MA678 Homework 2

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.

The intercept is significant with a mean of 1.3 and a 95% CI of [0.5, 2.1]. X1 is significant with a mean of 0.5 and a 95% CI of [0.5, 0.6]. X2 is significant with a 95% Ci of [0.8,0.8]. Sigma is significant with a mean of 0.9 and a 95% CI of [0.7,1.1]. All of these do not contain 0 and have relatively small variability.

```
Pyth<-read.table("~/Downloads/pyth.txt", header = TRUE)
pyth_subset<-Pyth[1:40, ]
model11.5a<-rstanarm::stan_glm(y~x1+x2, data = pyth_subset)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000118 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.18 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.079 seconds (Warm-up)
## Chain 1:                0.073 seconds (Sampling)
## Chain 1:                0.152 seconds (Total)
```

```
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.8e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.18 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.083 seconds (Warm-up)
## Chain 2:                0.071 seconds (Sampling)
## Chain 2:                0.154 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 1.9e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.19 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.096 seconds (Warm-up)
## Chain 3:                0.075 seconds (Sampling)
## Chain 3:                0.171 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
```

```
## Chain 4: Gradient evaluation took 1.9e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.19 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.082 seconds (Warm-up)
## Chain 4:                0.072 seconds (Sampling)
## Chain 4:                0.154 seconds (Total)
## Chain 4:
```

```
summary(model11.5a)
```

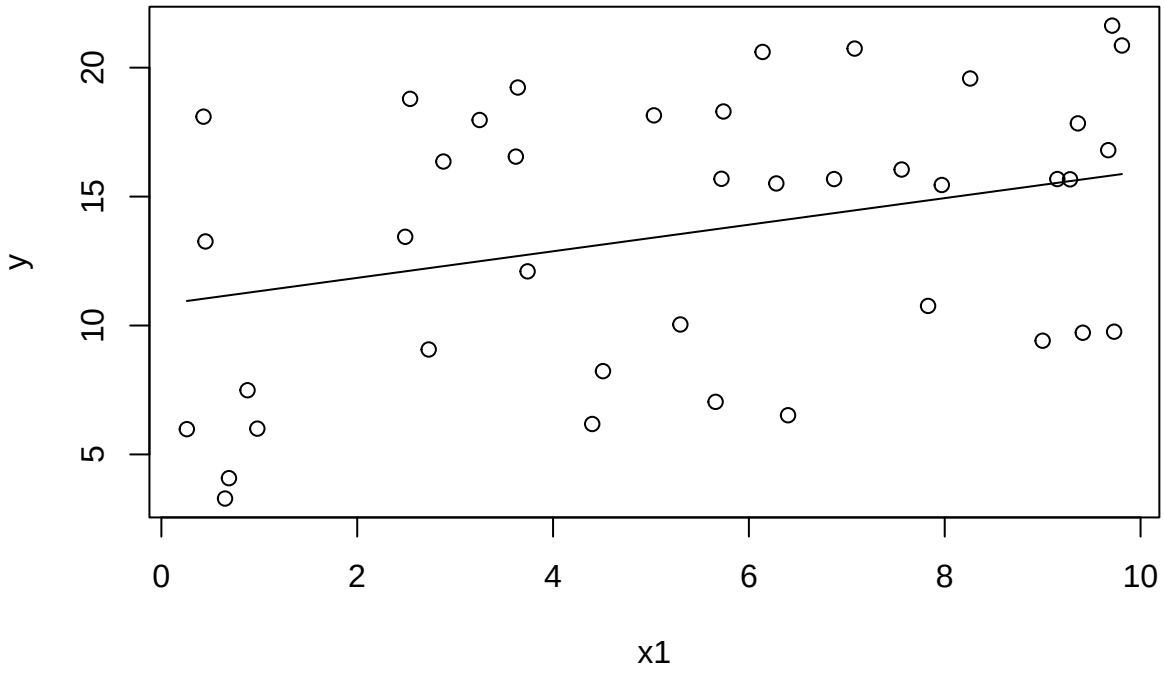
```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x1 + x2
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 40
##  predictors:   3
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) 1.3    0.4  0.8   1.3   1.8
## x1          0.5    0.0  0.5   0.5   0.6
## x2          0.8    0.0  0.8   0.8   0.8
## sigma       0.9    0.1  0.8   0.9   1.1
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 13.6    0.2 13.3  13.6  13.9
##
## 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  4450
## x1            0.0  1.0  4530
## x2            0.0  1.0  4250
```

```
## sigma         0.0  1.0  3400
## mean_PPD      0.0  1.0  4299
## log-posterior 0.0  1.0  1747
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

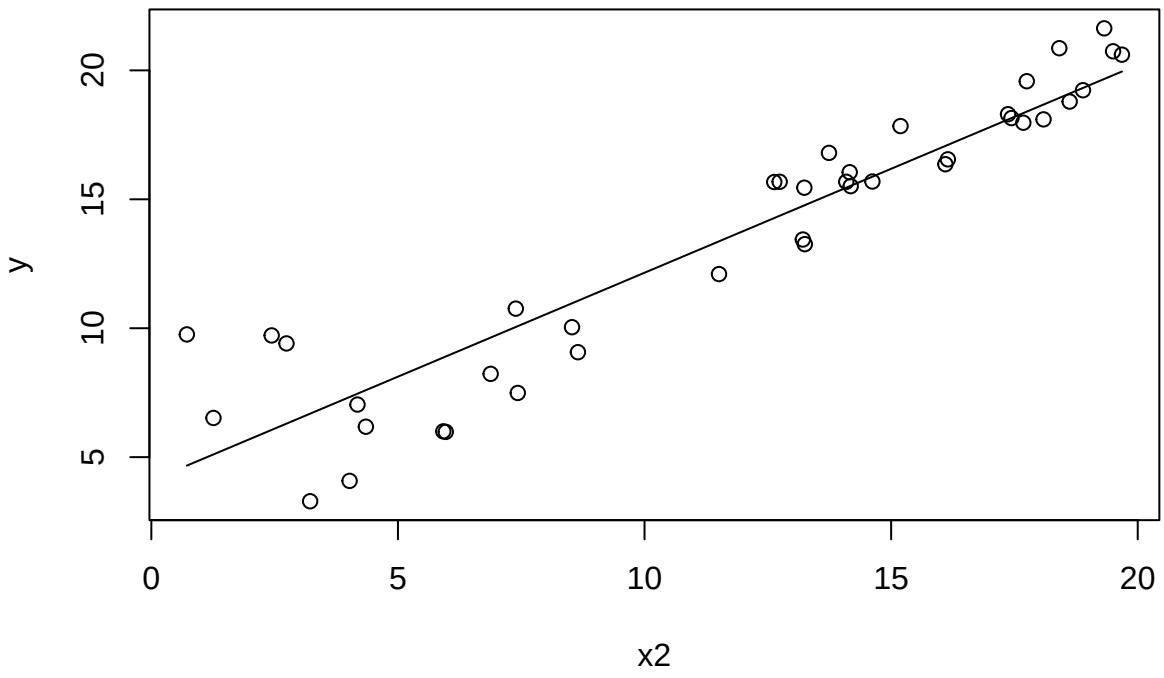
(b)

Display the estimated model graphically as in Figure 10.2

```
plot ( pyth_subset$x1, pyth_subset$y, xlab = "x1", ylab="y")
curve (coef(model11.5a)[1] + coef(model11.5a)[2]*x+ coef(model11.5a)[3]*mean(pyth_subset$x2), add =TRUE)
```



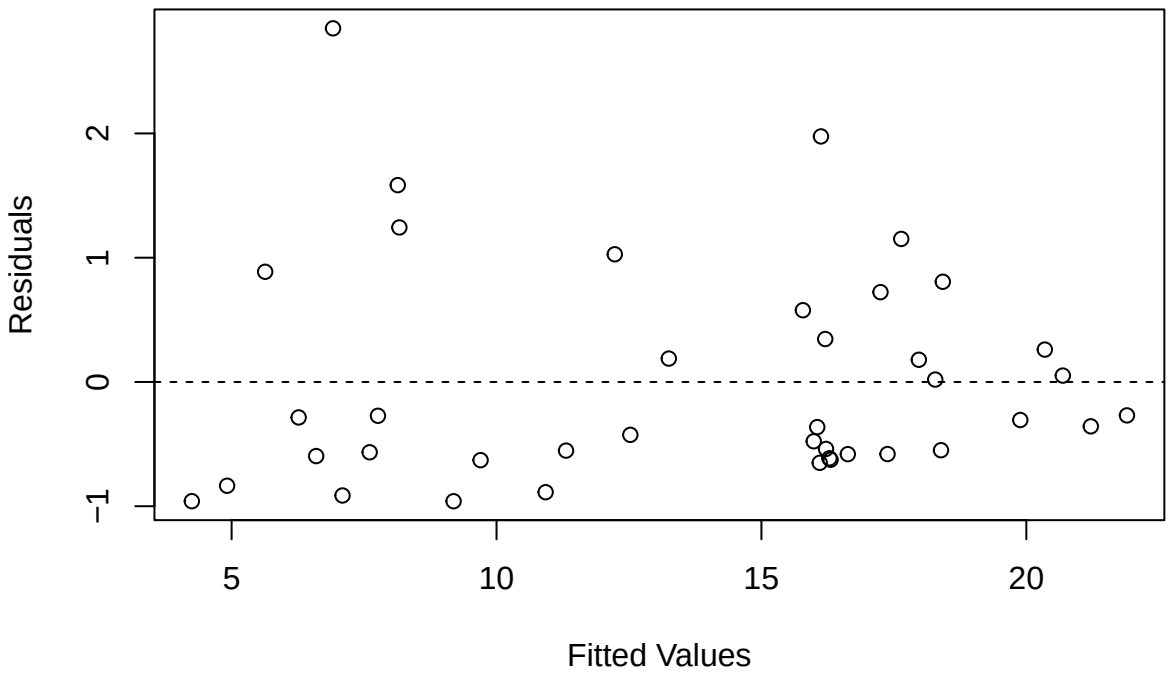

```
plot (pyth_subset$x2, pyth_subset$y, xlab = "x2", ylab="y")
curve (coef(model11.5a)[1] + coef(model11.5a)[2]*mean(pyth_subset$x1)+ coef(model11.5a)[3]*x, add=TRUE)
```



(c)

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

```
plot(fitted(model11.5a), residuals(model11.5a),
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(h = 0, lty = 2)
```



(d)

Make predictions for the remaining 20 data points in the file. How confident do you feel about these predictions? By using the standard errors, to create 95% CI for each predicted point, I feel 95% confident that each true value lies within the interval.

```
predictions<-predict(model11.5a, newdata = Pyth[41:60,], se.fit = TRUE)
predict11.5d<-data.frame(Pyth[41:60, ], Predicted = predictions$fit,
                                  Lower_95CI = predictions$fit - 2 * predictions$se.fit,
                                  Upper_95CI = predictions$fit + 2 * predictions$se.fit)
```

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 _____ and _____ of their predicted values from the regression. [0.78, 1.28]

(b)

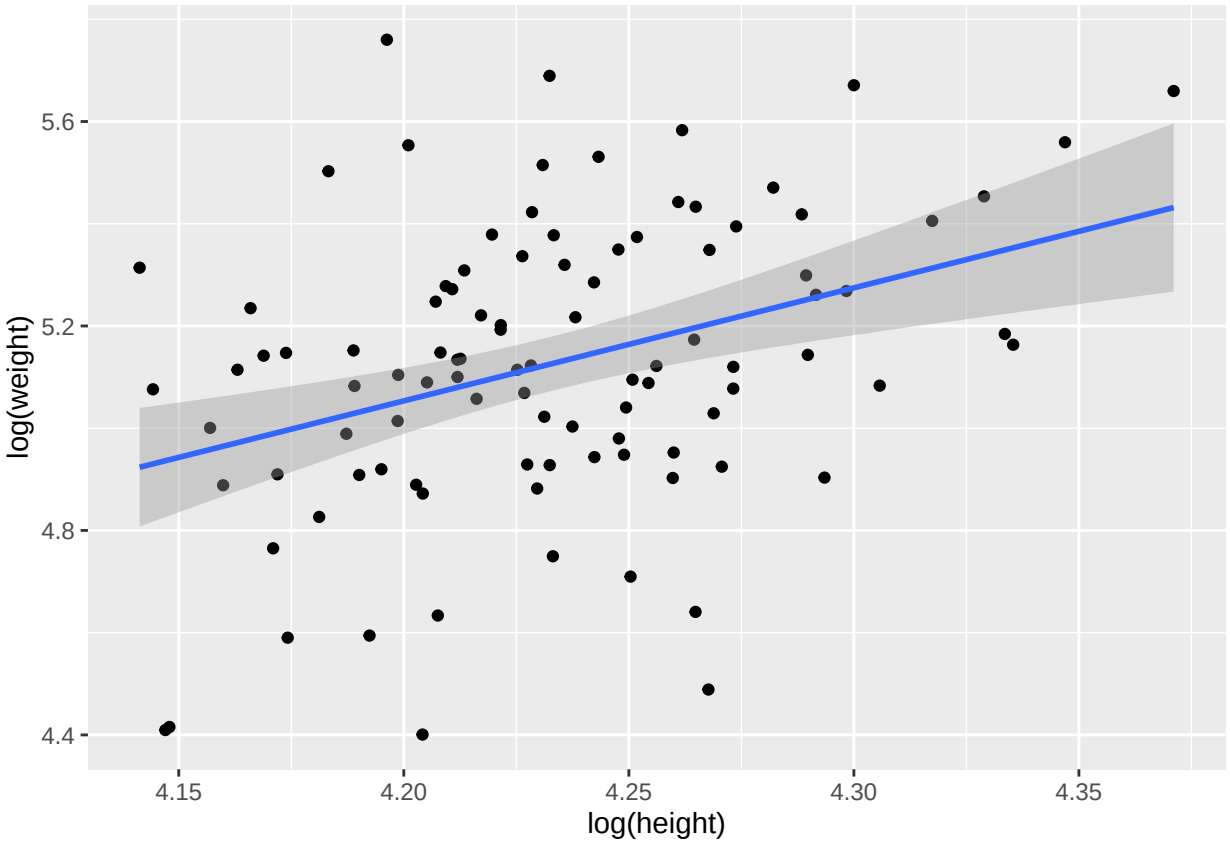
Using pen and paper, sketch the regression line and scatterplot of log(weight) versus log(height) that make sense and are consistent with the fitted model. Be sure to label the axes of your graph.

```
height=rnorm(n = 100, mean = 69, sd = 3)
weight=exp(-3.8+2.1*log(height)+rnorm(n = 100, mean = 0, sd = 0.25))

weight_height=data.frame(weight=weight, height=height)

ggplot(weight_height, aes(x=log(height), y=log(weight)))+
geom_point()+
geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



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.

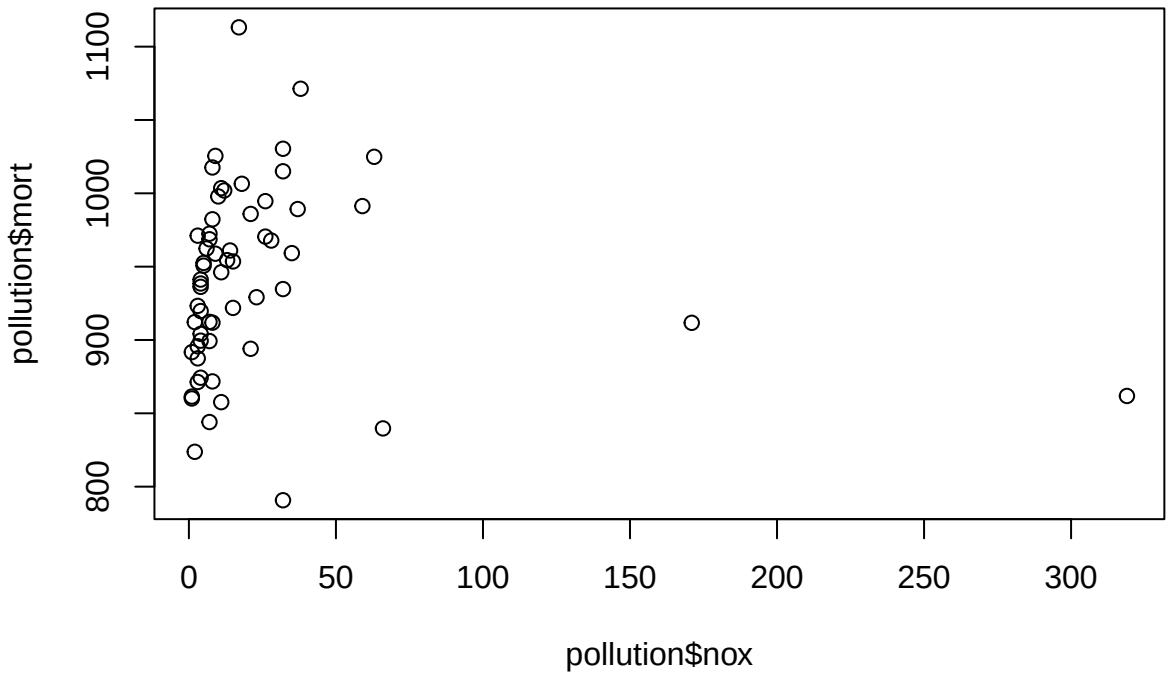
(a)

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. The linear model does not fit the data well due to several outliers and the y intercept does not make sense because there is never a day with 0 mortalities.

```
pollution<-read.csv("~/Downloads/pollution.csv")
head(pollution)
```

```
##   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
```

```
plot(x = pollution$nox, y= pollution$mort)
```



```r
model12.6a<-rstanarm::stan_glm(mort~nox, data = pollution)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 2.6e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.26 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.148 seconds (Warm-up)
## Chain 1:                0.096 seconds (Sampling)
## Chain 1:                0.244 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.9e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.19 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.132 seconds (Warm-up)
## Chain 2:                0.083 seconds (Sampling)
## Chain 2:                0.215 seconds (Total)
## Chain 2:
##
```

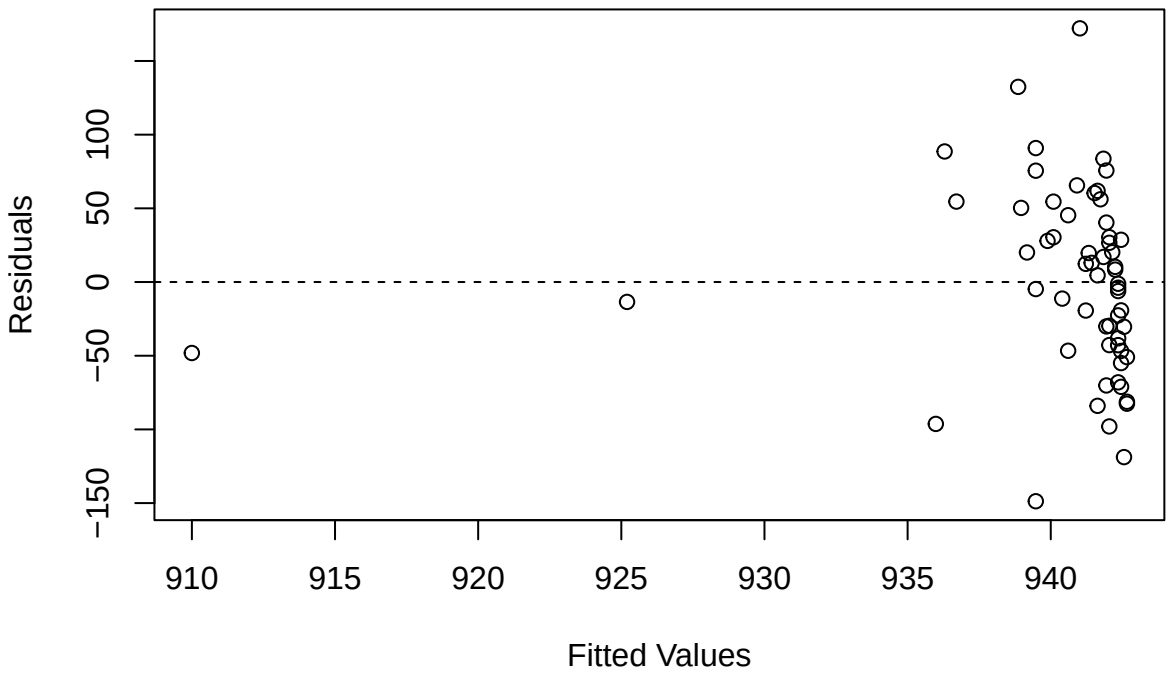
```
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 2.4e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.24 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.228 seconds (Warm-up)
## Chain 3:                0.089 seconds (Sampling)
## Chain 3:                0.317 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 3.5e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.35 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.204 seconds (Warm-up)
## Chain 4:                0.083 seconds (Sampling)
## Chain 4:                0.287 seconds (Total)
## Chain 4:
```

```
summary(model12.6a)
```

```
##
## Model Info:
```

```
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      mort ~ nox
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 60
##  predictors:   2
##
## Estimates:
##               mean   sd    10%   50%   90%
## (Intercept) 942.8    9.0 931.3 942.8 954.4
## nox          -0.1    0.2  -0.3  -0.1   0.1
## sigma        63.2    5.8  56.2  62.8  70.5
##
## Fit Diagnostics:
##            mean   sd    10%   50%   90%
## mean_PPD 940.8   11.4 926.3 940.7 955.7
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.2  1.0  3465
## nox           0.0  1.0  3283
## sigma         0.1  1.0  3102
## mean_PPD      0.2  1.0  3750
## log-posterior 0.0  1.0  1784
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```r
plot(fitted(model12.6a), residuals(model12.6a),
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(h = 0, lty = 2)
```



(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. The residual plot is much more compact than the original model, but there is still not equal variance of the residuals.

```
summary(pollution)
```

```
##       prec            jant            jult          ovr65
##  Min.   :10.00   Min.   :12.00   Min.   :63.00   Min.   : 5.600
##  1st Qu.:32.75   1st Qu.:27.00   1st Qu.:72.00   1st Qu.: 7.675
##  Median :38.00   Median :31.50   Median :74.00   Median : 9.000
##  Mean   :37.37   Mean   :33.98   Mean   :74.58   Mean   : 8.798
##  3rd Qu.:43.25   3rd Qu.:40.00   3rd Qu.:77.25   3rd Qu.: 9.700
##  Max.   :60.00   Max.   :67.00   Max.   :85.00   Max.   :11.800
##       popn            educ            hous            dens           nonw
##  Min.   :2.920   Min.   : 9.00   Min.   :66.80   Min.   :1441   Min.   : 0.80
##  1st Qu.:3.210   1st Qu.:10.40   1st Qu.:78.38   1st Qu.:3104   1st Qu.: 4.95
##  Median :3.265   Median :11.05   Median :81.15   Median :3567   Median :10.40
##  Mean   :3.263   Mean   :10.97   Mean   :80.91   Mean   :3876   Mean   :11.87
##  3rd Qu.:3.360   3rd Qu.:11.50   3rd Qu.:83.60   3rd Qu.:4520   3rd Qu.:15.65
##  Max.   :3.530   Max.   :12.30   Max.   :90.70   Max.   :9699   Max.   :38.50
##      wwdrk            poor             hc              nox
##  Min.   :33.80   Min.   : 9.40   Min.   :  1.00   Min.   :  1.00
##  1st Qu.:43.25   1st Qu.:12.00   1st Qu.:  7.00   1st Qu.:  4.00
```

```
##  Median :45.50   Median :13.20   Median : 14.50   Median :  9.00
##  Mean   :46.08   Mean   :14.37   Mean   : 37.85   Mean   : 22.65
##  3rd Qu.:49.52   3rd Qu.:15.15   3rd Qu.: 30.25   3rd Qu.: 23.75
##  Max.   :59.70   Max.   :26.40   Max.   :648.00   Max.   :319.00
##       so2             humid           mort
##  Min.   :  1.00   Min.   :38.00   Min.   : 790.7
##  1st Qu.: 11.00   1st Qu.:55.00   1st Qu.: 898.4
##  Median : 30.00   Median :57.00   Median : 943.7
##  Mean   : 53.77   Mean   :57.67   Mean   : 940.4
##  3rd Qu.: 69.00   3rd Qu.:60.00   3rd Qu.: 983.2
##  Max.   :278.00   Max.   :73.00   Max.   :1113.2
```

```
pollution$mort_centered<-pollution$mort-mean(pollution$mort)
model12.6b<-rstanarm::stan_glm(mort~log(nox), data = pollution)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 2.7e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.27 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.379 seconds (Warm-up)
## Chain 1:                0.094 seconds (Sampling)
## Chain 1:                0.473 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 8e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.8 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
```

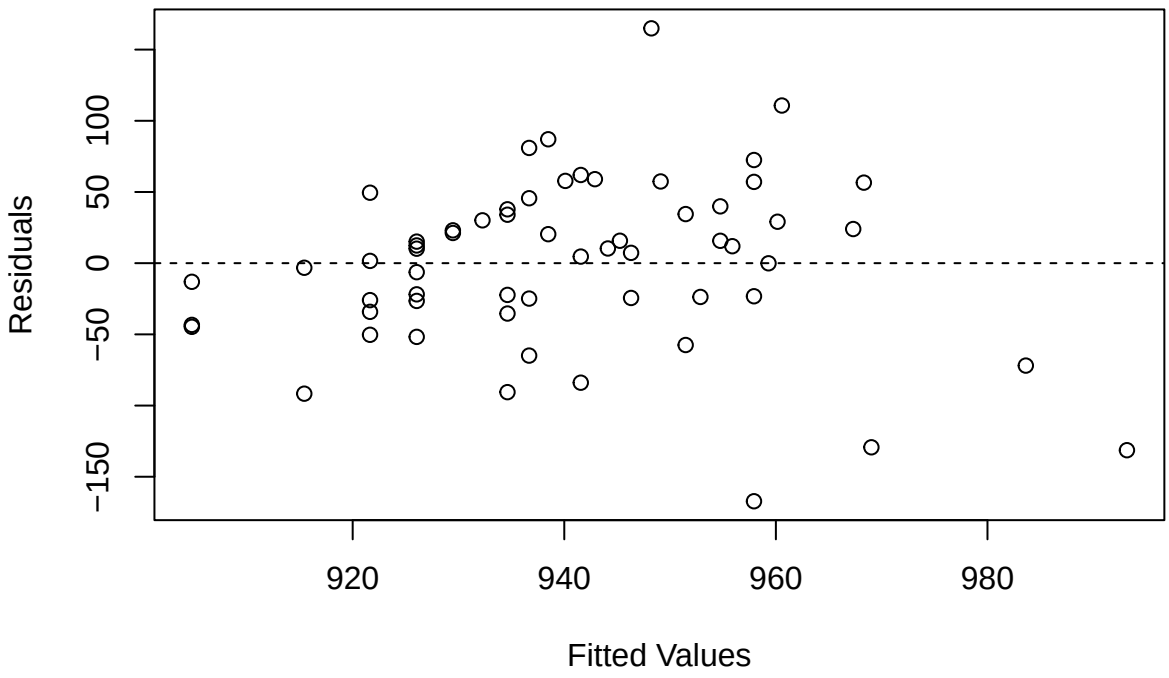
```
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.225 seconds (Warm-up)
## Chain 2:                0.095 seconds (Sampling)
## Chain 2:                0.32 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 2e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.202 seconds (Warm-up)
## Chain 3:                0.148 seconds (Sampling)
## Chain 3:                0.35 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 2e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
```

```
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.176 seconds (Warm-up)
## Chain 4:                0.099 seconds (Sampling)
## Chain 4:                0.275 seconds (Total)
## Chain 4:
```

```
summary(model12.6b)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      mort ~ log(nox)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 60
##  predictors:   2
##
## Estimates:
##               mean   sd    10%   50%   90%
## (Intercept) 905.1   16.9 883.8 904.8 926.7
## log(nox)     15.2    6.6   7.0  15.3  23.4
## sigma        60.7    5.7  53.8  60.2  68.1
##
## Fit Diagnostics:
##            mean   sd    10%   50%   90%
## mean_PPD 940.5   11.4 925.9 940.4 955.1
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.3  1.0  4081
## log(nox)      0.1  1.0  4144
## sigma         0.1  1.0  3811
## mean_PPD      0.2  1.0  3866
## log-posterior 0.0  1.0  1847
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
plot(fitted(model12.6b), residuals(model12.6b),
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(h = 0, lty = 2)
```



(c)

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

For a 1% increase in nitrous oxide, mortality rates increase by 0.151 units or for each one unit increase in log(nox) the mortality rate increases by 15.1.

(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.

```
model12.6d<-rstanarm::stan_glm(mort~log(nox)+so2+hc, data = pollution)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 3e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.3 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
```

```
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.327 seconds (Warm-up)
## Chain 1:                0.175 seconds (Sampling)
## Chain 1:                0.502 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 3.7e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.37 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.276 seconds (Warm-up)
## Chain 2:                0.171 seconds (Sampling)
## Chain 2:                0.447 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 3.8e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.38 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
```

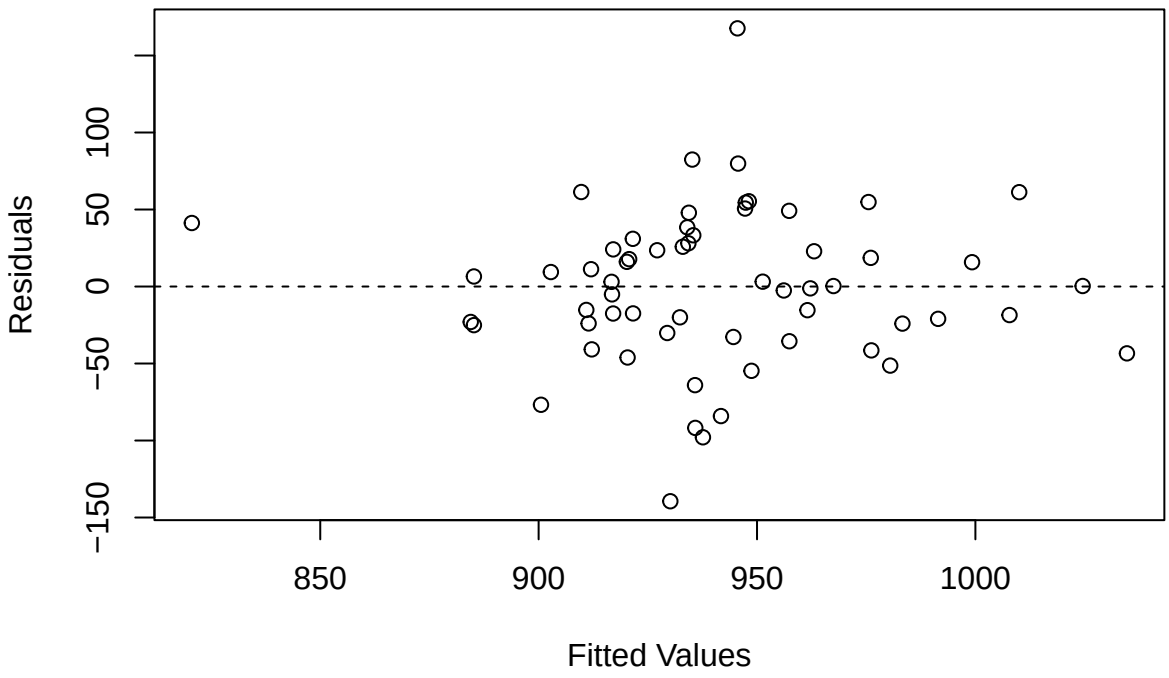
```
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.274 seconds (Warm-up)
## Chain 3:                0.162 seconds (Sampling)
## Chain 3:                0.436 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 2.8e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.28 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.398 seconds (Warm-up)
## Chain 4:                0.181 seconds (Sampling)
## Chain 4:                0.579 seconds (Total)
## Chain 4:
```

```
summary(model12.6d)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      mort ~ log(nox) + so2 + hc
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 60
##  predictors:   4
##
## Estimates:
##               mean   sd    10%   50%   90%
## (Intercept) 884.9   18.6 861.5 885.3 908.5
```

```
## log(nox)     23.7   10.5  10.4  23.7  37.2
## so2           0.3    0.2   0.1   0.3   0.5
## hc           -0.4    0.1  -0.5  -0.4  -0.2
## sigma        52.6    5.1  46.6  52.2  59.4
##
## Fit Diagnostics:
##            mean   sd    10%   50%   90%
## mean_PPD 940.3    9.7 928.0 940.2 952.7
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.4  1.0  2747
## log(nox)      0.2  1.0  2199
## so2           0.0  1.0  2478
## hc            0.0  1.0  2543
## sigma         0.1  1.0  2667
## mean_PPD      0.2  1.0  3221
## log-posterior 0.0  1.0  1287
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```r
plot(fitted(model12.6d), residuals(model12.6d),
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(h = 0, lty = 2)
```



(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.

```
half = dim(pollution)[1]/2
cv12.6e = rstanarm::stan_glm(mort~log(nox)+so2+hc, data = pollution, subset = 1:half)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 2.7e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.27 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
```

```
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.274 seconds (Warm-up)
## Chain 1:                0.155 seconds (Sampling)
## Chain 1:                0.429 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 2e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.242 seconds (Warm-up)
## Chain 2:                0.168 seconds (Sampling)
## Chain 2:                0.41 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 3.8e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.38 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
```

```
## Chain 3:
## Chain 3:  Elapsed Time: 0.353 seconds (Warm-up)
## Chain 3:                0.334 seconds (Sampling)
## Chain 3:                0.687 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 2.8e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.28 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.321 seconds (Warm-up)
## Chain 4:                0.211 seconds (Sampling)
## Chain 4:                0.532 seconds (Total)
## Chain 4:
```

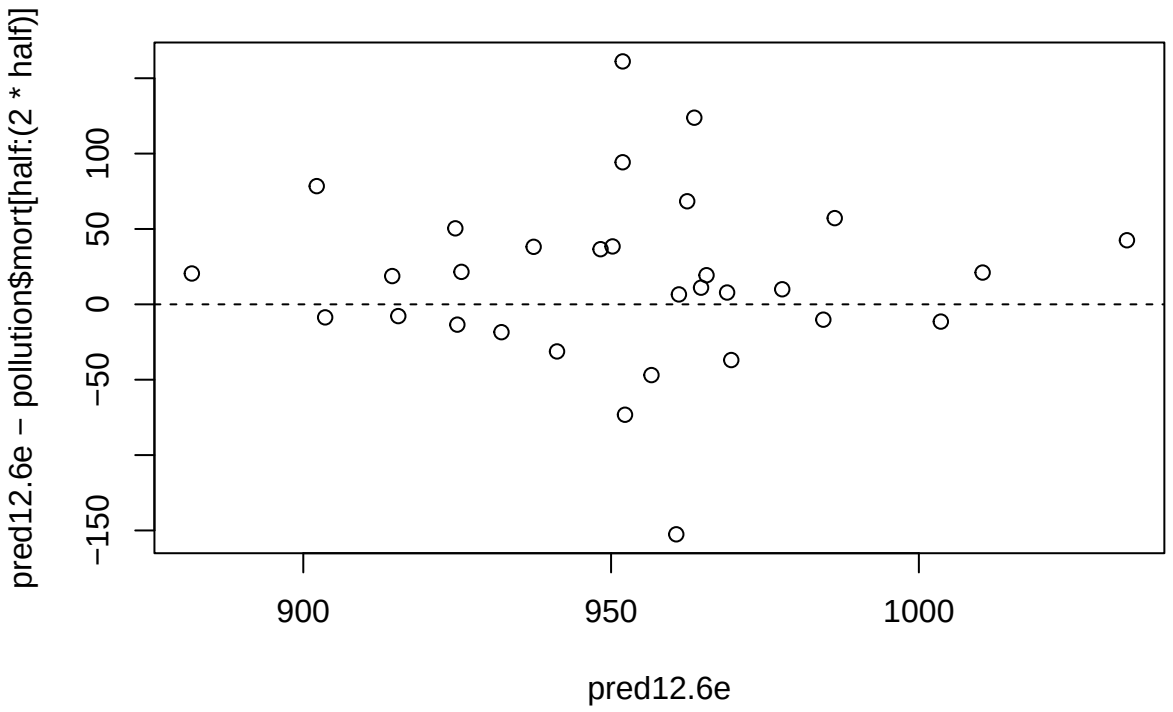
```
summary(cv12.6e)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      mort ~ log(nox) + so2 + hc
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 30
##  predictors:   4
##  subset:       1:half
##
## Estimates:
##               mean   sd    10%   50%   90%
## (Intercept) 882.7   22.3 853.9 883.0 910.7
## log(nox)     29.7   13.8  12.2  29.5  47.2
## so2           0.2    0.2  -0.1   0.2   0.4
## hc           -0.3    0.1  -0.4  -0.3  -0.2
## sigma        44.8    6.5  37.1  44.2  53.6
##
## Fit Diagnostics:
```

```
##            mean   sd    10%   50%   90%
## mean_PPD 946.6   11.7 931.9 946.5 961.5
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.5  1.0  2180
## log(nox)      0.3  1.0  1792
## so2           0.0  1.0  2153
## hc            0.0  1.0  2073
## sigma         0.1  1.0  2255
## mean_PPD      0.2  1.0  3495
## log-posterior 0.0  1.0  1389
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```r
pred12.6e=predict(cv12.6e, newdata = pollution[half:(2*half),])

plot(x = pred12.6e, y = pred12.6e- pollution$mort[half:(2*half)])
abline(h = 0, lty = 2)
```



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.

```
earnings <-read.csv("~/Downloads/earnings.csv")
model12.7a=rstanarm::stan_glm(earnk~height+male, data = earnings)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 2.7e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.27 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.152 seconds (Warm-up)
## Chain 1:                0.308 seconds (Sampling)
## Chain 1:                0.46 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 2.1e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.21 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
```

```
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.104 seconds (Warm-up)
## Chain 2:                0.259 seconds (Sampling)
## Chain 2:                0.363 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 2e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.155 seconds (Warm-up)
## Chain 3:                0.285 seconds (Sampling)
## Chain 3:                0.44 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 2.5e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.25 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.13 seconds (Warm-up)
## Chain 4:                0.344 seconds (Sampling)
## Chain 4:                0.474 seconds (Total)
## Chain 4:
```

```
model12.7a2=stan_glm(log(1+earnk)~height+male, data = earnings)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 2.7e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.27 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.098 seconds (Warm-up)
## Chain 1:                0.253 seconds (Sampling)
## Chain 1:                0.351 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 2.7e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.27 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
```

```
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.086 seconds (Warm-up)
## Chain 2:                0.243 seconds (Sampling)
## Chain 2:                0.329 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 2e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.104 seconds (Warm-up)
## Chain 3:                0.293 seconds (Sampling)
## Chain 3:                0.397 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 2.7e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.27 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.118 seconds (Warm-up)
```

```
## Chain 4:                0.292 seconds (Sampling)
## Chain 4:                0.41 seconds (Total)
## Chain 4:
```

```
summary(model12.7a)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      earnk ~ height + male
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1816
##  predictors:   3
##
## Estimates:
##               mean   sd    10%   50%   90%
## (Intercept) -26.0   11.6 -40.9 -25.9 -11.1
## height        0.6    0.2   0.4   0.6   0.9
## male         10.6    1.4   8.8  10.6  12.4
## sigma        21.4    0.4  21.0  21.4  21.9
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 21.2    0.7 20.2  21.2  22.1
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.3  1.0  2143
## height        0.0  1.0  2121
## male          0.0  1.0  2359
## sigma         0.0  1.0  3127
## mean_PPD      0.0  1.0  3567
## log-posterior 0.0  1.0  1932
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
summary(model12.7a2)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      log(1 + earnk) ~ height + male
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1816
##  predictors:   3
```

```
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept)  0.0    0.6 -0.8   0.0   0.8
## height       0.0    0.0  0.0   0.0   0.0
## male         0.6    0.1  0.5   0.6   0.7
## sigma        1.1    0.0  1.1   1.1   1.1
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 2.6    0.0  2.6   2.6   2.6
##
## 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  2823
## height        0.0  1.0  2767
## male          0.0  1.0  2626
## sigma         0.0  1.0  3307
## mean_PPD      0.0  1.0  3370
## log-posterior 0.0  1.0  1707
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
check12.7a1 = loo(model12.7a)
```

```
## Warning: Found 1 observation(s) with a pareto_k > 0.7. We recommend calling 'loo' again with argumen
```

```
check12.7a2 = loo(model12.7a2)
loo_compare(check12.7a1,check12.7a2)
```

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

```
##             elpd_diff se_diff
## model12.7a2     0.0       0.0
## model12.7a  -5408.3     169.3
```

(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(weight) = -5.52 + 0.02* log(height)+ error The residual standard deviation is 0.0476.

(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?
The R squared is 0.9448.

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, $D_i - R_i$ The advantages of D_i - R_i is that it would center the data, but its disadvantage is that it could mess up the scale and make the rsults harder to understand.

(b)

The ratio, D_i/R_i This can provide a good sense of scale between fundraising, but it fails if the republican candidate's fundraising is close to 0 or 0.

(c)

The difference on the logarithmic scale, $\log D_i - \log R_i$
Provides a good sense of proportion between campaigns, but its harder fopr the average person to interpret.

(d)

The relative proportion, $D_i/(D_i + R_i)$. The scaling by proportion makes it easy to understand, but it could mess up the understanding of scale.

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. For every 1% increase in the price of cigarettes there is a 0.3% increase in the quantity purchased on average.

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. I chose this to test the interaction between both being female and age as well beauty rating and being female. The other models I tried had most of the coefficients had CI's that included 0, so they were insignificant.

```
beauty<-read.csv("~/Downloads/beauty.csv")
model12.13<-rstanarm::stan_glm(eval~age*female+beauty*female, data = beauty)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 3e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.3 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.355 seconds (Warm-up)
## Chain 1:                0.383 seconds (Sampling)
## Chain 1:                0.738 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 2.1e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.21 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.441 seconds (Warm-up)
## Chain 2:                0.352 seconds (Sampling)
## Chain 2:                0.793 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 2e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.337 seconds (Warm-up)
## Chain 3:                0.437 seconds (Sampling)
## Chain 3:                0.774 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 1.9e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.19 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
```

```
## Chain 4:  Elapsed Time: 0.344 seconds (Warm-up)
## Chain 4:                0.397 seconds (Sampling)
## Chain 4:                0.741 seconds (Total)
## Chain 4:
```

```
summary(model12.13)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      eval ~ age * female + beauty * female
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 463
##  predictors:   6
##
## Estimates:
##                 mean   sd   10%   50%   90%
## (Intercept)    3.9    0.2  3.7   3.9   4.2
## age            0.0    0.0  0.0   0.0   0.0
## female         0.5    0.3  0.1   0.5   0.8
## beauty         0.2    0.0  0.2   0.2   0.3
## age:female     0.0    0.0  0.0   0.0   0.0
## female:beauty -0.1    0.1 -0.2  -0.1  -0.1
## sigma          0.5    0.0  0.5   0.5   0.6
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 4.0    0.0  4.0   4.0   4.0
##
## 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  1871
## age           0.0  1.0  1849
## female        0.0  1.0  1511
## beauty        0.0  1.0  2120
## age:female    0.0  1.0  1512
## female:beauty 0.0  1.0  2085
## sigma         0.0  1.0  3269
## mean_PPD      0.0  1.0  3778
## log-posterior 0.0  1.0  1603
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

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.

```
trees<-read.table("~/Downloads/mesquite.dat", header = TRUE)
#fit_2 <- stan_glm(formula = log(weight) ~ log(diam1) + log(diam2) + log(canopy_height) + log(total_hei
#pred12.14= posterior_predict(fit_2, newdata = new_trees, fun = exp)
#mean12.14 = apply(pred12.14, MARGIN = 2, FUN = mean)
#trees_mean = mean(mean12.14)
#trees_sd = sd(mean12.14)
```