MA678 Homework 2

9/10/2020

```
library(rstanarm)
```

```
## Loading required package: Rcpp
## This is rstanarm version 2.21.1
## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.
## - For execution on a local, multicore CPU with excess RAM we recommend calling
##   options(mc.cores = parallel::detectCores())
```

```
library(ggplot2)
library(loo)
```

```
## This is loo version 2.3.1
## - Online documentation and vignettes at mc-stan.org/loo
## - As of v2.0.0 loo defaults to 1 core but we recommend using as many as possible. Use the 'cores' arg
```

10.7 Predictive simulation for linear regression:

Take one of the models from the previous exercise.

10.7a

Instructor A is a 50-year-old woman who is a native English speaker and has a beauty score of -1. Instructor B is a 60-year-old man who is a native English speaker and has a beauty score of -0.5. Simulate 1000 random draws of the course evaluation rating of these two instructors. In your simulation, use posterior_predict to account for the uncertainty in the regression parameters as well as predictive uncertainty.

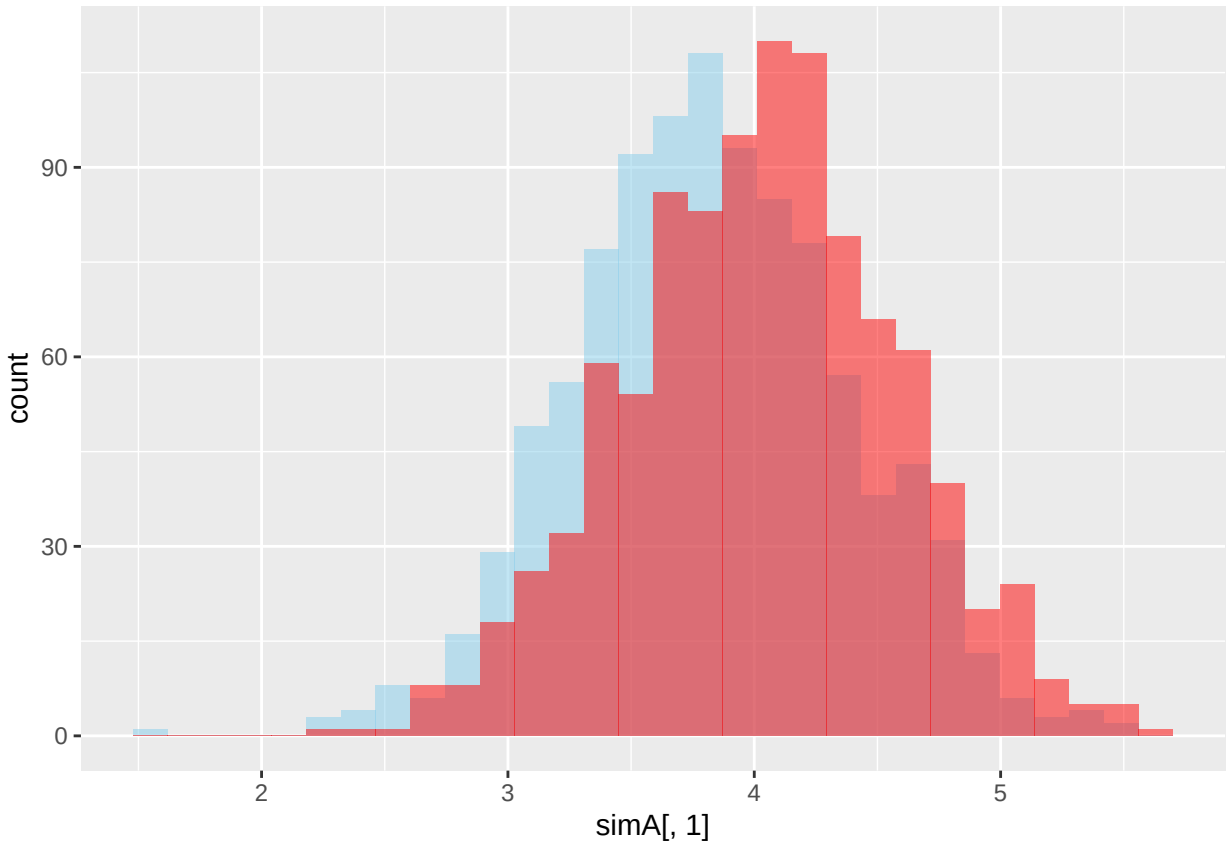
```
beauty <- read.csv("/Users/amelia/Documents/mssp/MA678/hw1/beauty.csv", header=T)
M10.6b <- stan_glm(eval ~ beauty + female + beauty:female,data=beauty,refresh=0)

instA <- data.frame(beauty=-1,female=1,age=50,minority=0,nonenglish=0)
instB <- data.frame(beauty=-0.5,female=0,age=60,minority=0,nonenglish=0)
simA <- posterior_predict(M10.6b,newdata=instA,draws=1000)
simB <- posterior_predict(M10.6b,newdata=instB,draws=1000)
```

10.7b

Make a histogram of the difference between the course evaluations for A and B. What is the probability that A will have a higher evaluation?

```
ggplot() + geom_histogram(aes(simA[,1]),fill="skyblue",alpha=0.5) +geom_histogram(aes(simB[,1]),fill="r
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
probA <- c(apply(simA,2,mean),apply(simA,2,sd))
probB <- c(apply(simB,2,mean),apply(simB,2,sd))
probC <- c(probA[1]-probB[1],sqrt(probA[2]^2+probB[2]^2))
1 - pnorm(0,probC[1],probC[2])
```

```
## [1] 0.3957532
```

10.8 How many simulation draws:

Take the model from Exercise 10.6 that predicts course evaluationsfrom beauty and other predictors.

10.8a

Display and discuss the fitted model. Focus on the estimate and standard error for the coefficient of beauty.

```
print(M10.6b)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      eval ~ beauty + female + beauty:female
##  observations: 463
##  predictors:   4
## ------
##                Median MAD_SD
## (Intercept)     4.1    0.0
## beauty          0.2    0.0
## female         -0.2    0.1
## beauty:female -0.1    0.1
```

```
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.5    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

The slope coefficient of 0.2 means that professors with 1 more point in beauty score seem to have evaluations 0.2 points higher. Standard error = 0.

10.8b

Compute the median and mad sd of the posterior simulations of the coefficient of beauty, and check that these are the same as the output from printing the fit.

```
sims <-as.matrix(M10.6b)
MEDIAN <- apply(sims,2,median)
MAD_SD <- apply(sims,2,mad)
print(cbind(round(MEDIAN,1),round(MAD_SD,1)))
```

```
##               [,1] [,2]
## (Intercept)    4.1  0.0
## beauty         0.2  0.0
## female        -0.2  0.1
## beauty:female -0.1  0.1
## sigma          0.5  0.0
```

10.8c

Fit again, this time setting iter = 1000 in your stan_glm call. Do this a few times in order to get a sense of the simulation variability.

```
df <- matrix(nrow = 4, ncol = 5)
for(i in 1:5){
M10.8c <- stan_glm(eval ~ beauty+ female + beauty:female, data = beauty, refresh = 0, iter = 1000)
df[,i] <- coef(M10.8c)
}
iteration <- data.frame(t(df))
colnames(iteration) <- c("intercept", "beauty", "female", "beauty:female")
iteration
```

```
##   intercept    beauty     female beauty:female
## 1  4.103958 0.1993333 -0.2051452    -0.1100176
## 2  4.103596 0.1985383 -0.2035496    -0.1120224
## 3  4.103777 0.2024329 -0.2061718    -0.1144307
## 4  4.104548 0.2011127 -0.2069857    -0.1132240
## 5  4.104106 0.1991295 -0.2053316    -0.1161149
```

10.8d

Repeat the previous step, setting iter = 100 and then iter = 10.

```
for(i in 1:5){
M10.8c <- stan_glm(eval ~ beauty+ female + beauty:female, data = beauty, refresh = 0, iter = 100)
df[,i] <- coef(M10.8c)
}
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```r
iteration <- data.frame(t(df))
colnames(iteration) <- c("intercept", "beauty", "female", "beauty:female")
iteration
```

```
##   intercept    beauty     female beauty:female
## 1  4.099812 0.2027244 -0.1997417    -0.1159675
## 2  4.108548 0.1920504 -0.2116853    -0.0999203
## 3  4.106298 0.2070463 -0.2009302    -0.1174576
## 4  4.100042 0.2061065 -0.1984242    -0.1170468
## 5  4.105387 0.2013142 -0.2030030    -0.1167627
```

```r
for(i in 1:5){
M10.8c <- stan_glm(eval ~ beauty+ female + beauty:female, data = beauty, refresh = 0, iter = 10)
df[,i] <- coef(M10.8c)
}
```

```
## Warning: There were 1 divergent transitions after warmup. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: There were 4 chains where the estimated Bayesian Fraction of Missing Information was low. Se
## http://mc-stan.org/misc/warnings.html#bfmi-low

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: The largest R-hat is 6.08, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat

## Warning: Markov chains did not converge! Do not analyze results!

## Warning: There were 7 divergent transitions after warmup. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: There were 2 chains where the estimated Bayesian Fraction of Missing Information was low. Se
## http://mc-stan.org/misc/warnings.html#bfmi-low

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: The largest R-hat is 6.05, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat

## Warning: Markov chains did not converge! Do not analyze results!

## Warning: There were 6 divergent transitions after warmup. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: There were 2 chains where the estimated Bayesian Fraction of Missing Information was low. Se
## http://mc-stan.org/misc/warnings.html#bfmi-low

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: The largest R-hat is 10.88, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat

## Warning: Markov chains did not converge! Do not analyze results!

## Warning: There were 15 divergent transitions after warmup. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: There were 1 chains where the estimated Bayesian Fraction of Missing Information was low. Se
## http://mc-stan.org/misc/warnings.html#bfmi-low

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: The largest R-hat is 15.28, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat

## Warning: Markov chains did not converge! Do not analyze results!

## Warning: There were 10 divergent transitions after warmup. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: There were 2 chains where the estimated Bayesian Fraction of Missing Information was low. Se
## http://mc-stan.org/misc/warnings.html#bfmi-low

## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: The largest R-hat is 12.78, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat
```

```
## Warning: Markov chains did not converge! Do not analyze results!
```

```
iteration <- data.frame(t(df))
colnames(iteration) <- c("intercept", "beauty", "female", "beauty:female")
iteration
```

```
##     intercept     beauty      female beauty:female
## 1  1.50683813  1.9600508 -0.30506021   -1.00735455
## 2  0.92199291  0.9256488  2.50135278   -1.27965766
## 3  2.00899520  1.5502403 -1.94787448    0.06491031
## 4 -0.19172271  1.1989016  2.82787046   -1.81544272
## 5  0.02096185 -0.7460674  0.01637064   -0.75119616
```

10.8e

How many simulations were needed to give a good approximation to the mean and standard error for the coefficient of beauty?
I probably would want to use 1000 or more.

11.5

Residuals and predictions: The folder Pyth contains outcome y and predictors x1, x2 for 40 data points, with a further 20 points with the predictors but no observed outcome. Save the file to your working directory, then read it into R using read.table().

```
pyth <- read.table("/Users/amelia/Documents/mssp/MA678/hw2/pyth.txt", header = T)
```

(a)

Use R to fit a linear regression model predicting y from x1, x2, using the first 40 data points in the file. Summarize the inferences and check the fit of your model.

```
data1 <- head(pyth, 40)
fit1 <- lm(y~x1+x2, data = data1)
print(fit1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = data1)
##
## Coefficients:
## (Intercept)           x1           x2
##      1.3151       0.5148       0.8069
```

```
summary(fit1)["r.squared"]
```

```
## $r.squared
## [1] 0.9724241
```

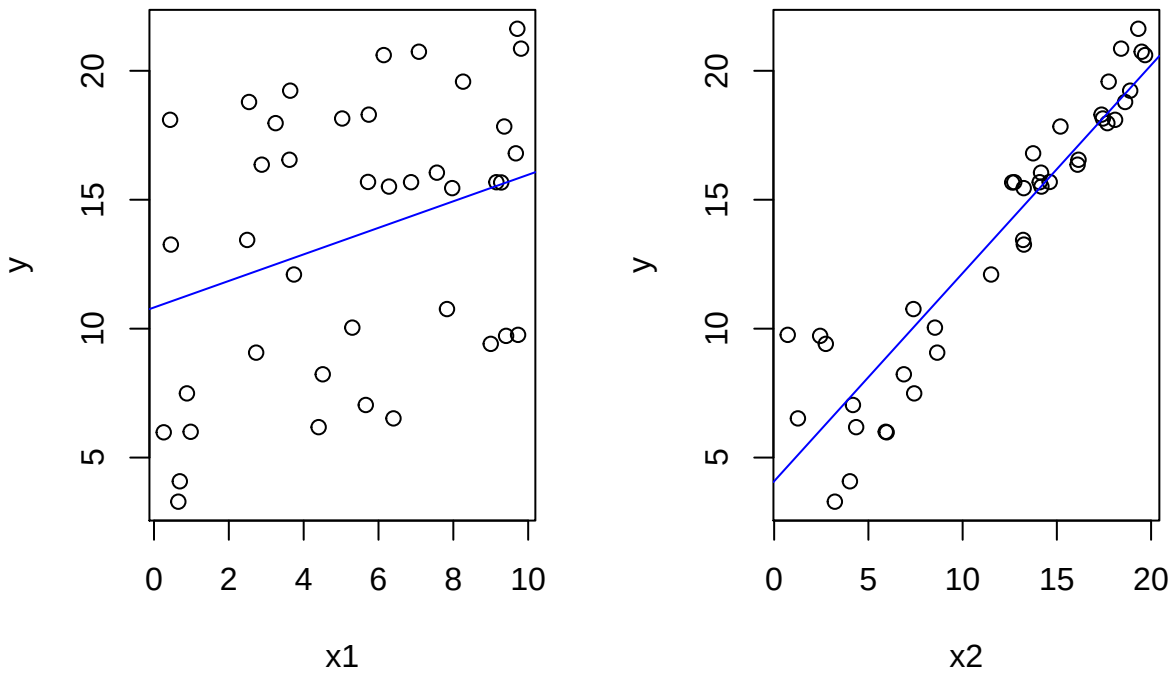
y = 1.31 + 0.51x1 + 0.81x2
The r^2 is close to 1, so the fit is good.

(b)

Display the estimated model graphically as in Figure 10.2

I display y versus x_1 with the average of x_2 and y versus x_2 with the mean of x1.

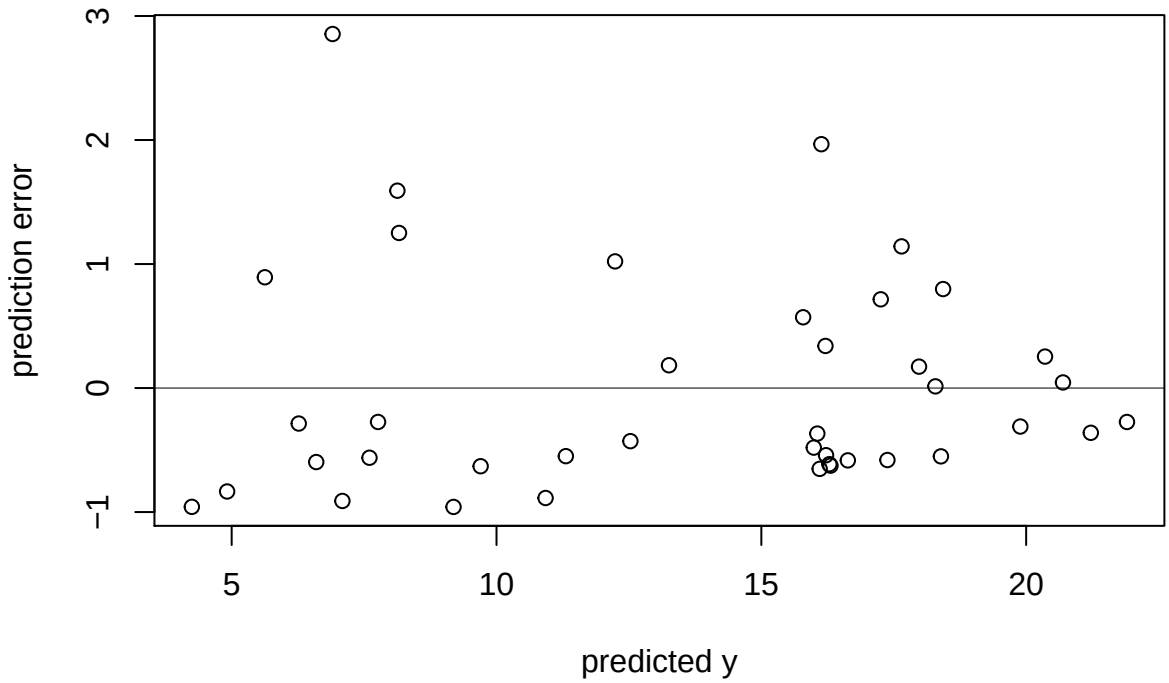
```
par(mfrow=c(1,2))
beta <- coef(fit1)
x1_bar <- mean(data1$x1)
x2_bar <- mean(data1$x2)
figure1 <- plot(data1$x1, data1$y, xlab = "x1", ylab = "y")
abline(beta[1] + beta[3]*x2_bar, beta[2], col = "blue")
figure2 <- plot(data1$x2, data1$y, xlab = "x2", ylab = "y")
abline(beta[1] + beta[2]*x1_bar, beta[3], col = "blue")
```



(c)

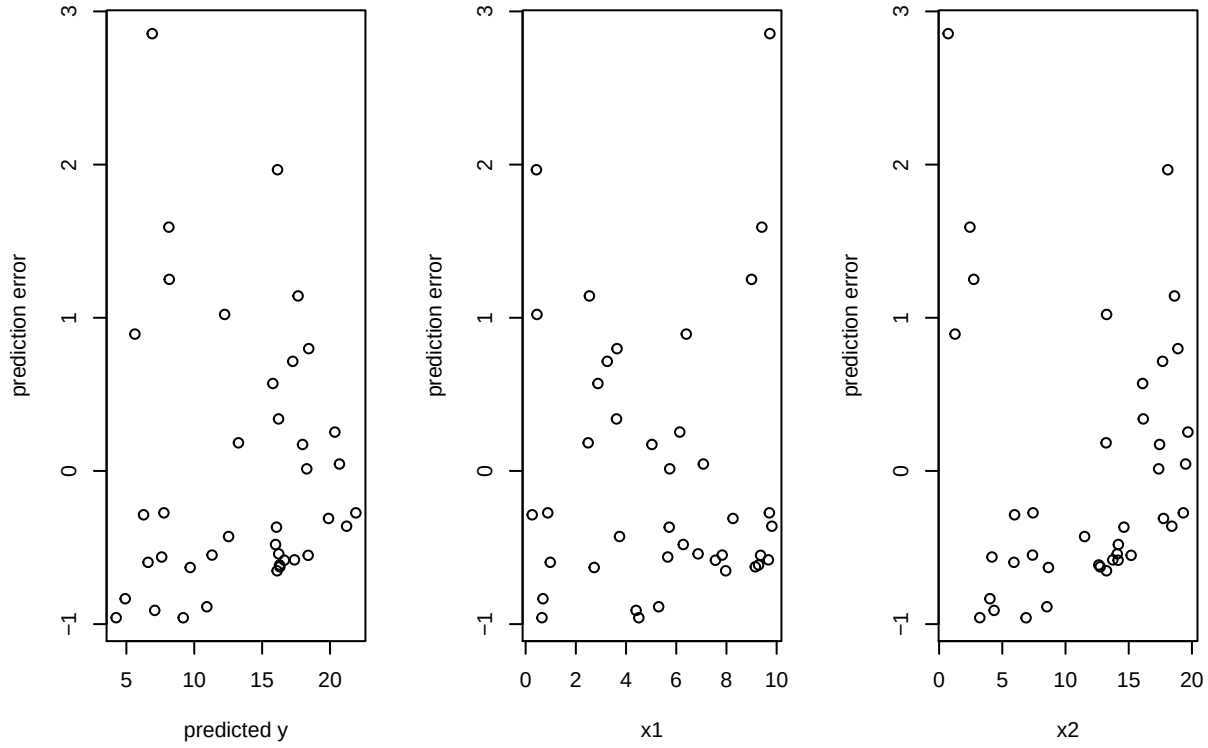
Make a residual plot for this model. Do the assumptions appear to be met?

```
par(mfrow=c(1,1))
predicted <- predict(fit1)
residual <- data1$y - predicted
sigma <- sigma(fit1)
df <- data.frame(data1, predicted, residual)
plot(df$predicted, df$residual, xlab = "predicted y", ylab = "prediction error")
abline(c(0, 0), c(0, sigma), lwd=.5)
```



The residuals are centered around zero for all fitted values. But the distribution is not normal.

```
par(mfrow=c(1,3))
plot(df$predicted, df$residual, xlab = "predicted y", ylab = "prediction error")
plot(df$x1, df$residual, xlab = "x1", ylab = "prediction error")
plot(df$x2, df$residual, xlab = "x2", ylab = "prediction error")
```



(d)

Make predictions for the remaining 20 data points in the file. How confident do you feel about these predictions?

```
data2 <- pyth[41:60,]
predict (fit1, data2, interval="prediction", level=0.95)
```

```
##          fit       lwr       upr
## 41 14.812484 12.916966 16.708002
## 42 19.142865 17.241520 21.044211
## 43  5.916816  3.958626  7.875005
## 44 10.530475  8.636141 12.424809
## 45 19.012485 17.118597 20.906373
## 46 13.398863 11.551815 15.245911
## 47  4.829144  2.918323  6.739965
## 48  9.145767  7.228364 11.063170
## 49  5.892489  3.979060  7.805918
## 50 12.338639 10.426349 14.250929
## 51 18.908561 17.021818 20.795303
## 52 16.064649 14.212209 17.917088
## 53  8.963122  7.084081 10.842163
## 54 14.972786 13.094194 16.851379
## 55  5.859744  3.959679  7.759808
## 56  7.374900  5.480921  9.268879
## 57  4.535267  2.616996  6.453539
## 58 15.133280 13.282467 16.984094
## 59  9.100899  7.223395 10.978403
## 60 16.084900 14.196990 17.972810
```

The residual plot shows that the linearity assumption hasn't been met. So I'm not very confident of the predictions.

12.5

Logarithmic transformation and regression: Consider the following regression: log(weight)=-3.8+2.1log(height)+error, with errors that have standard deviation 0.25. Weights are in pounds and heights are in inches.

(a)

Fill in the blanks: Approximately 68% of the people will have weights within a factor of _____ and _____ of their predicted values from the regression.

```
exp(0.25)
```

```
## [1] 1.284025
```

```
-exp(0.25)
```

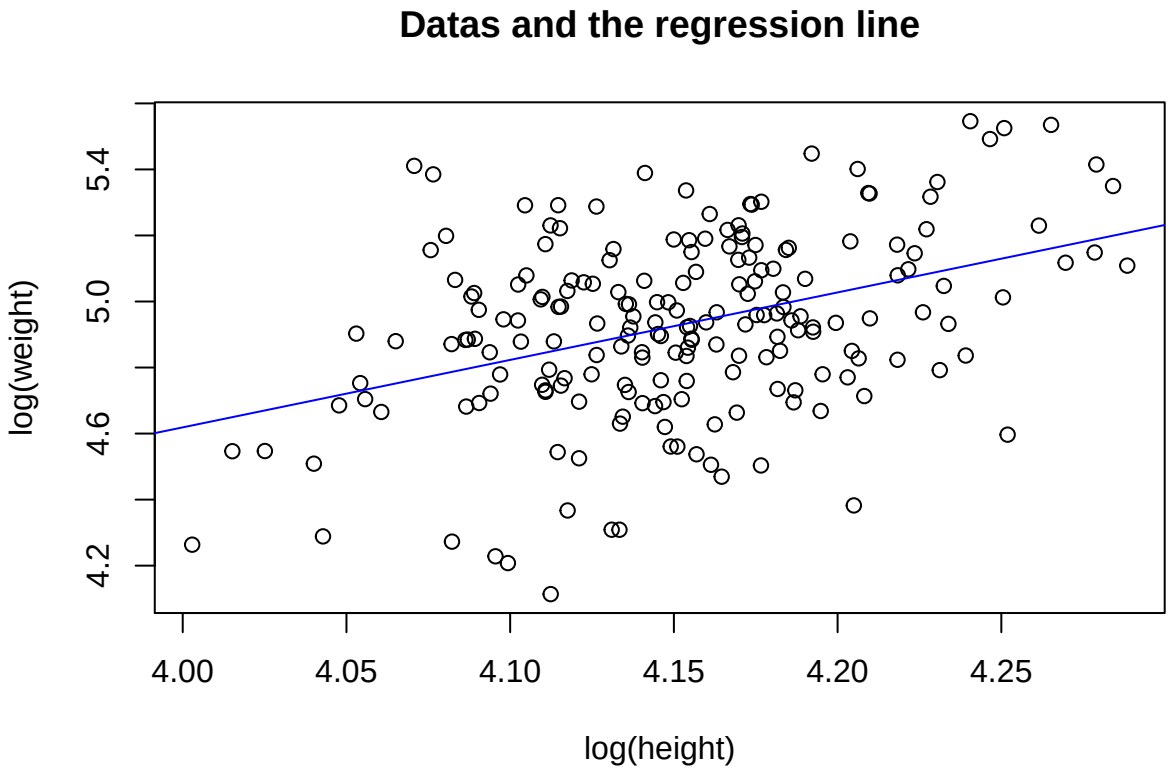
```
## [1] -1.284025
```

The answer is -1.28 and 1.28.

(b)

Using pen and paper, sketch the regression line and scatterplot of log(weight) versus log(height) that make sense and are consistent with the fitted model. Be sure to label the axes of your graph.

```r
set.seed(1)
female_height <- data.frame(rnorm(100, 61.4, 3))
male_height <- data.frame(rnorm(100, 65.7, 3.1))
colnames(female_height) <- "height"
colnames(male_height) <- "height"
total_height <- rbind(female_height, male_height)
weight <- exp(-3.8 + 2.1 * log(total_height$height) + rnorm(200, 0, 0.25))
df <- data.frame(weight, total_height)
df$logweight <- log(df$weight)
df$logheight <- log(df$height)
fit <- lm(logweight~logheight, data = df)
plot(df$logheight, df$logweight, xlab = "log(height)", ylab = "log(weight)", main = "Datas and the regr
abline(fit, col = "blue")
```



12.6

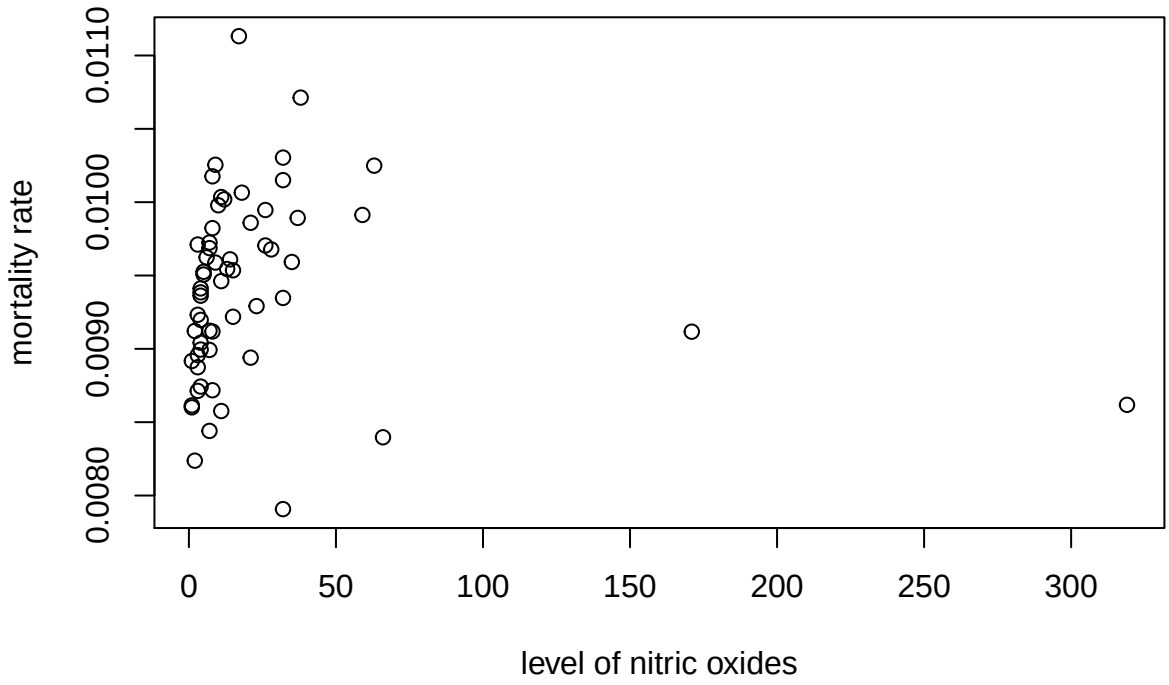
Logarithmic transformations: The folder Pollution contains mortality rates and various environmental factors from 60 US metropolitan areas. For this exercise we shall model mortality rate given nitric oxides, sulfur dioxide, and hydrocarbons as inputs. this model is an extreme oversimplication, as it combines all sources of mortality and does not adjust for crucial factors such as age and smoking. We use it to illustrate log transformation in regression.

```r
pollution <- read.csv("/Users/amelia/Documents/mssp/MA678/hw2/pollution.csv", header = T)
# scale `mort` (which is defined as "total age-adjusted mortality rate per 100,000")
pollution$mort <- pollution$mort / 100000
```

(a)

create a scatterplot of mortality rate versus level of nitric oxides. Do you think linear regression will fit these data well? Fit the regression and evaluate a residual plot from the regression.

```r
plot(pollution$nox, pollution$mort, xlab = "level of nitric oxides", ylab = "mortality rate")
```



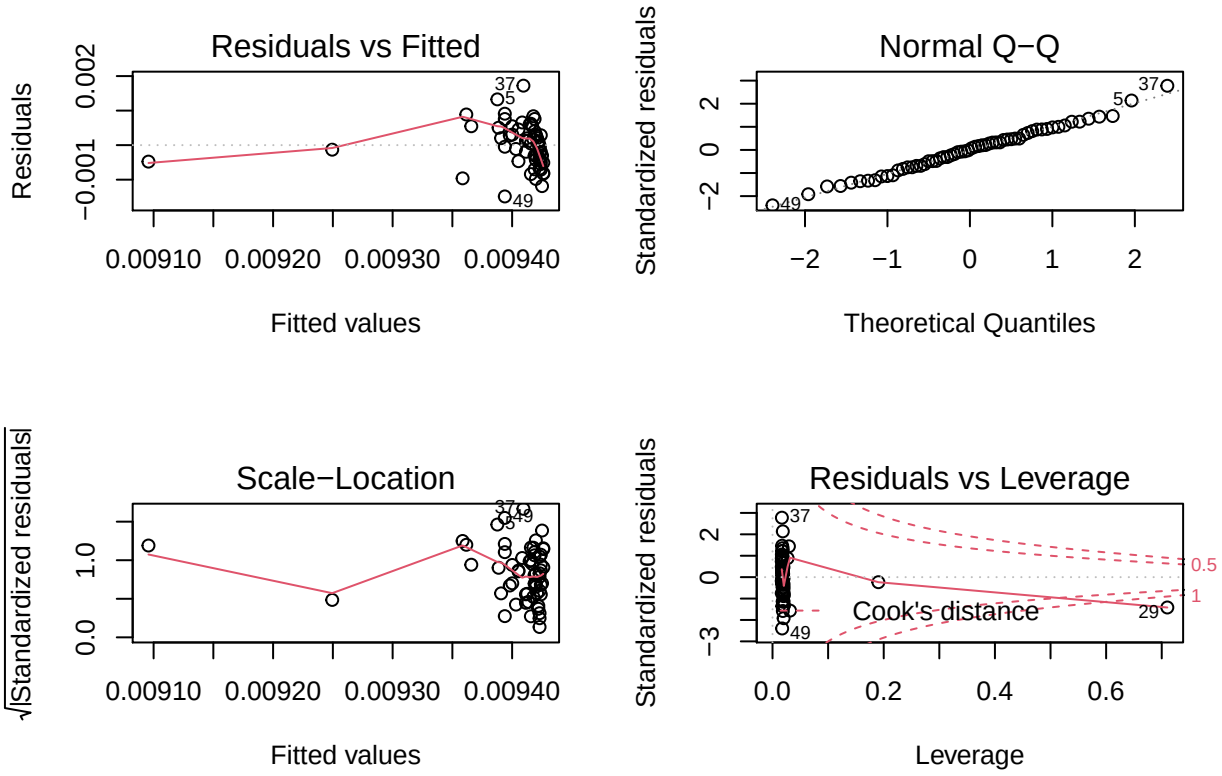

```r
fit1 <- lm(mort~nox, data = pollution)
print(fit1)
```

```
##
## Call:
## lm(formula = mort ~ nox, data = pollution)
##
## Coefficients:
## (Intercept)          nox
##   9.427e-03   -1.039e-06
```

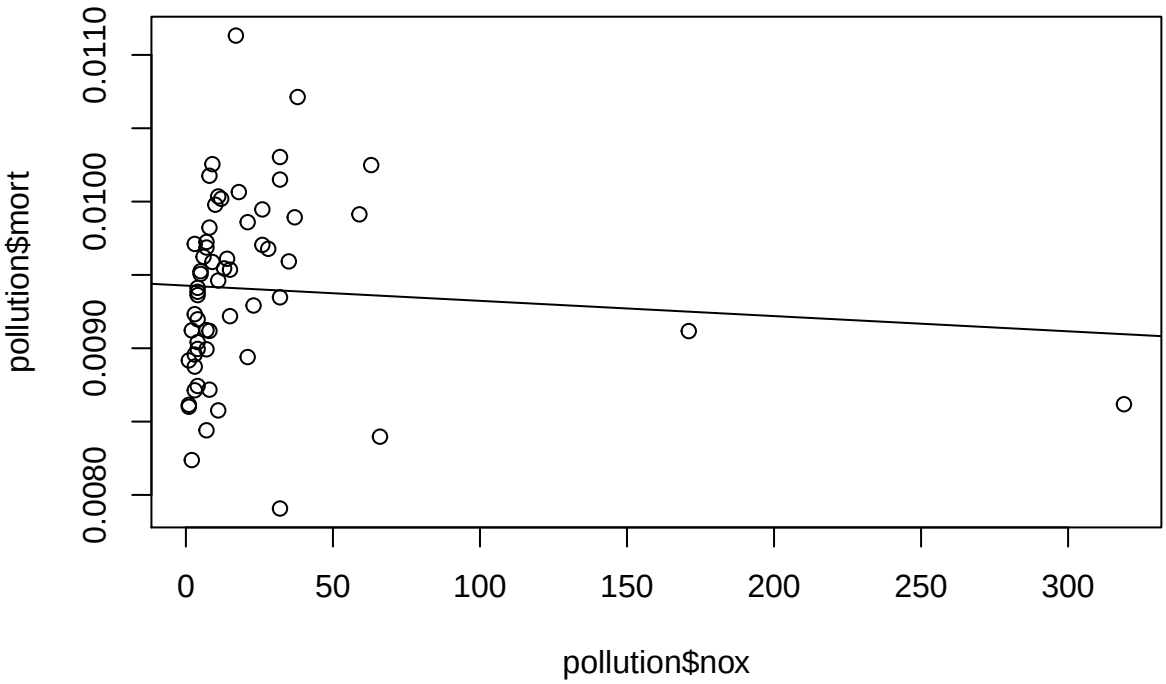
```r
summary(fit1)["r.squared"]
```

```
## $r.squared
## [1] 0.005987434
```

```r
par(mfrow=c(2,2))
plot(fit1)
```




```
par(mfrow=c(1,1))
plot(pollution$nox, pollution$mort)
abline(fit1)
```



I think linear regression will not fit the data well.

(b)

Find an appropriate reansformation that will result in data more appropriate for linear regression. Fit a regression to the transformed data and evaluate the new residual plot.

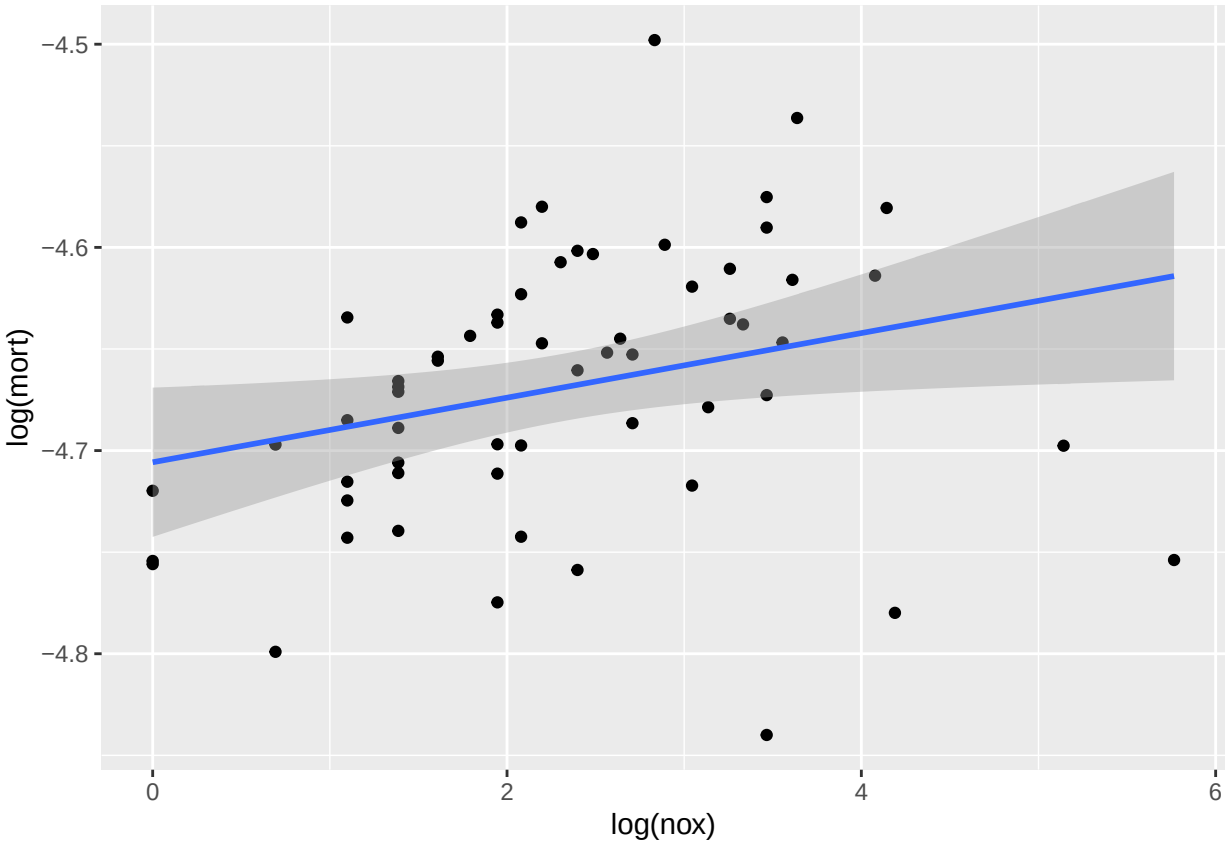
```
fit2 <- lm(log(mort)~log(nox), data = pollution)
print(fit2)
```

```
##
## Call:
## lm(formula = log(mort) ~ log(nox), data = pollution)
##
## Coefficients:
## (Intercept)     log(nox)
##    -4.70575      0.01589
```

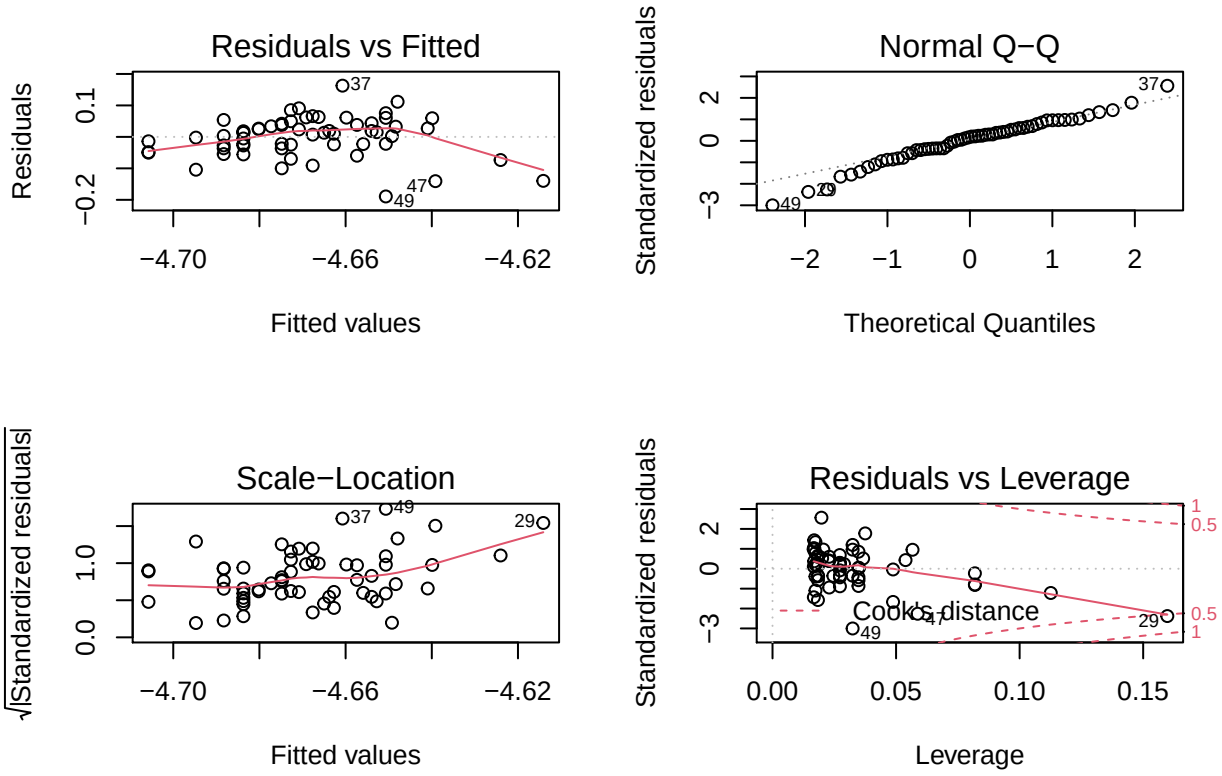
```
summary(fit2)["r.squared"]
```

```
## $r.squared
## [1] 0.0806116
```

```
ggplot(data = pollution, aes(x = log(nox), y = log(mort))) +
  geom_point() +
  stat_smooth(method = "lm", formula = y~x)
```




```
par(mfrow=c(2,2))
plot(fit2)
```



(c)

Interpret the slope coefficient from the model you chose in (b)

```
exp(coef(fit2)[1])
```

```
## (Intercept)
## 0.009043123
```

The intercept. exp(-4.7)=0.9% is the average mortality rate.
The coefficient of log(nox).For each 1% difference in height, the predicted difference in mortality rate is 0.02%.

(d)

Now fit a model predicting mortality rate using levels of nitric oxides, sulfur dioxide, and hydrocarbons as inputs. Use appropriate transformation when helpful. Plot the fitted regression model and interpret the coefficients.

```
apply(pollution[, c("hc", "nox", "so2")], FUN=IQR, MARGIN = 2)
```

```
##    hc   nox   so2
## 23.25 19.75 58.00
```

```
scale2 <- function(X) (X - mean(X)) / (2*sd(X))
pollution[, c("z.hc", "z.nox", "z.so2")] <- apply(pollution[, c("hc", "nox", "so2")], FUN=scale2, MARGI

apply(pollution[, c("z.hc", "z.nox", "z.so2")], FUN=IQR, MARGIN = 2)
```

```
##      z.hc     z.nox     z.so2
## 0.1263894 0.2131297 0.4574820
```

```r
fit3 <- lm(log(mort) ~ z.nox + z.so2 + z.hc, data=pollution)
summary(fit3)
```

```
##
## Call:
## lm(formula = log(mort) ~ z.nox + z.so2 + z.hc, data = pollution)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.112676 -0.033540 -0.003781  0.041982  0.168553
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -4.668821   0.007098 -657.753  < 2e-16 ***
## z.nox        0.296217   0.124494    2.379  0.02077 *
## z.so2        0.026428   0.023236    1.137  0.26022
## z.hc        -0.323153   0.118398   -2.729  0.00846 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05498 on 56 degrees of freedom
## Multiple R-squared:  0.3473, Adjusted R-squared:  0.3123
## F-statistic: 9.931 on 3 and 56 DF,  p-value: 2.39e-05
```

Intercept: The mortality rate for an individual exposed to average levels of nitric oxides, sulfur dioxide, and hydrocarbons is exp(-4.67) = 0.00937 = 0.94%.
z.nox: 1 standard deviation difference for nitric oxides, all rest being average, corresponds to a mortality rate exp(0.30) = 1.34985 times higher, which is 35% more.
z.so2: 1 standard deviation difference for sulfur dioxide corresponds to 0.03% increase in mortality rate.
z.hc: 1 standard deviation difference in hydrocarbons, all rest being average, corresponds to a mortality rate exp(-0.32) = 0.726149 times lower, which is a decrease of 27%.

(e)

Cross validate: fit the model you chose above to the first half of the data and then predict for the second half. You used all the data to construct the model in (d), so this is not really cross validation, but it gives a sense of how the steps of cross validation can be implemented.

```r
train <- pollution[1:(nrow(pollution)/2), ]
test <- pollution[((nrow(pollution)/2)+1):nrow(pollution), ]

fit4 <- lm(log(mort) ~ z.nox + z.so2 + z.hc, data=train)
summary(fit4)
```

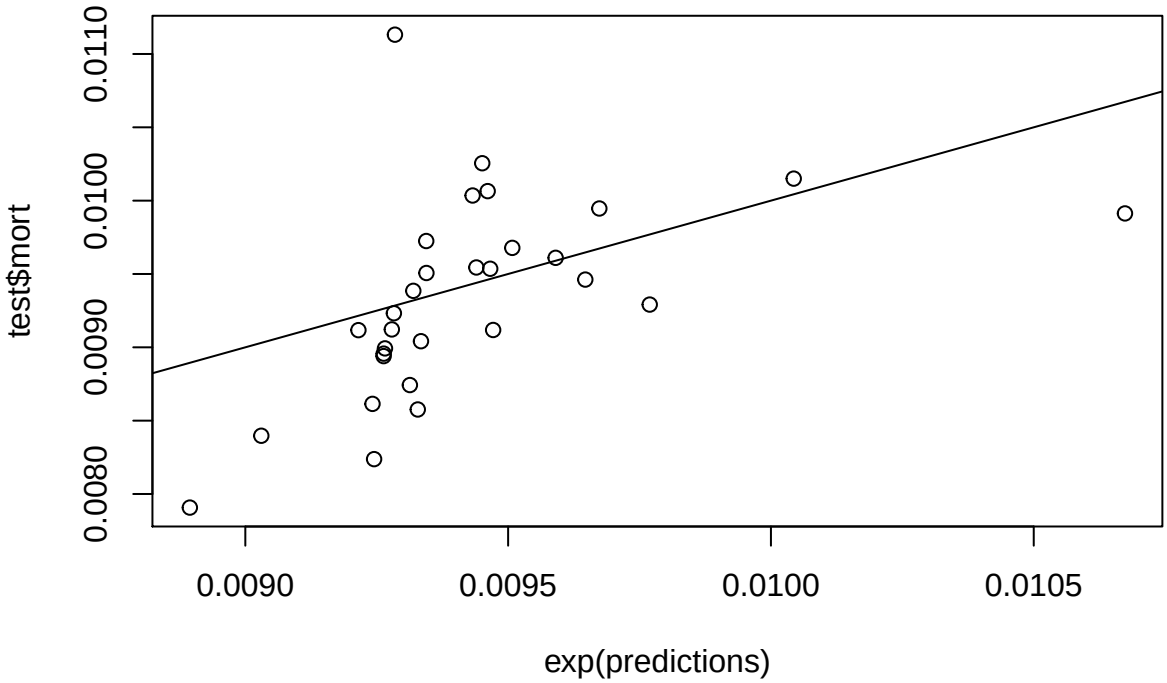
```
##
## Call:
## lm(formula = log(mort) ~ z.nox + z.so2 + z.hc, data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.104750 -0.029486 -0.004945  0.036401  0.088267
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -4.663132   0.009281 -502.439   <2e-16 ***
```

```
## z.nox        0.095321   0.214682    0.444    0.661
## z.so2        0.054983   0.032040    1.716    0.098 .
## z.hc        -0.128191   0.202298   -0.634    0.532
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05038 on 26 degrees of freedom
## Multiple R-squared:  0.3757, Adjusted R-squared:  0.3037
## F-statistic: 5.216 on 3 and 26 DF,  p-value: 0.005925
```

```
predictions <- predict(fit4, test)
cbind(predictions=exp(predictions), observed=test$mort)
```

```
##    predictions   observed
## 31 0.009461116 0.01006490
## 32 0.009241960 0.00861439
## 33 0.009769405 0.00929150
## 34 0.009328144 0.00857622
## 35 0.009590416 0.00961009
## 36 0.009282680 0.00923234
## 37 0.009284681 0.01113156
## 38 0.009673537 0.00994648
## 39 0.010043378 0.01015023
## 40 0.010673597 0.00991290
## 41 0.009263130 0.00893991
## 42 0.009319622 0.00938500
## 43 0.009646836 0.00946185
## 44 0.009450756 0.01025502
## 45 0.009313130 0.00874281
## 46 0.009465939 0.00953560
## 47 0.009030423 0.00839709
## 48 0.009215209 0.00911701
## 49 0.008894473 0.00790733
## 50 0.009265700 0.00899264
## 51 0.009334211 0.00904155
## 52 0.009344545 0.00950672
## 53 0.009344234 0.00972464
## 54 0.009278740 0.00912202
## 55 0.009507915 0.00967803
## 56 0.009245027 0.00823764
## 57 0.009432453 0.01003502
## 58 0.009263294 0.00895696
## 59 0.009471499 0.00911817
## 60 0.009439589 0.00954442
```

```
plot(exp(predictions), test$mort)
abline(a=0, b=1)
```




```
sqrt(mean((test$mort-exp(predictions))^2))
```

```
## [1] 0.0005810359
```

12.7

Cross validation comparison of models with different transformations of outcomes: when we compare models with transformed continuous outcomes, we must take into account how the nonlinear transformation warps the continuous outcomes. Follow the procedure used to compare models for the mesquite bushes example on page 202.

```
earning <- read.csv("/Users/amelia/Documents/mssp/MA678/hw2/earnings.csv", header = T)
earning$log_earnk <- log(earning$earnk)
df <- earning[,c(1,3,5,16)]
```

(a)

Compare models for earnings and for log(earnings) given height and sex as shown in page 84 and 192. Use earnk and log(earnk) as outcomes.

```
#earnk~height+male
fit1 <- stan_glm(earnk ~ height + male, data = df, subset = earnk > 0, refresh=0)
print(fit1)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      earnk ~ height + male
##  observations: 1629
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept) -19.1   12.7
## height        0.6    0.2
## male          8.8    1.5
```

```
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 21.7    0.4
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
kfold_1 <- kfold(fit1, K=10)
```

```
## Fitting model 1 out of 10
## Fitting model 2 out of 10
## Fitting model 3 out of 10
## Fitting model 4 out of 10
## Fitting model 5 out of 10
## Fitting model 6 out of 10
## Fitting model 7 out of 10
## Fitting model 8 out of 10
## Fitting model 9 out of 10
## Fitting model 10 out of 10
```

```
fit2 <- stan_glm(log(earnk) ~ height + male, data = earning, subset = earnk > 0, refresh = 0)
print(fit2)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      log(earnk) ~ height + male
##  observations: 1629
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept) 1.1    0.5
## height      0.0    0.0
## male        0.4    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.9    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
(loo_1 <- loo(fit1))
```

```
## Warning: Found 1 observation(s) with a pareto_k > 0.7. We recommend calling 'loo' again with argumen
##
## Computed from 4000 by 1629 log-likelihood matrix
##
##          Estimate    SE
```

```
## elpd_loo  -7339.8 166.9
## p_loo        28.2  20.8
## looic     14679.5 333.8
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     1628  99.9%   701
##  (0.5, 0.7]   (ok)          0   0.0%   <NA>
##    (0.7, 1]   (bad)         0   0.0%   <NA>
##    (1, Inf)   (very bad)    1   0.1%   3
## See help('pareto-k-diagnostic') for details.
```

```
(loo_2 <- loo(fit2))
```

```
##
## Computed from 4000 by 1629 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2083.5 38.7
## p_loo         4.8  0.4
## looic      4167.1 77.5
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
loo_2_with_jacobian <- loo_2
loo_2_with_jacobian$pointwise[,1] <- loo_2_with_jacobian$pointwise[,1] - log(df$earnk[df$earnk>0])
(elpd_loo_2_with_jacobian <- sum(loo_2_with_jacobian$pointwise[,1]))
```

```
## [1] -6663.212
```

```
loo_compare(kfold_1, loo_2_with_jacobian)
```

```
## Warning: Not all models have the same y variable. ('yhash' attributes do not
## match)
```

```
## Warning: Comparing LOO-CV to K-fold-CV. For a more accurate comparison use the
## same number of folds or loo for all models compared.
```

```
##      elpd_diff se_diff
## fit2    0.0       0.0
## fit1 -673.1     154.9
```

(b)

Compare models from other exercises in this chapter.

12.8

Log-log transformations: Suppose that, for a certain population of animals, we can predict log weight from log height as follows:

- An animal that is 50 centimeters tall is predicted to weigh 10 kg.
- Every increase of 1% in height corresponds to a predicted increase of 2% in weight.

- The weights of approximately 95% of the animals fall within a factor of 1.1 of predicted values.

(a)

Give the equation of the regression line and the residual standard deviation of the regression.

```
log(10) - 2*log(50)
```

```
## [1] -5.521461
```

```
sigma <- log(1.1)
sigma
```

```
## [1] 0.09531018
```

The regression line is log(weight) = -5.52 + 2log(height).
The residual standard deviation of the regression is 0.1.

(b)

Suppose the standard deviation of log weights is 20% in this population. What, then, is the R^2 of the regression model described here?

```
r_square <- 1 - (sigma^2/0.2^2)
r_square
```

```
## [1] 0.7728992
```

12.9

Linear and logarithmic transformations: For a study of congressional elections, you would like a measure of the relative amount of money raised by each of the two major-party candidates in each district. Suppose that you know the amount of money raised by each candidate; label these dollar values Di and Ri. You would like to combine these into a single variable that can be included as an input variable into a model predicting vote share for the Democrats. Discuss the advantages and disadvantages of the following measures:

(a)

The simple difference, $D_i - R_i$
Advantage:
The transformation is symmetric and centered at zero.
Disadvantage:
The transformation is not propotional. For example, when $D_i - R_i = 2$, we don't know D_i/R_i. So this may limits the efficientiveness of the predictor.

(b)

The ratio, D_i/R_i
Advantage:
The transformation is propotional. Disadvantage:
The transformation is asymmetric. If $D_i \gg R_i$, $D_i/R_i \to \infty$. And if $D_i \ll R_i$, $D_i/R_i \to 0$.This means that it will weight more cases where Democrats raised more money than the opposite party.

(c)

The difference on the logarithmic scale, $log\ D_i - log\ R_i$
Advantage:
The transformation is less sensitive to outliers. It is centered to zero and is symmetric. It is also proportional to the magnitude of the difference.

(d)

The relative proportion, $D_i/(D_i + R_i)$.
Advantage:
this transformation is centered at 0.5 and symmetric.

12.11

Elasticity: An economist runs a regression examining the relations between the average price of cigarettes, P, and the quantity purchased, Q, across a large sample of counties in the United States, assuming the functional form, $logQ = \alpha + \beta logP$. Suppose the estimate for β is 0.3. Interpret this coefficient.
For each 1% difference in cigarettes price, the predicted difference in quantity purchased is 0.3%.

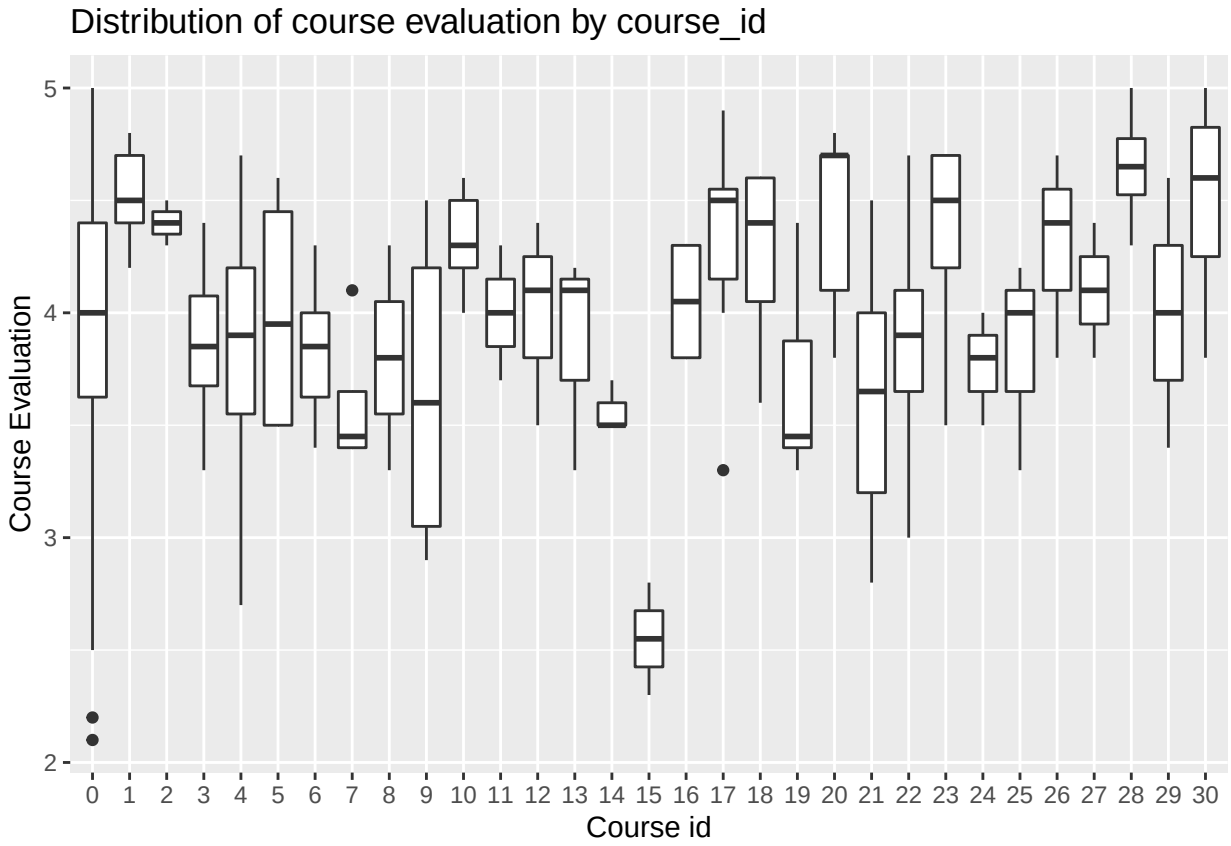
12.13

Building regression models: Return to the teaching evaluations data from Exercise 10.6. Fit regression models predicting evaluations given many of the inputs in the dataset. Consider interactions, combinations of predictors, and transformations, as appropriate. Consider several models, discuss in detail the final model that you choose, and also explain why you chose it rather than the others you had considered.

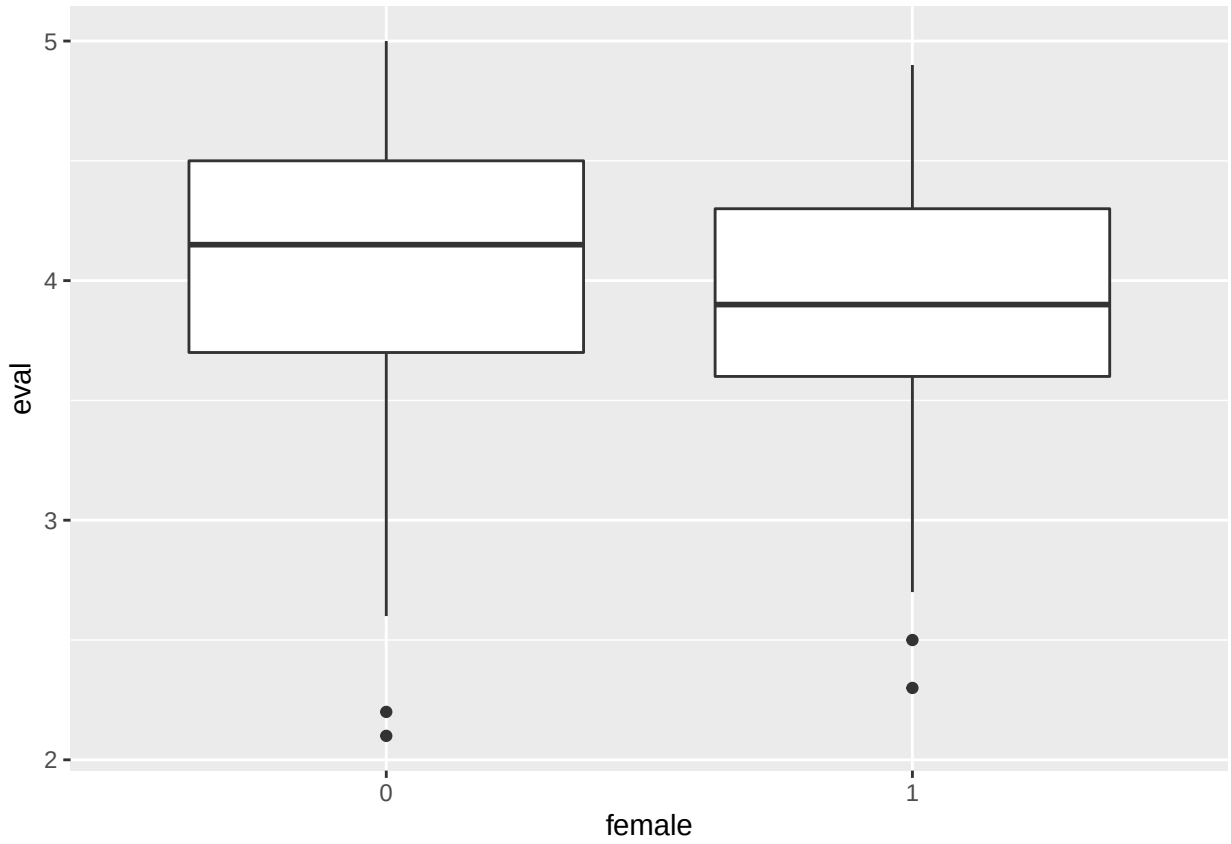
```
beauty <- read.csv("/Users/amelia/Documents/mssp/MA678/hw1/beauty.csv", header=T)
df <- beauty
```

Using the Beauty datasets, we will try to predict the course evaluations given by the students, based on a number of factors. In our dataset, each row corresponds in a score. There are many variables included in the dataset, including some characteristic of the professor (i.e. age, female, minority, beauty, minority) and the class (i.e. nonenglish, lower, course_id).

```
df$course_id <- as.factor(df$course_id)
df$female <- as.factor(df$female)
# boxplot of course_id vs course evaluation
ggplot(data=df, aes(x=course_id, y=eval)) + geom_boxplot() +
  labs(title="Distribution of course evaluation by course_id", x="Course id", y="Course Evaluation")
```




```
# boxplot of female vs course evaluation
ggplot(data=df, aes(x = female, y=eval)) + geom_boxplot()
```




```
fit1 <- lm(eval~female + beauty + age + minority + nonenglish + lower + course_id, data = beauty)
summary(fit1)["r.squared"]
```

```
## $r.squared
## [1] 0.09578021
```

```
fit2 <- lm(eval~female + beauty + age + minority + nonenglish + lower, data = beauty)
summary(fit2)["r.squared"]
```

```
## $r.squared
## [1] 0.09574628
```

```
fit3 <- lm(log(eval)~female + beauty + age + minority + nonenglish + lower, data = beauty)
summary(fit3)["r.squared"]
```

```
## $r.squared
## [1] 0.08938975
```

```
fit4 <- lm(eval~female + beauty + minority + nonenglish + lower + female:beauty + beauty:age, data = bea
summary(fit4)["r.squared"]
```

```
## $r.squared
## [1] 0.1218656
```

```
summary(fit4)
```

```
##
## Call:
## lm(formula = eval ~ female + beauty + minority + nonenglish +
##     lower + female:beauty + beauty:age, data = beauty)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.79645 -0.36964  0.02188  0.41657  1.13214
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.132014   0.040974 100.845  < 2e-16 ***
## female        -0.206393   0.050644  -4.075 5.42e-05 ***
## beauty        -0.417148   0.187230  -2.228 0.026369 *
## minority      -0.076150   0.078186  -0.974 0.330595
## nonenglish    -0.311716   0.109706  -2.841 0.004694 **
## lower          0.070046   0.053829   1.301 0.193823
## female:beauty  0.014719   0.073643   0.200 0.841671
## beauty:age     0.011903   0.003532   3.370 0.000815 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5239 on 455 degrees of freedom
## Multiple R-squared:  0.1219, Adjusted R-squared:  0.1084
## F-statistic: 9.021 on 7 and 455 DF,  p-value: 1.943e-10
```

The model I choose is eval=4.19 - 0.21female - 0.42beauty - 0.08minority - 0.31nonenglish + 0.07lower + 0.01female:beauty + 0.01beauty:age.
I choose this model because it has the biggest R^2.

12.14

Prediction from a fitted regression: Consider one of the fitted models for mesquite leaves, for example fit_4, in Section 12.6. Suppose you wish to use this model to make inferences about the average mesquite yield in a new set of trees whose predictors are in data frame called new_trees. Give R code to obtain an estimate and standard error for this population average. You do not need to make the prediction; just give the code.

```
#df <- read.table("/Users/amelia/Documents/mssp/MA678/hw2/mesquite.txt", header = T)

#log(weight) ~ log(canopy_volume) + log(canopy_area) + log(canopy_shape) + log(total_height) + log(dens
#new_trees <- data.frame()
#new_trees$canopy_volume <- new_trees$diam1 * new_trees$diam2 * #new_trees$canopy_height
#new_trees$canopy_area <- new_trees$diam1 * new_trees$diam2
#new_trees$canopy_shape <- new_trees$diam1 / new_trees$diam2
#new_trees$predicted <- exp(log(new_trees$canopy_volume) + log(new_trees$canopy_area) + log(new_trees$c

#estimate <- mean(new_trees$predicted)
#std <- sd(new_trees$predicted)
```