MA678 Homework 2

Chang Lu

9/20/2022

11.5

Residuals and predictions: The folder `Pyth` contains outcome y and predictors x_1, x_2 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()`.

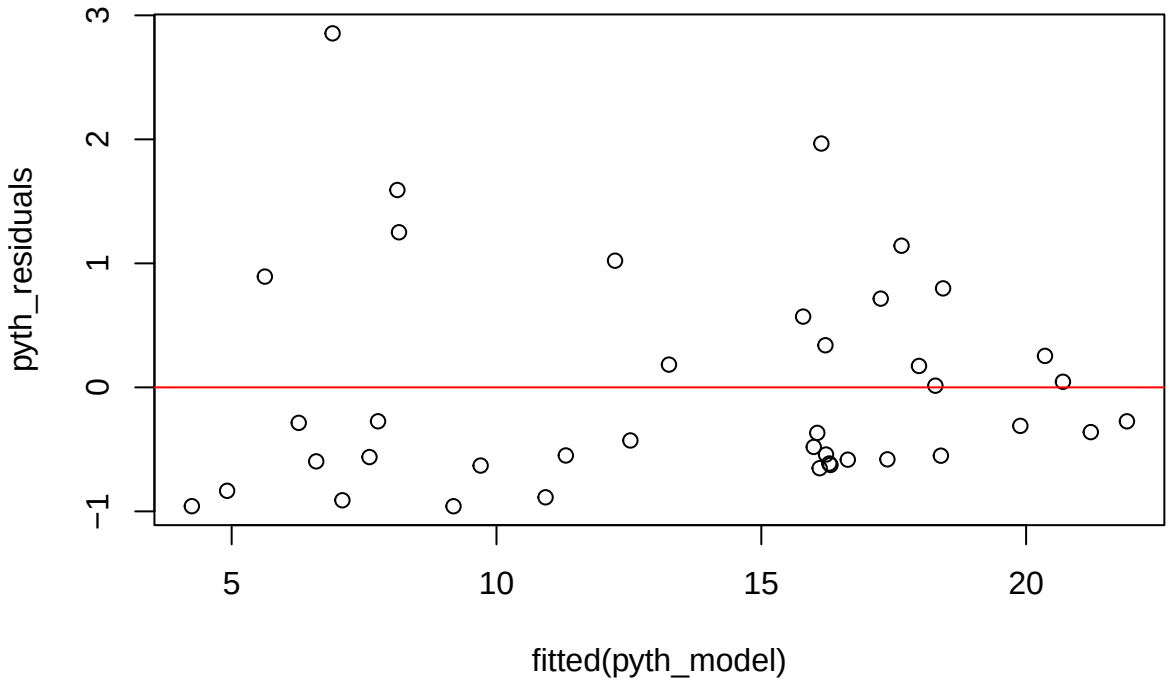
(a)

Use R to fit a linear regression model predicting y from x_1, x_2, using the first 40 data points in the file. Summarize the inferences and check the fit of your model.

```
pyth_data <- read.table("pyth.txt",header = TRUE)
pyth_data_subset <- pyth_data[0:40, ]
pyth_model <- lm(y ~ x1 + x2, data = pyth_data_subset)
summary(pyth_model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = pyth_data_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9585 -0.5865 -0.3356  0.3973  2.8548
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.31513    0.38769   3.392  0.00166 **
## x1           0.51481    0.04590  11.216 1.84e-13 ***
## x2           0.80692    0.02434  33.148  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9 on 37 degrees of freedom
## Multiple R-squared:  0.9724, Adjusted R-squared:  0.9709
## F-statistic: 652.4 on 2 and 37 DF,  p-value: < 2.2e-16
```

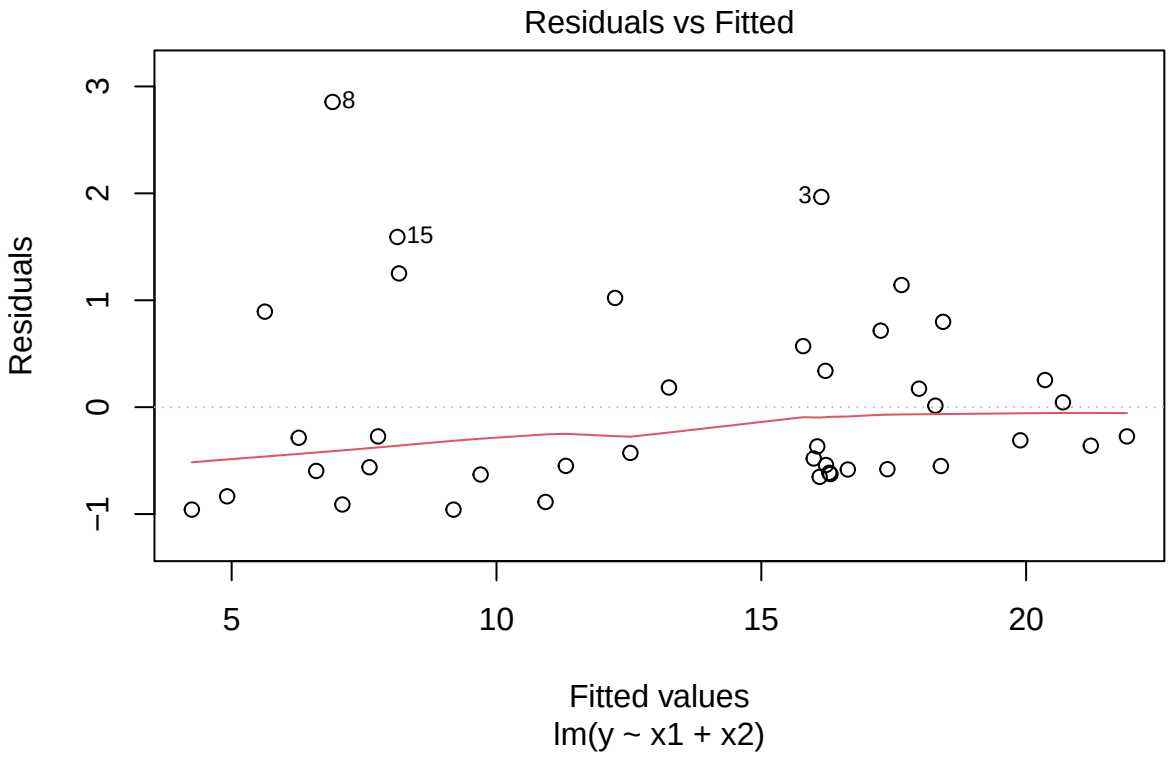
```
pyth_residuals <- residuals(pyth_model)
plot(fitted(pyth_model),pyth_residuals)
abline(h = 0, col="red")
```

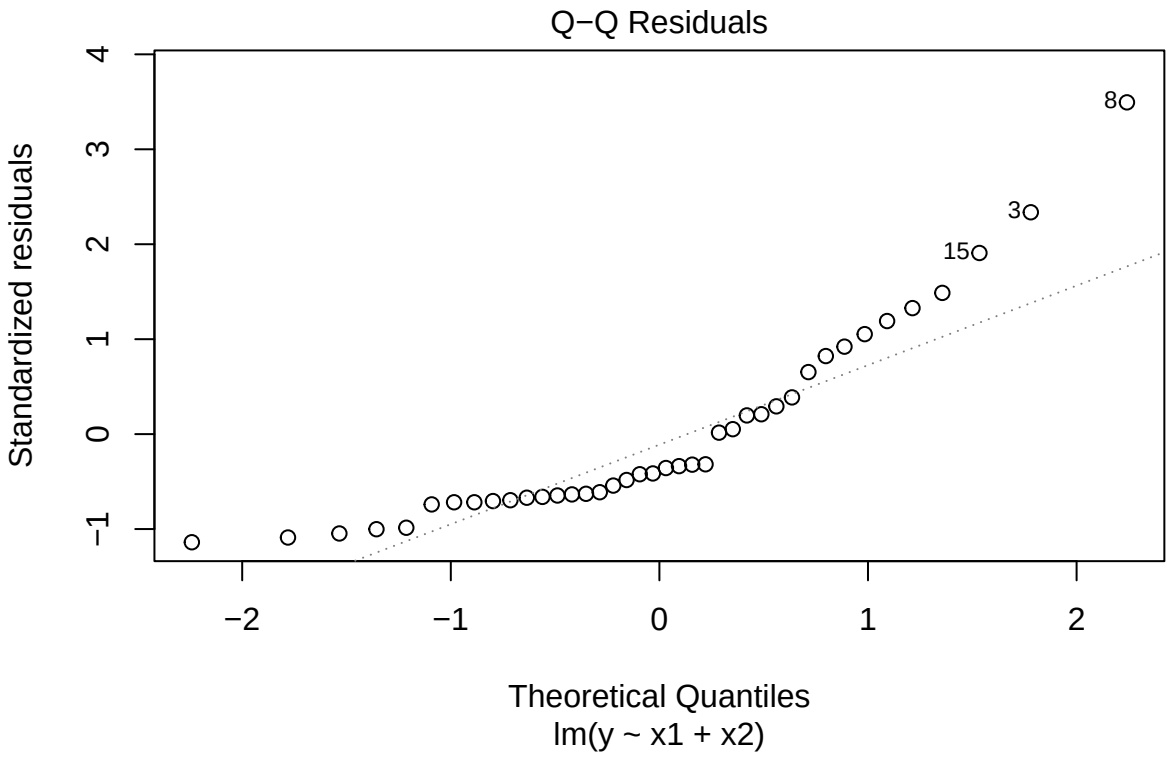


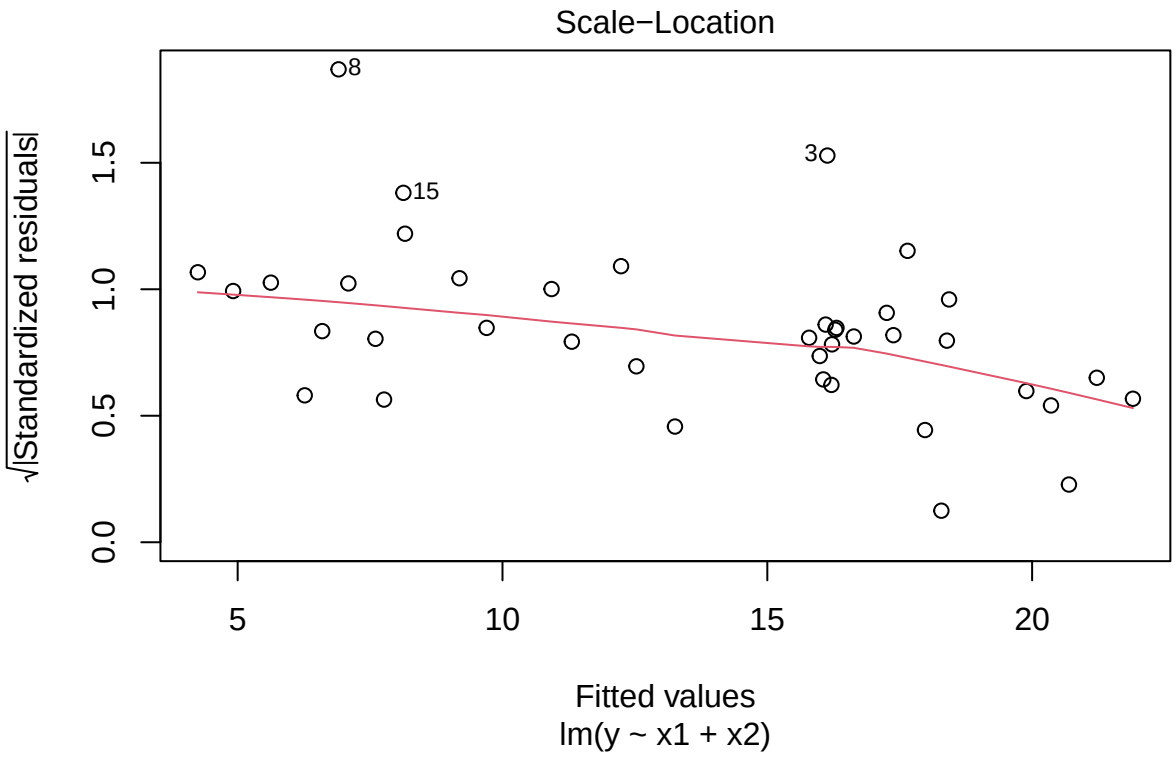
(b)

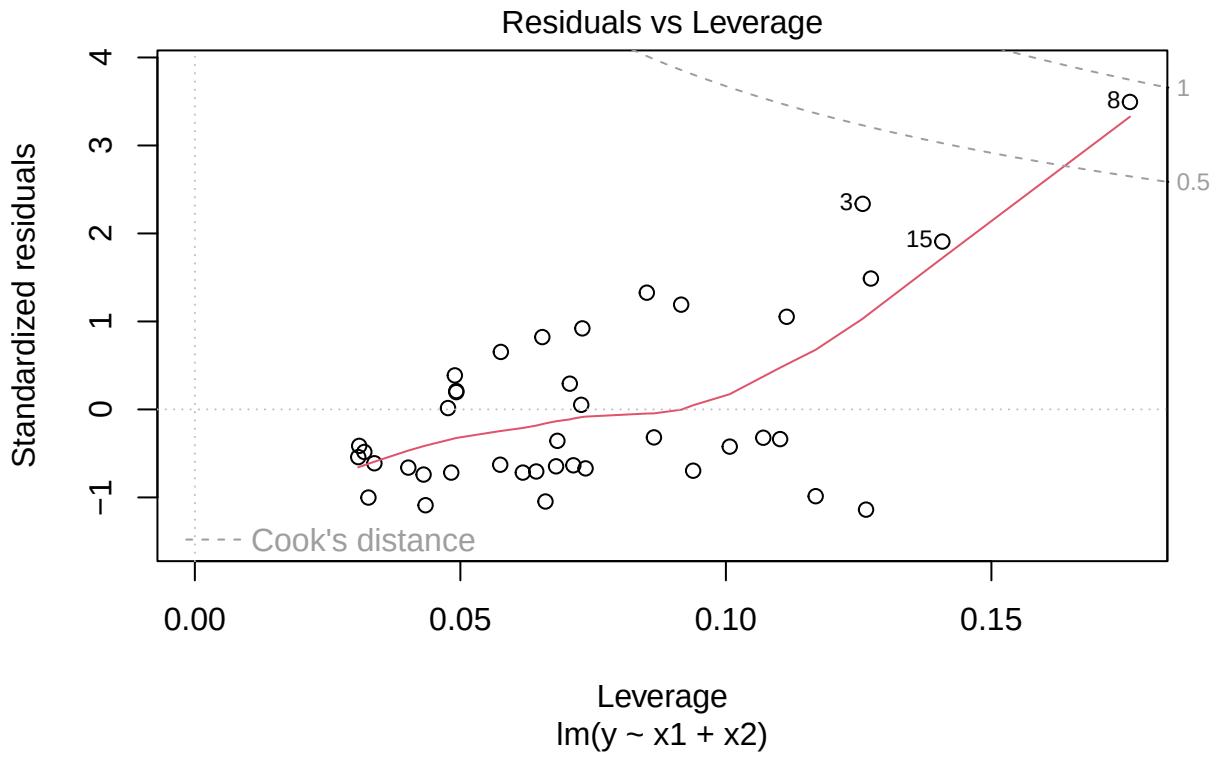
Display the estimated model graphically as in Figure 10.2

```
plot(pyth_model)
```


Residuals vs Fitted
Residuals
Fitted values
lm(y ~ x1 + x2)
8
15
3
−1
0
1
2
3
5
10
15
20


Q−Q Residuals
Standardized residuals
−1
0
1
2
3
4
8
3
15
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x1 + x2)

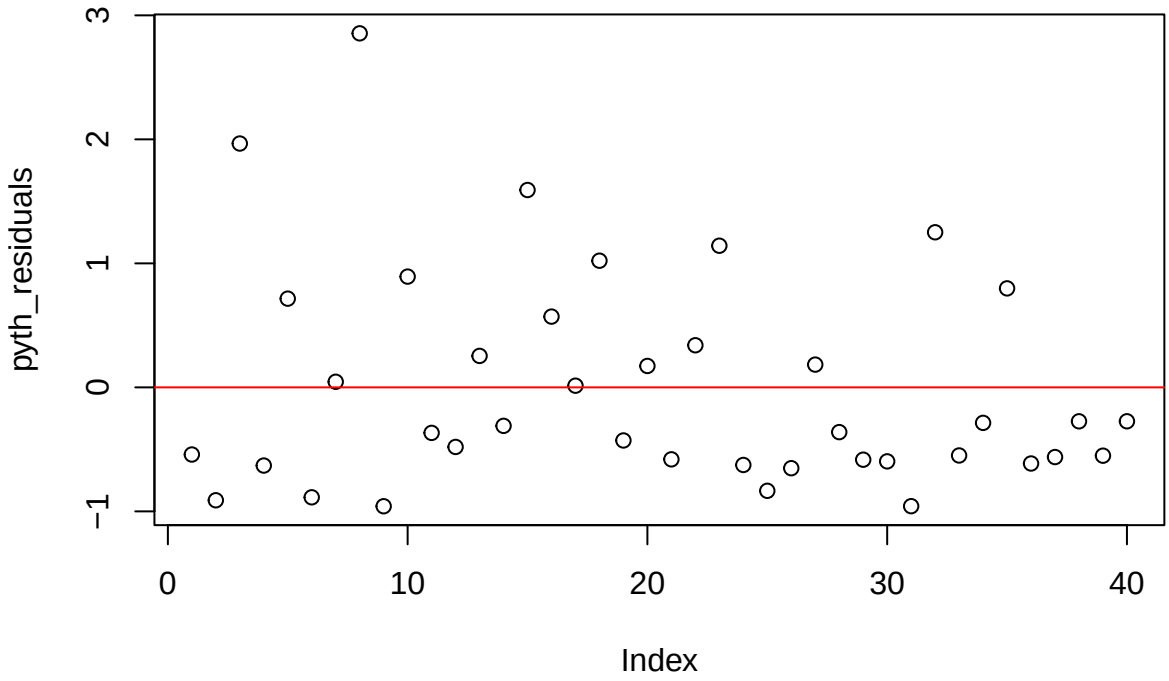

Scale−Location
8
15
3
√|Standardized residuals|
0.0
0.5
1.0
1.5
5
10
15
20
Fitted values
lm(y ~ x1 + x2)



(c)

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

```
pyth_residuals <- residuals(pyth_model)
plot(pyth_residuals)
abline(h = 0, col="red")
```



1. The residuals seem randomly scattered around the horizontal line (zero), with no clear pattern, which suggests that linearity is met.

2. The spread of residuals seems fairly consistent across the index range, indicating that homoscedasticity is likely met.

As a result, the model assumptions seem to be generally met.

(d)

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

```
newpyth_data <- pyth_data[41:60,]
pyth_prediction <- predict(pyth_model, data = newpyth_data)
pyth_prediction_interval <-  predict(pyth_model, data = newpyth_data, interval = "confidence")
pyth_prediction_interval
```

```
##          fit       lwr       upr
## 1  16.221378 15.886052 16.556704
## 2   7.090400  6.621804  7.558997
## 3  16.133677 15.486970 16.780385
## 4   9.700421  9.299704 10.101138
## 5  17.254606 16.788121 17.721090
## 6  10.926653 10.596958 11.256349
```

```
## 7  20.694923 20.203020 21.186826
## 8   6.905222  6.139964  7.670480
## 9   9.188536  8.808430  9.568642
## 10  5.626640  5.017832  6.235448
## 11 16.057014 15.736186 16.377841
## 12 15.990263 15.670355 16.310171
## 13 20.356247 19.871665 20.840828
## 14 19.890290 19.413804 20.366777
## 15  8.128384  7.444186  8.812582
## 16 15.789193 15.351429 16.226957
## 17 18.286339 17.888176 18.684501
## 18 12.238483 11.686614 12.790352
## 19 12.528169 12.202384 12.853955
## 20 17.977308 17.572644 18.381971
## 21 17.380425 16.885756 17.875095
## 22 16.210499 15.806995 16.614003
## 23 17.647595 17.115564 18.179626
## 24 16.305805 15.852592 16.759017
## 25  4.914170  4.290628  5.537713
## 26 16.101788 15.723323 16.480253
## 27 13.256420 12.851781 13.661058
## 28 21.220813 20.642038 21.799589
## 29 16.633082 16.267445 16.998719
## 30  6.596613  6.037982  7.155243
## 31  4.248042  3.599698  4.896386
## 32  8.159388  7.508710  8.810065
## 33 11.309235 10.871920 11.746551
## 34  6.266295  5.660875  6.871716
## 35 18.431754 17.939120 18.924389
## 36 16.283969 15.821576 16.746362
## 37  7.601885  7.126248  8.077522
## 38 21.903629 21.306994 22.500264
## 39 18.390868 17.903994 18.877742
## 40  7.763580  7.227368  8.299792
```

I would feel moderately confident in these predictions, especially for those with narrower confidence intervals, which indicating high precision and showing the model is quite confident with the predictions.

12.5

Logarithmic transformation and regression: Consider the following regression:

$$\log(\text{weight}) = -3.8 + 2.1 \log(\text{height}) + \text{error},$$

with errors that have standard deviation 0.25. Weights are in pounds and heights are in inches.

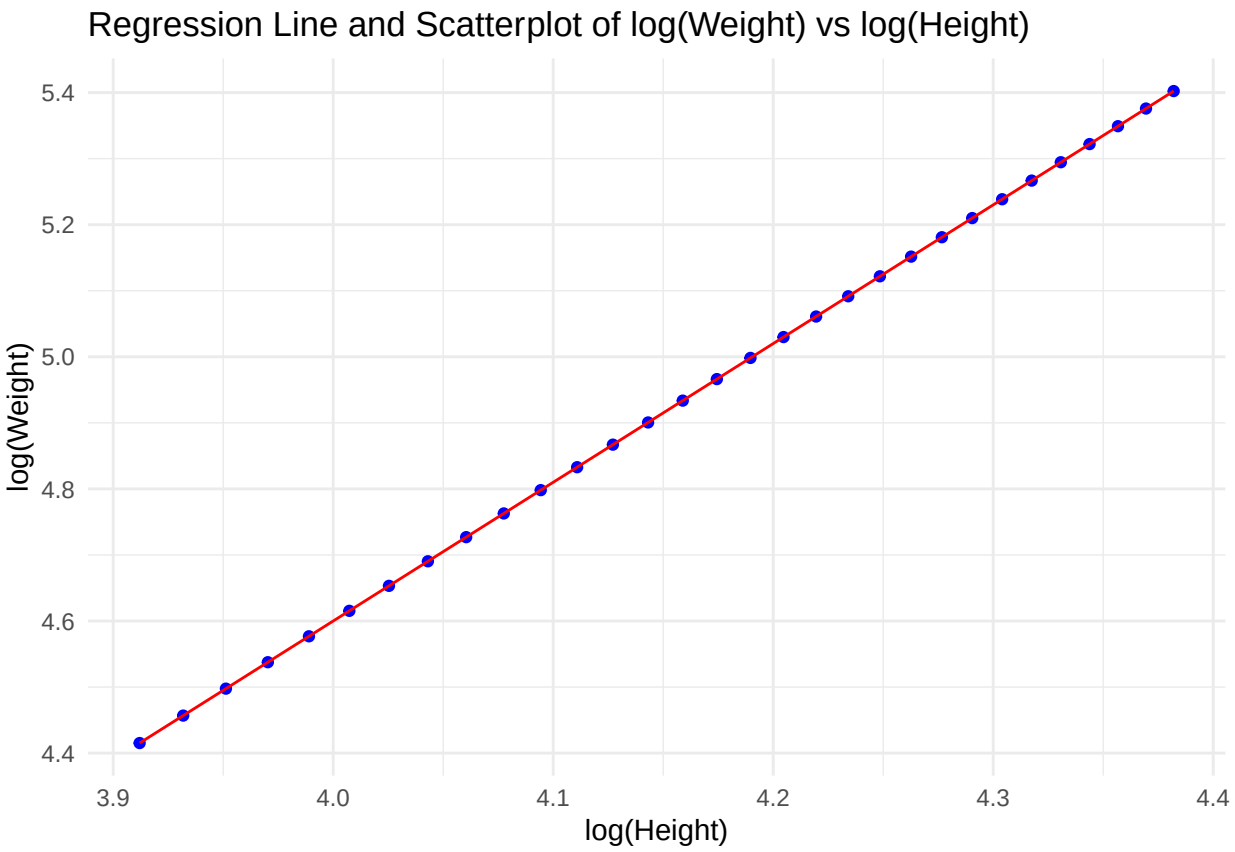
(a)

Fill in the blanks: Approximately 68% of the people will have weights within a factor of $0.778 (e^{-0.25})$ and $1.284 (e^{0.25})$ of their predicted values from the regression.

(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.

```
library(ggplot2)

height <- seq(50, 80, by = 1)
log_height <- log(height)

log_weight <- -3.8 + 2.1 * log_height

data <- data.frame(log_height, log_weight)

ggplot(data, aes(x = log_height, y = log_weight)) +
  geom_point(color = 'blue') +
  geom_line(color = 'red') +
  labs(x = "log(Height)", y = "log(Weight)") +
  ggtitle("Regression Line and Scatterplot of log(Weight) vs log(Height)") +
  theme_minimal()
```



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.

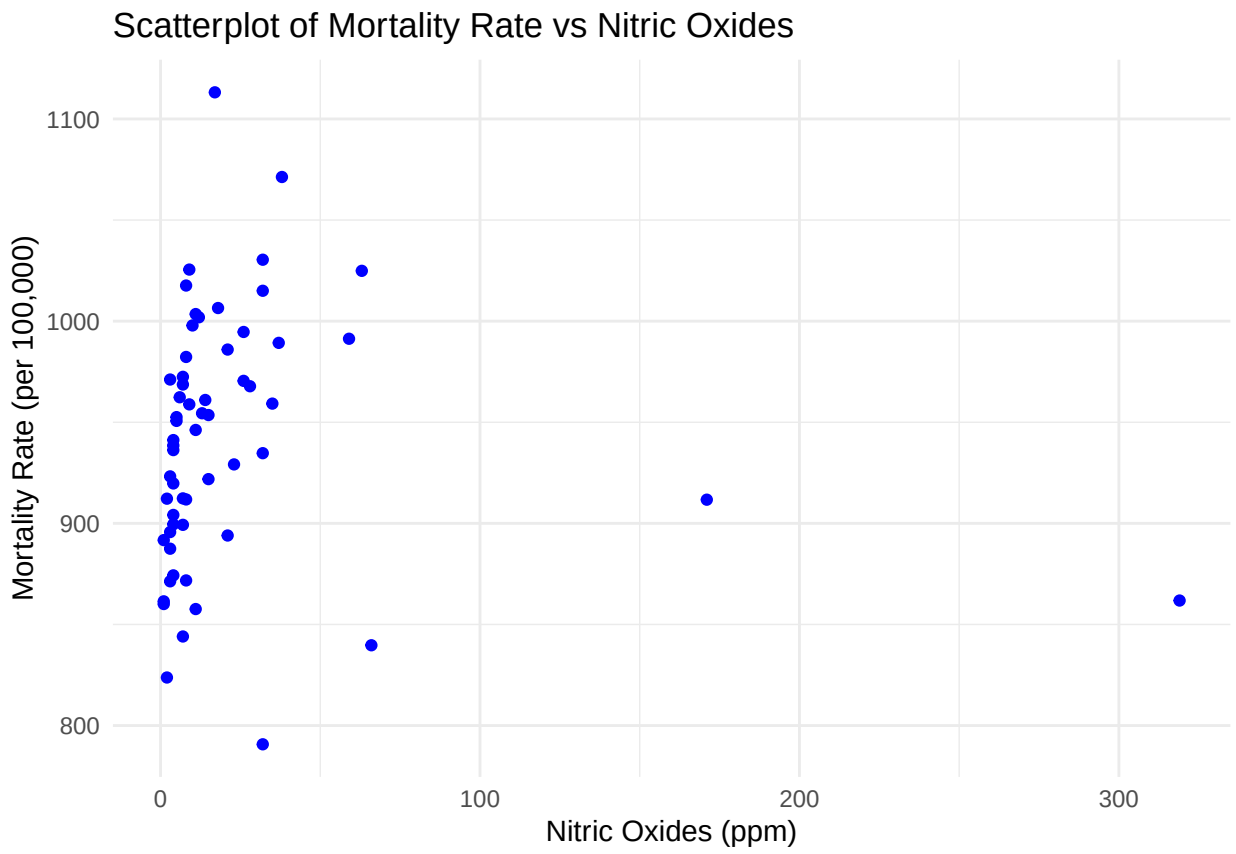
(a)

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.

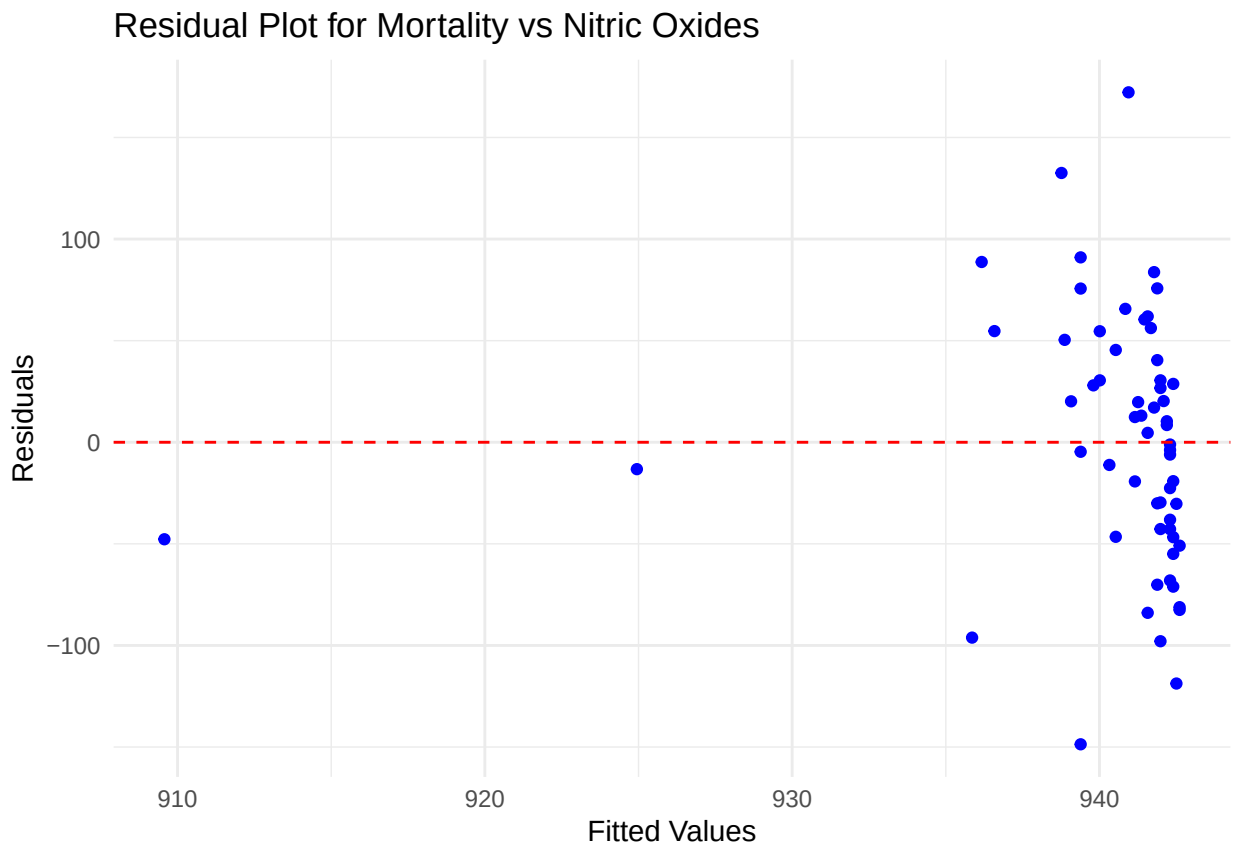
```
pollution_data <- read.csv("pollution.csv")

head(pollution_data)
```

```
##   prec jant jult ovr65 popn educ hous dens nonw wwdrk poor hc nox so2 humid
## 1   36   27   71   8.1 3.34 11.4 81.5 3243  8.8  42.6 11.7 21  15  59    59
## 2   35   23   72  11.1 3.14 11.0 78.8 4281  3.5  50.7 14.4  8  10  39    57
## 3   44   29   74  10.4 3.21  9.8 81.6 4260  0.8  39.4 12.4  6   6  33    54
## 4   47   45   79   6.5 3.41 11.1 77.5 3125 27.1  50.2 20.6 18   8  24    56
## 5   43   35   77   7.6 3.44  9.6 84.6 6441 24.4  43.7 14.3 43  38 206    55
## 6   53   45   80   7.7 3.45 10.2 66.8 3325 38.5  43.1 25.5 30  32  72    54
##       mort
## 1  921.870
## 2  997.875
## 3  962.354
## 4  982.291
## 5 1071.289
## 6 1030.380
```

```
ggplot(pollution_data, aes(x = nox, y = mort)) +
  geom_point(color = 'blue') +
  labs(x = "Nitric Oxides (ppm)", y = "Mortality Rate (per 100,000)") +
  ggtitle("Scatterplot of Mortality Rate vs Nitric Oxides") +
  theme_minimal()
```




```r
model_a <- lm(mort ~ nox, data = pollution_data)

ggplot(data.frame(Residuals = residuals(model_a), Fitted = fitted(model_a)), aes(x = Fitted, y = Residua
  geom_point(color = 'blue') +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(x = "Fitted Values", y = "Residuals") +
  ggtitle("Residual Plot for Mortality vs Nitric Oxides") +
  theme_minimal()
```



From the scatterplot of Mortality Rate vs Nitric Oxides, it is clear that the data does not exhibit a strong linear relationship. Most of the data points are clustered tightly near lower levels of nitric oxides, with some outliers at higher levels, which introduces heteroscedasticity. This suggests that a simple linear regression may not be the best fit for this data due to the presence of non-linearity and potentially influential outliers.

Additionally, the residual plot reveals a pattern where the residuals are not randomly dispersed around zero, especially toward higher fitted values. This is a further indication that the linear model is not adequately capturing the underlying relationship between mortality and nitric oxides. There is also potential non-constant variance (heteroscedasticity), which violates one of the assumptions of linear regression.

Based on this, a linear regression model likely does not fit the data well.

(b)

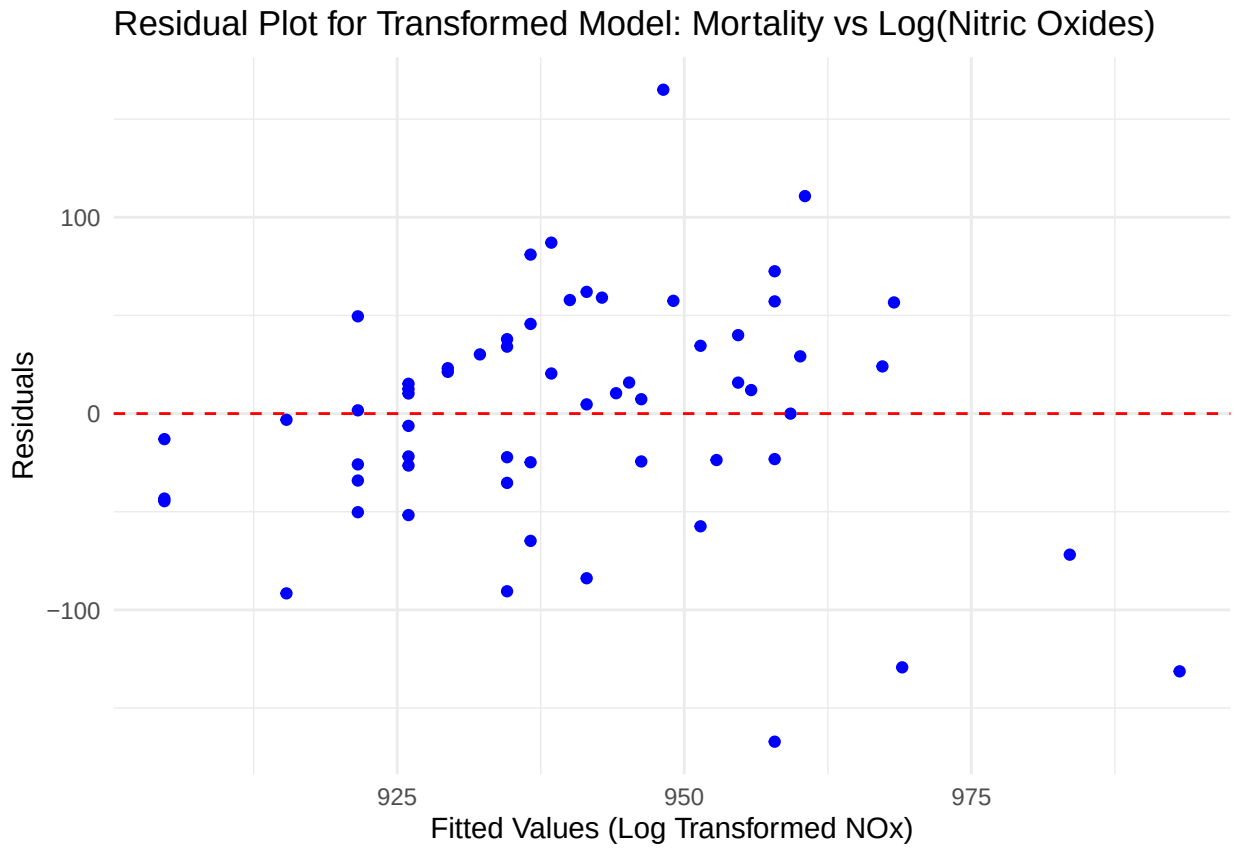
Find an appropriate reansformation that will result in data more appropriate for linear regression. Fit a regression to the transformed data and evaluate the new residual plot.

```
pollution_data$log_nox <- log(pollution_data$nox)

model_b <- lm(mort ~ log_nox, data = pollution_data)

summary(model_b)
```

```
##
## Call:
## lm(formula = mort ~ log_nox, data = pollution_data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -167.140  -28.368    8.778   35.377  164.983
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  904.724     17.173  52.684   <2e-16 ***
## log_nox       15.335      6.596   2.325   0.0236 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.01 on 58 degrees of freedom
## Multiple R-squared:  0.08526,    Adjusted R-squared:  0.06949
## F-statistic: 5.406 on 1 and 58 DF,  p-value: 0.02359
```

```
ggplot(data.frame(Residuals = residuals(model_b), Fitted = fitted(model_b)), aes(x = Fitted, y = Residua
  geom_point(color = 'blue') +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(x = "Fitted Values (Log Transformed NOx)", y = "Residuals") +
  ggtitle("Residual Plot for Transformed Model: Mortality vs Log(Nitric Oxides)") +
  theme_minimal()
```



The Log-transformed residual plot suggests that the log transformation has helped linearize the relationship between mortality and nitric oxides.

(c)

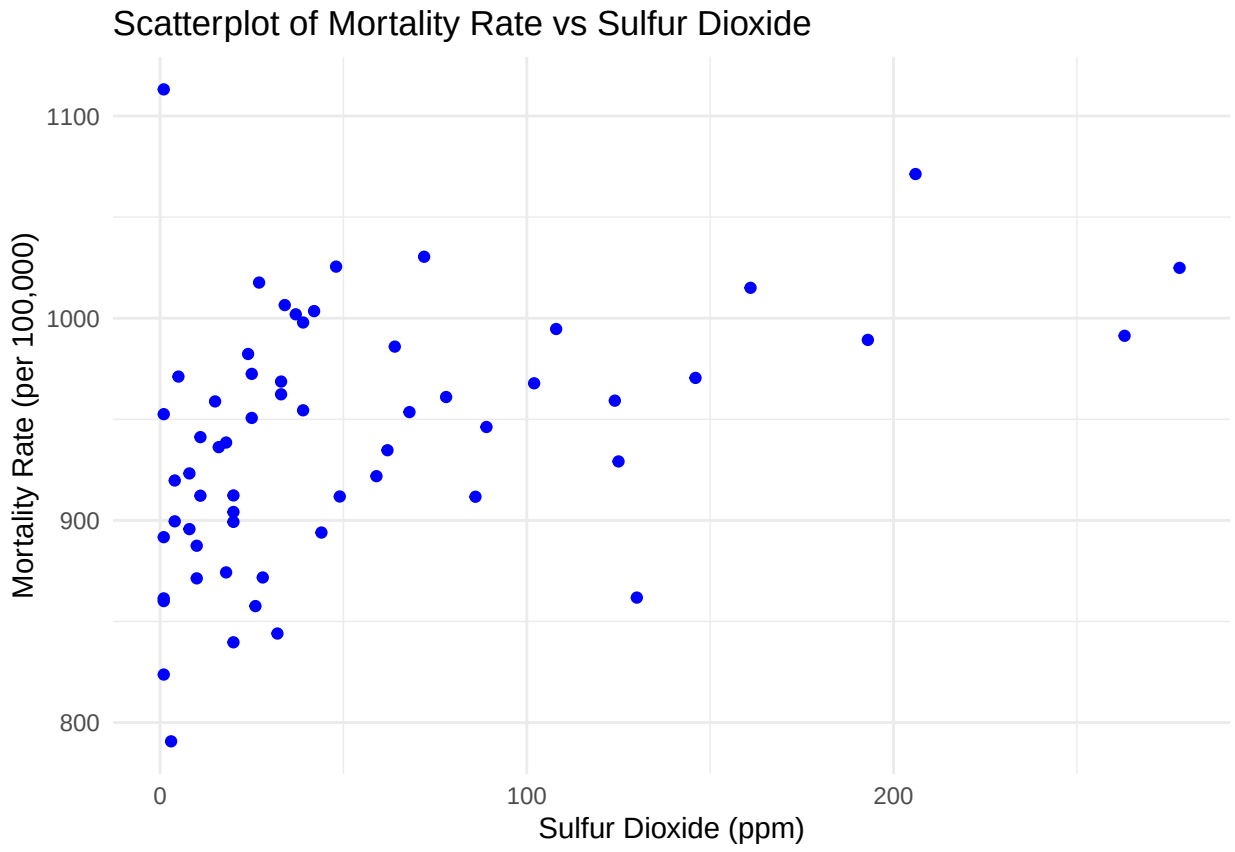
Interpret the slope coefficient from the model you chose in (b)

Intercept (Estimate = 904.724): This is the predicted mortality rate when the log of nitric oxides (log_nox) is zero. Mathematically, this means when nitric oxides is 1, the predicted mortality rate is about 904.724 deaths per 100,000 people.

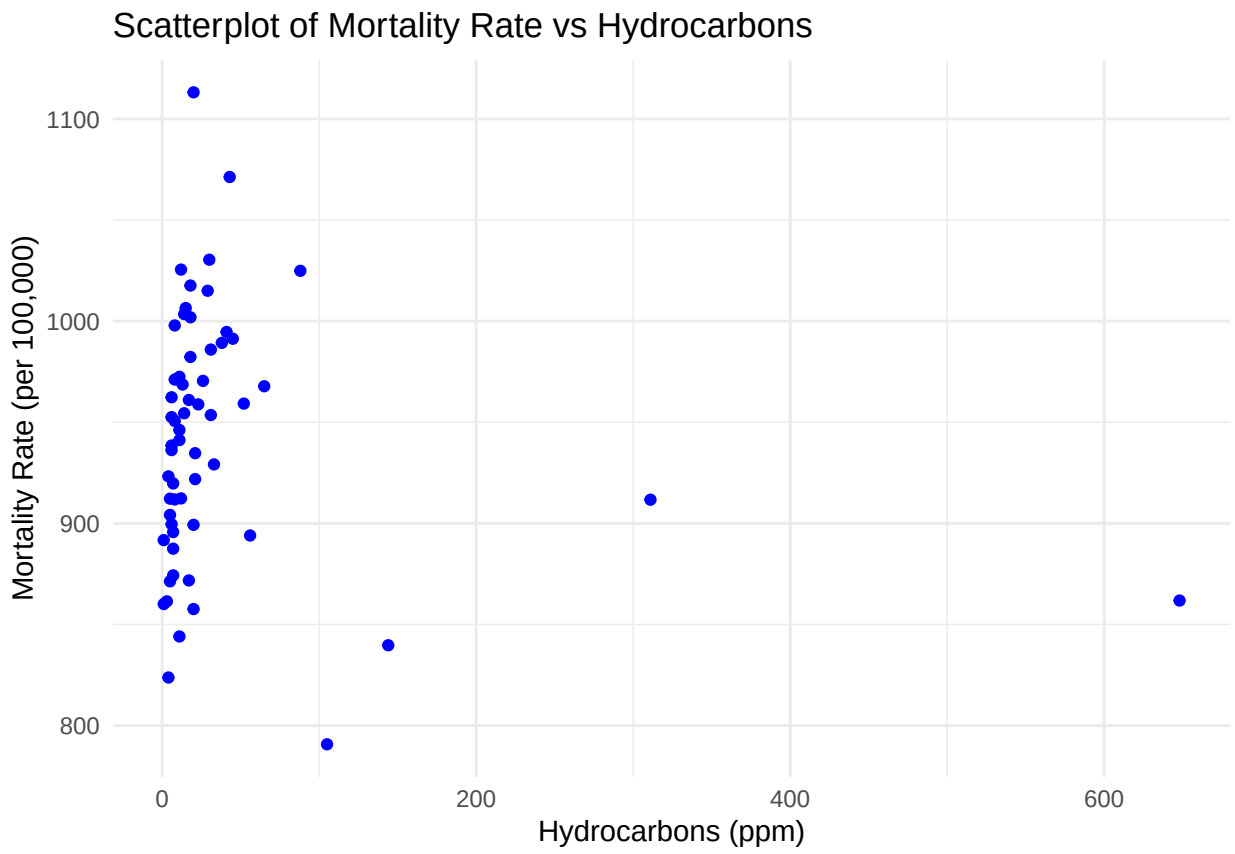
(d)

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.

```
ggplot(pollution_data, aes(x = so2, y = mort)) +
  geom_point(color = 'blue') +
  labs(x = "Sulfur Dioxide (ppm)", y = "Mortality Rate (per 100,000)") +
  ggtitle("Scatterplot of Mortality Rate vs Sulfur Dioxide") +
  theme_minimal()
```




```
ggplot(pollution_data, aes(x = hc, y = mort)) +
  geom_point(color = 'blue') +
  labs(x = "Hydrocarbons (ppm)", y = "Mortality Rate (per 100,000)") +
  ggtitle("Scatterplot of Mortality Rate vs Hydrocarbons") +
  theme_minimal()
```



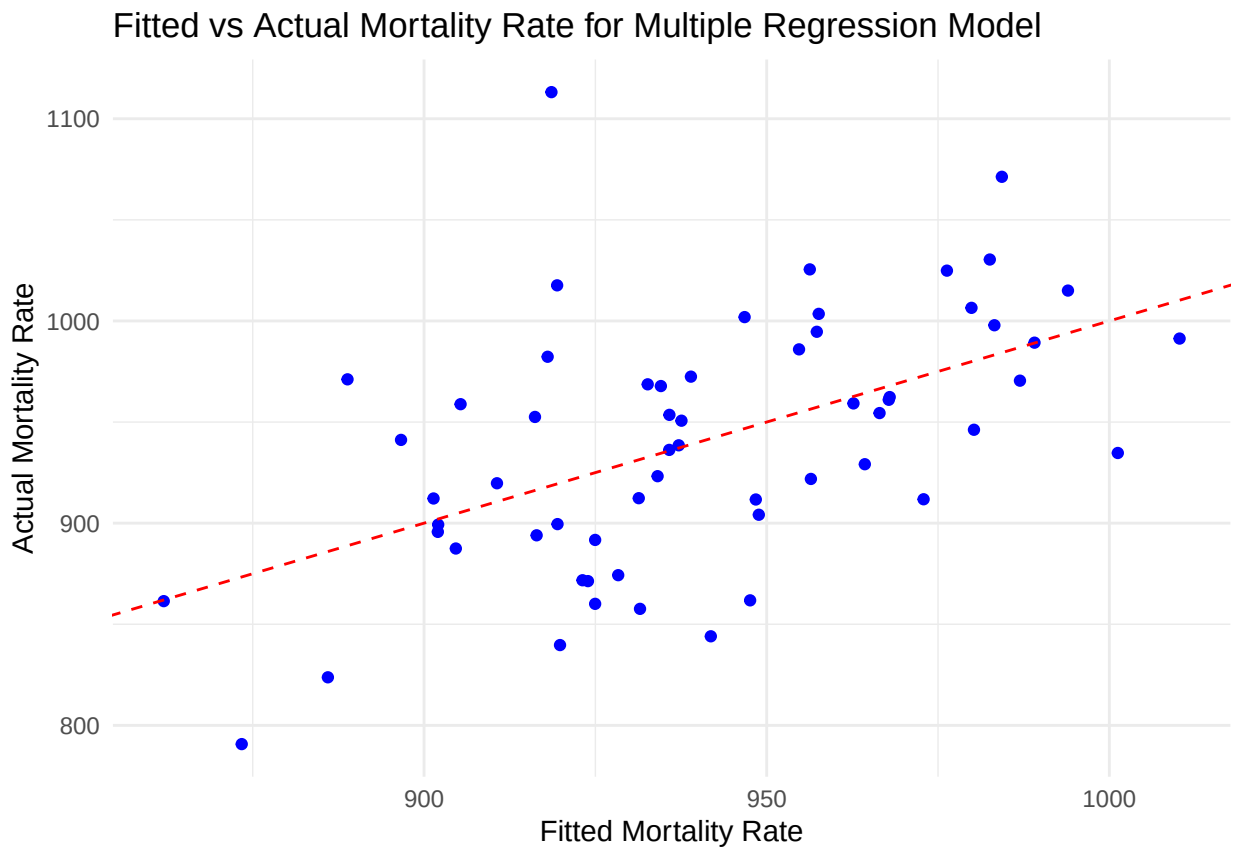
We can defer from two plots above that so2 and hc needs to be transformed.

```
pollution_data$log_so2 <- log(pollution_data$so2)  # log transform sulfur dioxide
pollution_data$log_hc <- log(pollution_data$hc)    # log transform hydrocarbons

# Fit multiple regression model
model_multiple <- lm(mort ~ log_nox + log_so2 + log_hc, data = pollution_data)

# Summary of the model
summary(model_multiple)
```

```
##
## Call:
## lm(formula = mort ~ log_nox + log_so2 + log_hc, data = pollution_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -97.793 -34.728  -3.118  34.148 194.567
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  924.965     21.449  43.125  < 2e-16 ***
## log_nox       58.336     21.751   2.682  0.00960 **
## log_so2       11.762      7.165   1.642  0.10629
## log_hc       -57.300     19.419  -2.951  0.00462 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.36 on 56 degrees of freedom
## Multiple R-squared:  0.2752, Adjusted R-squared:  0.2363
## F-statistic: 7.086 on 3 and 56 DF,  p-value: 0.0004044
```

```r
# Create a plot of fitted vs actual values
ggplot(pollution_data, aes(x = fitted(model_multiple), y = mort)) +
  geom_point(color = 'blue') +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(x = "Fitted Mortality Rate", y = "Actual Mortality Rate") +
  ggtitle("Fitted vs Actual Mortality Rate for Multiple Regression Model") +
  theme_minimal()
```



Residuals: The residuals show how well the model fits the data. The spread from the min to the max shows that some observations are not perfectly predicted. A few residuals are large, particularly the maximum residual (194.567), indicating some points where the model overpredicts significantly.

Coefficients:

(a) Intercept (924.965): The estimated mortality rate when all three log-transformed predictors (log_nox, log_so2, and log_hc) are zero. This means in areas where levels of nitric oxides, sulfur dioxide, and hydrocarbons are very low, the model predicts a baseline mortality rate of around 924.965 per 100,000 people.

(b) log_nox (58.336): The coefficient for log_nox (nitric oxides) suggests that for a 1% increase in nitric oxide levels, the mortality rate is expected to increase by approximately 58.34 deaths per 100,000, holding the other pollutants constant. This effect is statistically significant with a p-value of 0.0096, meaning there's strong evidence that higher levels of nitric oxides are associated with higher mortality.

(c) log_so2 (11.762): The coefficient for log_so2 (sulfur dioxide) suggests that for a 1% increase in sulfur dioxide levels, the mortality rate is expected to increase by about 11.76 deaths per 100,000, holding the other pollutants constant. However, the p-value is 0.106, which is not statistically significant at the 5% level. This means there's insufficient evidence to claim a significant association between sulfur dioxide and mortality rates.

(d) log_hc (-57.300): The coefficient for log_hc (hydrocarbons) is -57.30, indicating that for a 1% increase in hydrocarbon levels, the mortality rate is expected to decrease by about 57.30 deaths per 100,000, holding the other pollutants constant. This is statistically significant with a p-value of 0.00462, meaning there is strong evidence that hydrocarbons are negatively associated with mortality rates.

(e) R-squared

The adjusted R-squared value of 0.2363 accounts for the number of predictors in the model. Since this value is close to the R-squared value, it indicates that the number of predictors included is not drastically overfitting the data.

(f) F-statistic (7.086, p-value = 0.0004044):

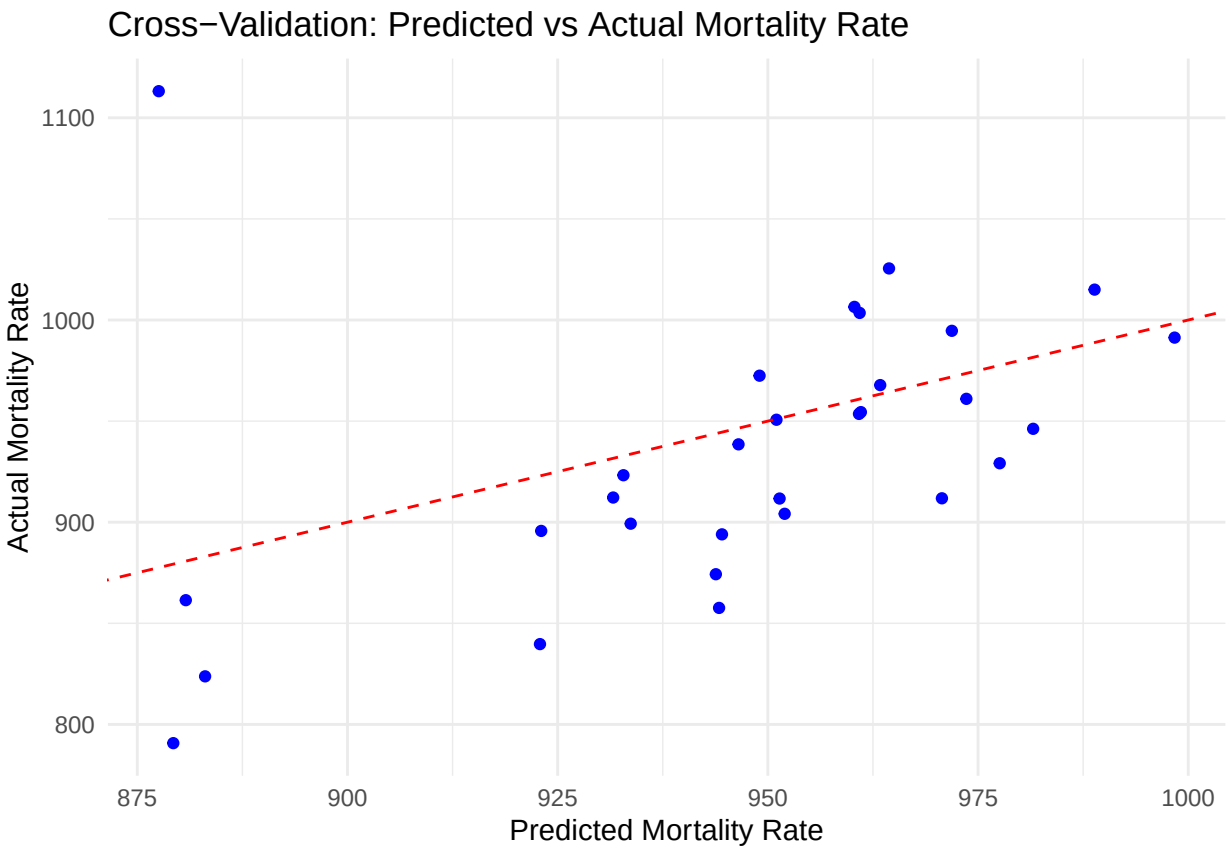
The F-statistic tests the null hypothesis that all coefficients are zero (i.e., none of the predictors have an effect on mortality rates). The p-value (0.0004044) indicates that the overall model is statistically significant, meaning that at least one of the predictors is significantly associated with mortality rates.‘

(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.

```
set.seed(1)

n <- nrow(pollution_data)

half <- floor(n / 2)
train_data <- pollution_data[1:half, ]
test_data <- pollution_data[(half + 1):n, ]

model_cv <- lm(mort ~ log_nox + log_so2 + log_hc, data = train_data)

predictions <- predict(model_cv, newdata = test_data)

comparison <- data.frame(
  Actual = test_data$mort,
  Predicted = predictions
)

ggplot(comparison, aes(x = Predicted, y = Actual)) +
  geom_point(color = 'blue') +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
```

```
  labs(x = "Predicted Mortality Rate", y = "Actual Mortality Rate") +
  ggtitle("Cross-Validation: Predicted vs Actual Mortality Rate") +
  theme_minimal()
```




```
mse <- mean((comparison$Actual - comparison$Predicted)^2)
mse
```

```
## [1] 3739.941
```

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.

(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.

```r
library(rstanarm)
```

```
## Loading required package: Rcpp
```

```
## This is rstanarm version 2.32.1
```

```
## - 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())
```

```r
earnings <- read.csv("earnings.csv")
earnings$earnk <- earnings$earn/1000
fit <- stan_glm(earnk ~ height + male, data=earnings, refresh = 0)
print(fit)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      earnk ~ height + male
##  observations: 1816
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept) -26.0   11.7
## height        0.6    0.2
## male         10.6    1.5
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 21.4    0.4
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```r
earnings$'log(earnk)' <- log(earnings$earn)
logmodel <- stan_glm(log(earnk) ~ height + male, data=earnings, subset=earn>0, refresh = 0)
print(logmodel)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      log(earnk) ~ height + male
##  observations: 1629
##  predictors:   3
##  subset:       earn > 0
## ------
##             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
```

(b)

Compare models from other exercises in this chapter.

```
mesquite <- read.csv("mesquite_cleaned.csv")
head(mesquite)
```

```
##   obs group diam1 diam2 total_height canopy_height density weight
## 1   1   MCD   1.8  1.15         1.30          1.00       1  401.3
## 2   2   MCD   1.7  1.35         1.35          1.33       1  513.7
## 3   3   MCD   2.8  2.55         2.16          0.60       1 1179.2
## 4   4   MCD   1.3  0.85         1.80          1.20       1  308.0
## 5   5   MCD   3.3  1.90         1.55          1.05       1  855.2
## 6   6   MCD   1.4  1.40         1.20          1.00       1  268.7
```

```
fit_2 <- stan_glm(weight ~ canopy_height, data=mesquite, refresh = 0)
print(fit_2)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      weight ~ canopy_height
##  observations: 46
##  predictors:   2
## ------
##               Median MAD_SD
## (Intercept)   -730.0  220.5
## canopy_height 1162.2  185.6
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 475.9   50.4
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
logmodel_2 <- stan_glm(log(weight) ~ canopy_height, data=mesquite, refresh = 0)
print(logmodel_2)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      log(weight) ~ canopy_height
##  observations: 46
##  predictors:   2
## ------
##               Median MAD_SD
## (Intercept)   4.2    0.3
## canopy_height 1.6    0.3
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.7    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

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. $log(W) = -2.39+2log(H)$ RSE=0.02 ### (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?
0.989 ## 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 D_i and R_i. 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: This is straightforward and easy to interpret. A positive value means the Democratic candidate raised more money, while a negative value means the Republican candidate raised more.

Disadvantage:The absolute difference might miss the relative scale of fundraising.

(b)

The ratio, D_i/R_i Advantage: Ratios automatically adjust for scale.

Disadvantage: Ratios can become highly skewed when one candidate raises significantly more money than the other, which can be problematic in linear models.

(c)

The difference on the logarithmic scale, $\log D_i - \log R_i$
Advantage: It reflects the proportional difference between the candidates' fundraising. It accounts for relative differences while maintaining the interpretability of differences.

Disadvantage: If D_i or R_i is zero, the log difference becomes undefined. ### (d) The relative proportion, $D_i/(D_i + R_i)$. Advantage: Proportions are often used in models predicting probabilities or shares (e.g., logistic regression), making this a natural candidate for vote share predictions.

Disadvantage: Changes in the proportion become less meaningful as values approach 0 or 1, as the scale becomes compressed. This can lead to loss of sensitivity in extreme cases.

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, $\log Q = \alpha + \beta \log P$. Suppose the estimate for β is 0.3. Interpret this coefficient.

The coefficient β=0.3 means that for a 1% increase in the price of cigarettes, the quantity purchased increases by 0.3%.

In other words, there is a positive elasticity between price and quantity purchased, which is somewhat counterintuitive in the context of typical demand models.

This could imply that in this particular sample or context:

Higher prices are associated with higher quantities sold, which could happen for several reasons, such as specific market dynamics, increased demand in higher-income areas, or other confounding factors. In general, β=0.3 means that the relationship between price and quantity purchased is such that a proportional change in price leads to a smaller proportional change in quantity purchased (with a factor of 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.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(broom)


data <- read.csv("beauty.csv")

# Model 1: Basic Linear Model
model_1 <- lm(eval ~ beauty + female + age + minority + nonenglish + lower, data = data)
summary(model_1)
```

```
##
## Call:
## lm(formula = eval ~ beauty + female + age + minority + nonenglish +
##     lower, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.84713 -0.35266  0.04673  0.38961  1.05248
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.194984   0.145899  28.753  < 2e-16 ***
## beauty       0.140213   0.032858   4.267 2.41e-05 ***
## female      -0.197257   0.052730  -3.741 0.000207 ***
## age         -0.002238   0.002756  -0.812 0.417182
## minority    -0.070909   0.076930  -0.922 0.357154
## nonenglish  -0.274246   0.110484  -2.482 0.013415 *
## lower        0.098401   0.054097   1.819 0.069570 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5311 on 456 degrees of freedom
## Multiple R-squared:  0.09575,    Adjusted R-squared:  0.08385
## F-statistic: 8.047 on 6 and 456 DF,  p-value: 2.836e-08
```

```
# Model 2: Adding interaction term (Beauty * Female)
model_2 <- lm(eval ~ beauty * female + age + minority + nonenglish + lower, data = data)
summary(model_2)
```

```
##
## Call:
## lm(formula = eval ~ beauty * female + age + minority + nonenglish +
##     lower, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.81944 -0.34492  0.04647  0.40890  1.08254
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.180026   0.145957  28.639  < 2e-16 ***
```

```
## beauty         0.189308   0.045031   4.204 3.16e-05 ***
## female        -0.203999   0.052812  -3.863 0.000128 ***
## age           -0.001766   0.002767  -0.638 0.523605
## minority      -0.044161   0.078619  -0.562 0.574588
## nonenglish    -0.292610   0.110901  -2.638 0.008613 **
## lower          0.092356   0.054139   1.706 0.088711 .
## beauty:female -0.103650   0.065132  -1.591 0.112216
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5302 on 455 degrees of freedom
## Multiple R-squared:  0.1008, Adjusted R-squared:  0.08692
## F-statistic: 7.283 on 7 and 455 DF,  p-value: 2.74e-08
```

```
# Model 3: Log transformation of age
data$log_age <- log(data$age + 1)
model_3 <- lm(eval ~ beauty * female + log_age + minority + nonenglish + lower, data = data)
summary(model_3)
```

```
##
## Call:
## lm(formula = eval ~ beauty * female + log_age + minority + nonenglish +
##     lower, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.82147 -0.34628  0.05241  0.40317  1.08197
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.33863    0.51798   8.376 6.85e-16 ***
## beauty         0.19139    0.04495   4.258 2.51e-05 ***
## female        -0.20166    0.05269  -3.828 0.000148 ***
## log_age       -0.06327    0.13162  -0.481 0.630938
## minority      -0.04231    0.07851  -0.539 0.590249
## nonenglish    -0.29312    0.11092  -2.643 0.008507 **
## lower          0.09352    0.05408   1.729 0.084416 .
## beauty:female -0.10519    0.06505  -1.617 0.106556
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5303 on 455 degrees of freedom
## Multiple R-squared:  0.1004, Adjusted R-squared:  0.08656
## F-statistic: 7.255 on 7 and 455 DF,  p-value: 2.967e-08
```

```
AIC(model_1, model_2, model_3)
```

```
##         df      AIC
## model_1  8 736.8960
## model_2  9 736.3261
## model_3  9 736.5054
```

The best is model2 for it has the least AIC.

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.

```
library(rstanarm)
mesquite <- read_csv("mesquite_cleaned.csv")
```

```
## Rows: 46 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): group
## dbl (7): obs, diam1, diam2, total_height, canopy_height, density, weight
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
mesquite$canopy_volume <- mesquite$diam1 * mesquite$diam2 * mesquite$canopy_height
mesquite$canopy_area <- mesquite$diam1 * mesquite$diam2
mesquite$canopy_shape <- mesquite$diam1 / mesquite$diam2
fit_4 <- stan_glm(formula = log(weight) ~ log(canopy_volume) + log(canopy_area) +
log(canopy_shape) + log(total_height) + log(density) + group, data=mesquite,refresh = 0)
summary(fit_4)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      log(weight) ~ log(canopy_volume) + log(canopy_area) + log(canopy_shape) +
##     log(total_height) + log(density) + group
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 46
##  predictors:   7
##
## Estimates:
##                       mean   sd   10%   50%   90%
## (Intercept)          4.8    0.2  4.6   4.8   5.0
## log(canopy_volume)   0.4    0.3  0.0   0.4   0.7
## log(canopy_area)     0.4    0.3  0.0   0.4   0.8
## log(canopy_shape)   -0.4    0.2 -0.7  -0.4  -0.1
## log(total_height)    0.4    0.3  0.0   0.4   0.8
## log(density)         0.1    0.1  0.0   0.1   0.3
## groupMCD             0.6    0.1  0.4   0.6   0.7
## sigma                0.3    0.0  0.3   0.3   0.4
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 5.9    0.1  5.8   5.9   6.0
##
```

```
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                     mcse Rhat n_eff
## (Intercept)         0.0  1.0  3674
## log(canopy_volume) 0.0  1.0  1423
## log(canopy_area)   0.0  1.0  1541
## log(canopy_shape)  0.0  1.0  2961
## log(total_height)  0.0  1.0  1870
## log(density)       0.0  1.0  2638
## groupMCD           0.0  1.0  2268
## sigma              0.0  1.0  2751
## mean_PPD           0.0  1.0  3628
## log-posterior      0.1  1.0  1485
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```