fit <- lm(Hwt ~ Bwt, data=df)
  return(coefficients(fit))
}

cats.lm.samp.dist.resids <- replicate(B,coefs.cats.lm(sim.cats.resids()))

hist(cats.lm.samp.dist.resids)
abline(v=t, col="blue")
```



Problem 2 (e)

```
count = 0
for(resid in cats.lm.samp.dist.resids){
  if(t >= resid){
    count = count + 1
  }
}
count/length(cats.lm.samp.dist.resids)
```

```
## [1] 0.257
```

Based on the result, we conclude that there is most likely no significant difference between the regression lines for male and female cats.