MA678 Homework 2

9/10/2020

11.5

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

(a)

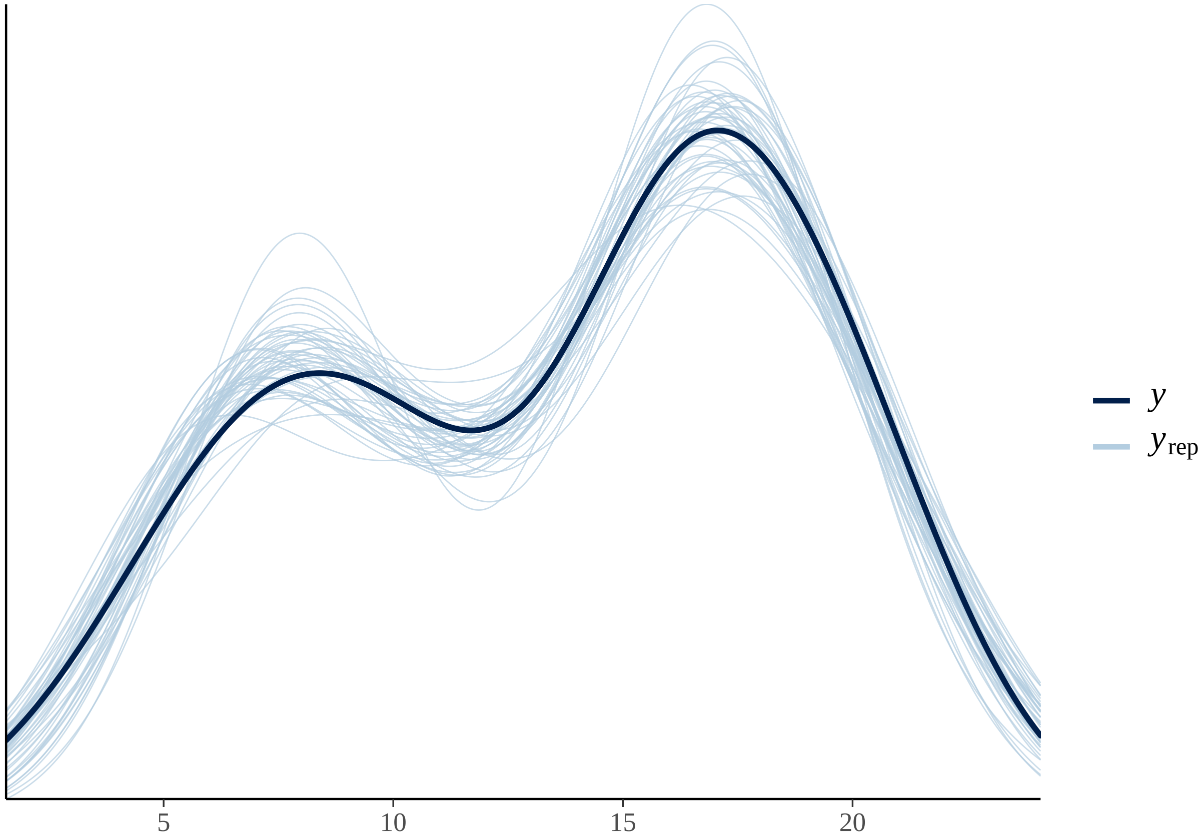
Use R to fit a linear regression model predicting y from x1, x2, using the first 40 data points in the file. Summarize the inferences and check the fit of your model.

```
#Pyth=read.table("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/Pyth/pyth.txt", header
ghv_data_dir <- "https://raw.githubusercontent.com/avehtari/ROS-Examples/master/"
Pyth <- read.table (paste0(ghv_data_dir,"Pyth/pyth.txt"), header=T)
fit_11.5alm=lm(formula = y~x1+x2, data = Pyth, subset = 1:40)
fit_11.5a=stan_glm(formula = y~x1+x2, data = Pyth, subset = 1:40, refresh=0)
summary(fit_11.5a)
```

```
##
## 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  4564
## x1            0.0  1.0  4417
## x2            0.0  1.0  3598
## sigma         0.0  1.0  3301
## mean_PPD      0.0  1.0  3600
## log-posterior 0.0  1.0  1705
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
# coef(fit_11.5a)
```

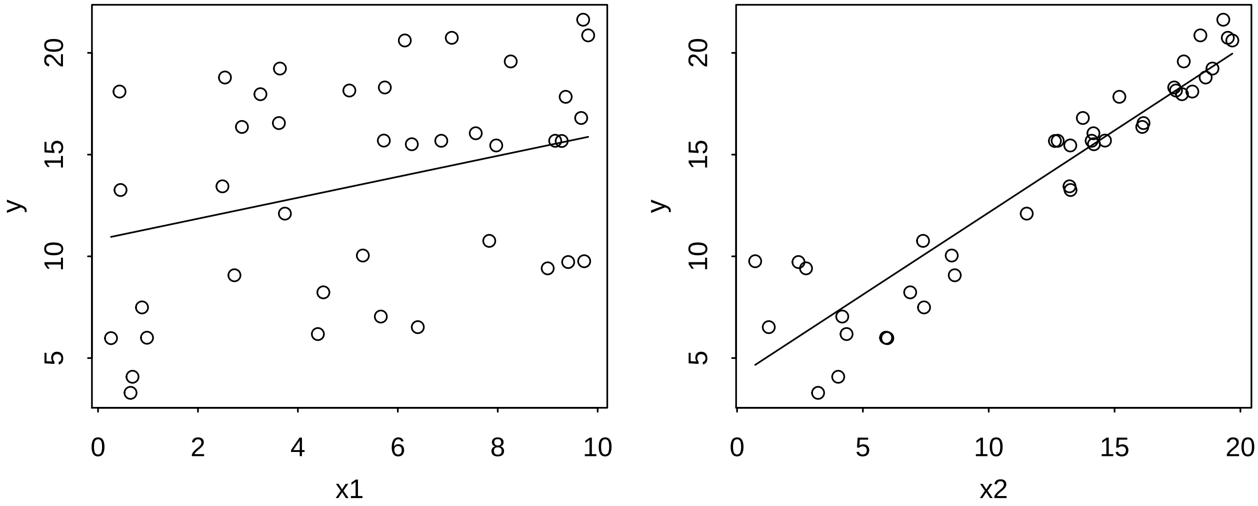
```
## posterior predictive check
pp_check(fit_11.5a)
```



(b)

Display the estimated model graphically as in Figure 10.2

```
par (mar=c(3,3,2,1), mgp=c(2,.7,0), tck=-.01)
par(mfrow=c(1,2))
plot ( Pyth[1:40,]$x1,  Pyth[1:40,]$y, xlab = "x1", ylab="y")
curve (coef(fit_11.5alm)[1] + coef(fit_11.5alm)[2]*x+ coef(fit_11.5alm)[3]*mean(Pyth[1:40,]$x2), add=TRU
#abline(lm(y~x1,data=Pyth[1:40,]),lty=2)
plot ( Pyth[1:40,]$x2,  Pyth[1:40,]$y, xlab = "x2", ylab="y")
curve (coef(fit_11.5alm)[1] + coef(fit_11.5alm)[2]*mean(Pyth[1:40,]$x1)+ coef(fit_11.5alm)[3]*x, add=TRU
```

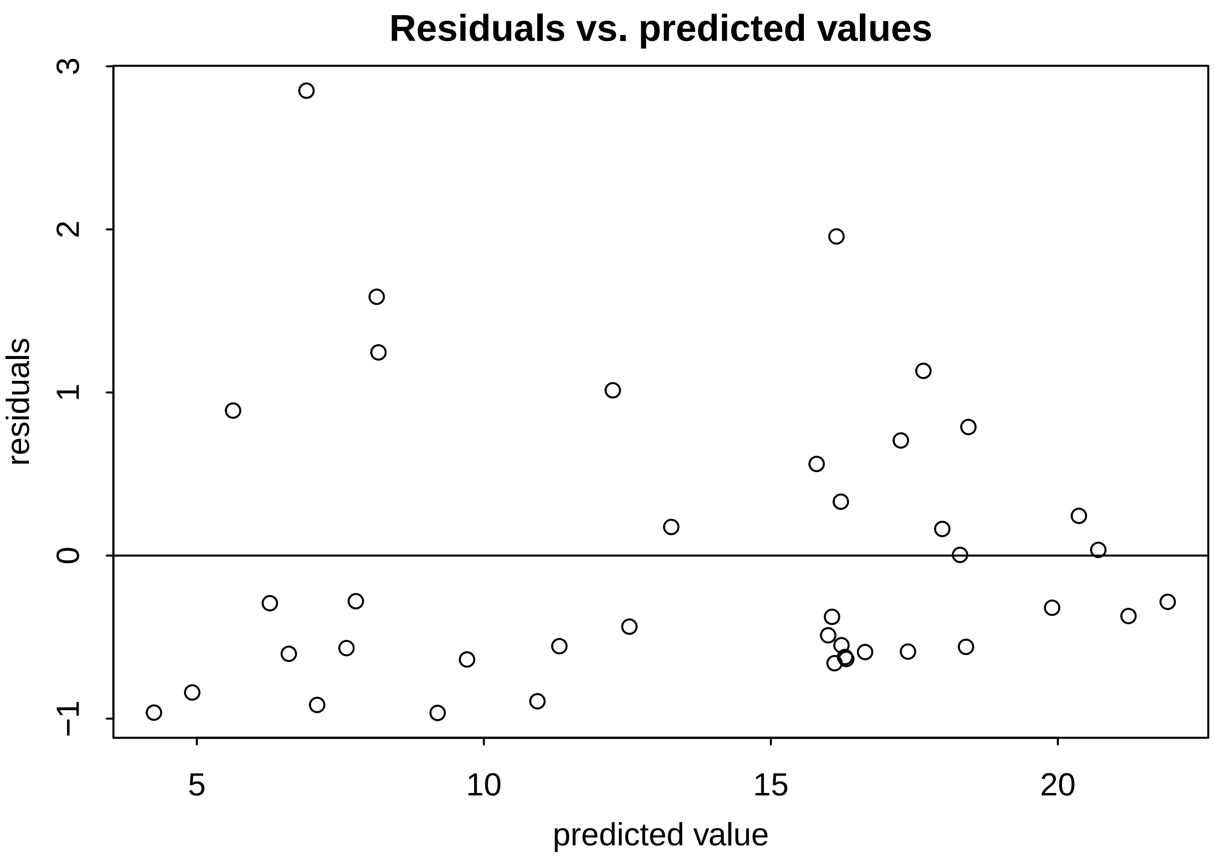


```
#abline(lm(y~x2,data=Pyth[1:40,]),lty=2)
```

(c)

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

```
## residual plot
predicted_11.5a <- predict(fit_11.5a)
resid_11.5a <- Pyth$y[1:40] - predicted_11.5a
par(mar=c(3,3,2,1), mgp=c(2,.7,0), tck=-.01)
plot(x=predicted_11.5a, y=resid_11.5a, type = "p", xlab = "predicted value", ylab = "residuals",
     main = "Residuals vs. predicted values")
abline(h=0)
```

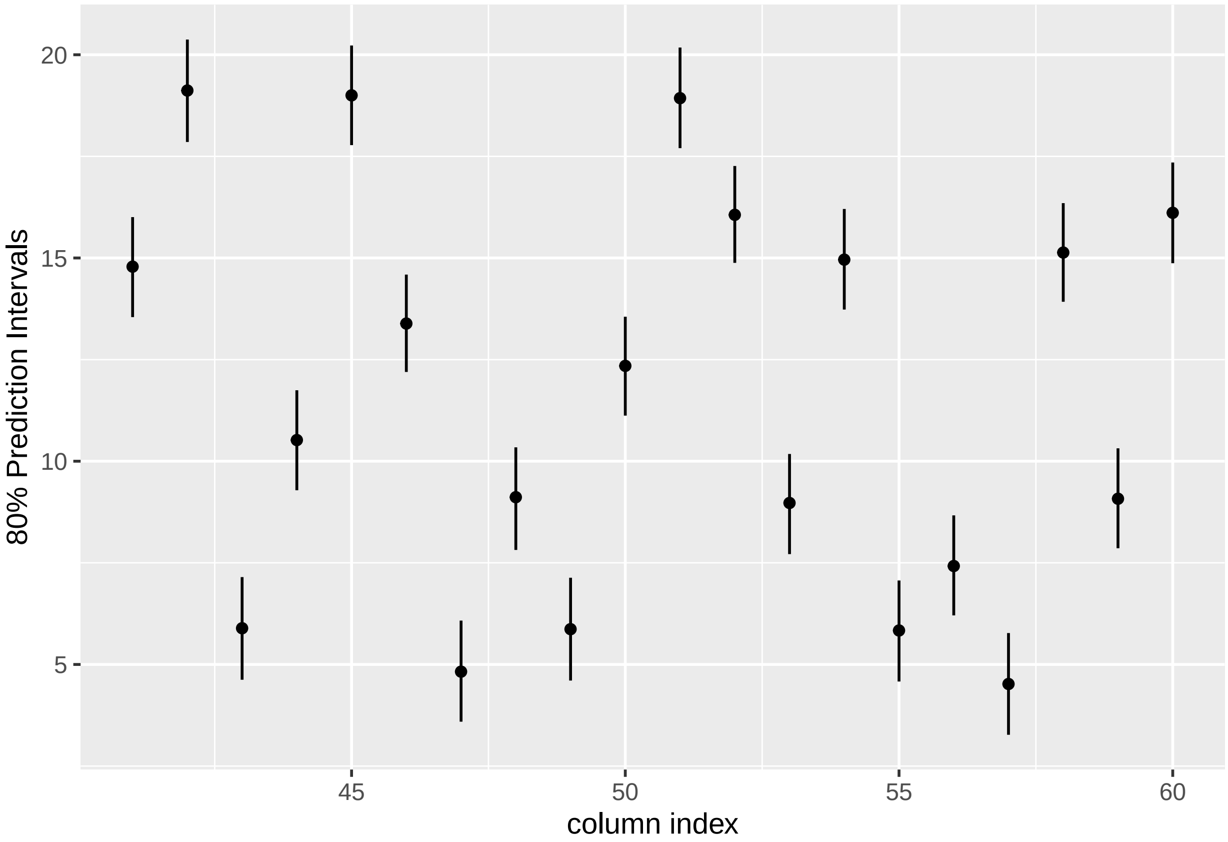


The equal variance of error assumption does not seem to be met

(d)

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

```
pred_fit_11.5a=posterior_predict(fit_11.5a, newdata = Pyth[41:60, 2:3])
pred_fit_11.5a_upper=apply(pred_fit_11.5a, MARGIN = 2, FUN = quantile, probs=0.9)
pred_fit_11.5a_lower=apply(pred_fit_11.5a, MARGIN = 2, FUN = quantile, probs=0.1)
pred_fit_11.5a_mean=apply(pred_fit_11.5a, MARGIN = 2, FUN = mean)
prediction=data.frame(upper=pred_fit_11.5a_upper, lower=pred_fit_11.5a_lower,
                      point_estimate=pred_fit_11.5a_mean, index=41:60)
ggplot(prediction)+
  geom_point(aes(x=index, y=point_estimate))+
  geom_segment(aes(x=index, xend=index, y=upper, yend=lower))+
  labs(x="column index", y="80% Prediction Intervals")
```

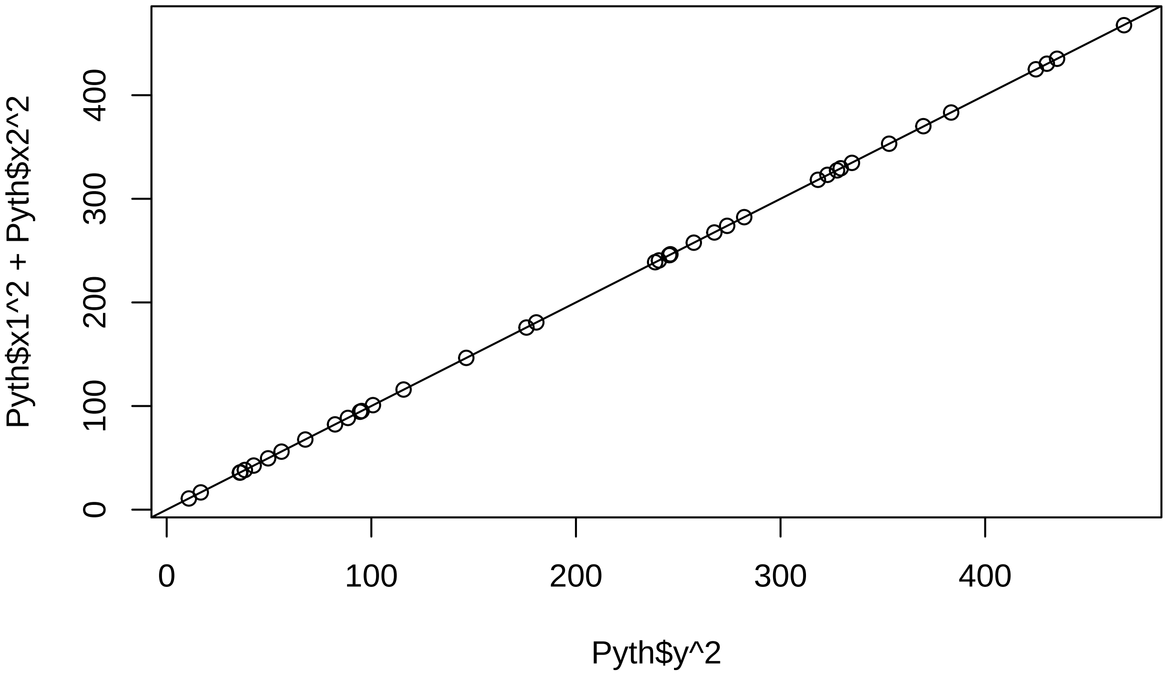


After doing this exercise, take a look at Gelman and Nolan (2002, section 9.4) to see where these data came from.

When you look at Gelman and Nolan, it turns out that this data was generated from a model

$$y^2 = x1^2 + x2^2$$

```r
plot(Pyth$y^2,Pyth$x1^2+Pyth$x2^2)
abline(0,1)
```



however, ys are calculated only up to the second decimal creating small error.

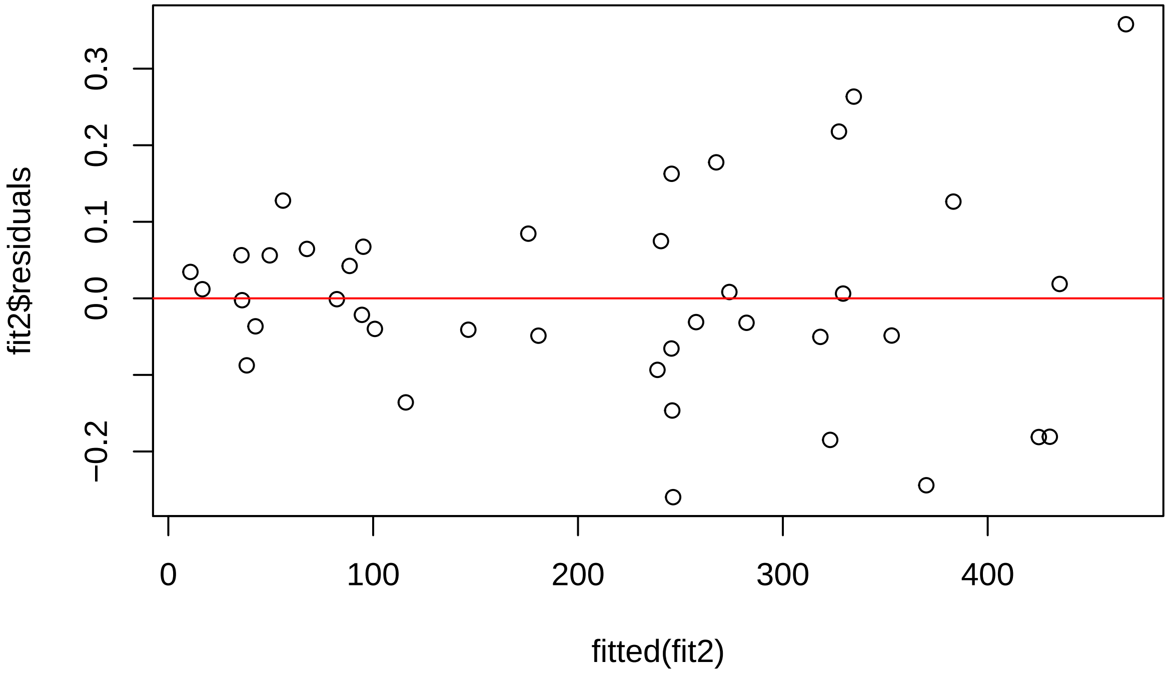
```
round(sqrt(Pyth[1:40,]$x1^2+Pyth[1:40,]$x2^2),2)-Pyth[1:40,]$y
```

```
##  [1]  0.00  0.01  0.00  0.00  0.01  0.00  0.01  0.00  0.00  0.00  0.01  0.00
## [13]  0.01  0.00  0.00  0.00 -0.01  0.00  0.00  0.00  0.00  0.00  0.00  0.01
## [25]  0.00  0.00  0.00  0.00  0.00  0.00 -0.01  0.00  0.01  0.00  0.01  0.00
## [37]  0.00 -0.01  0.00 -0.01
```

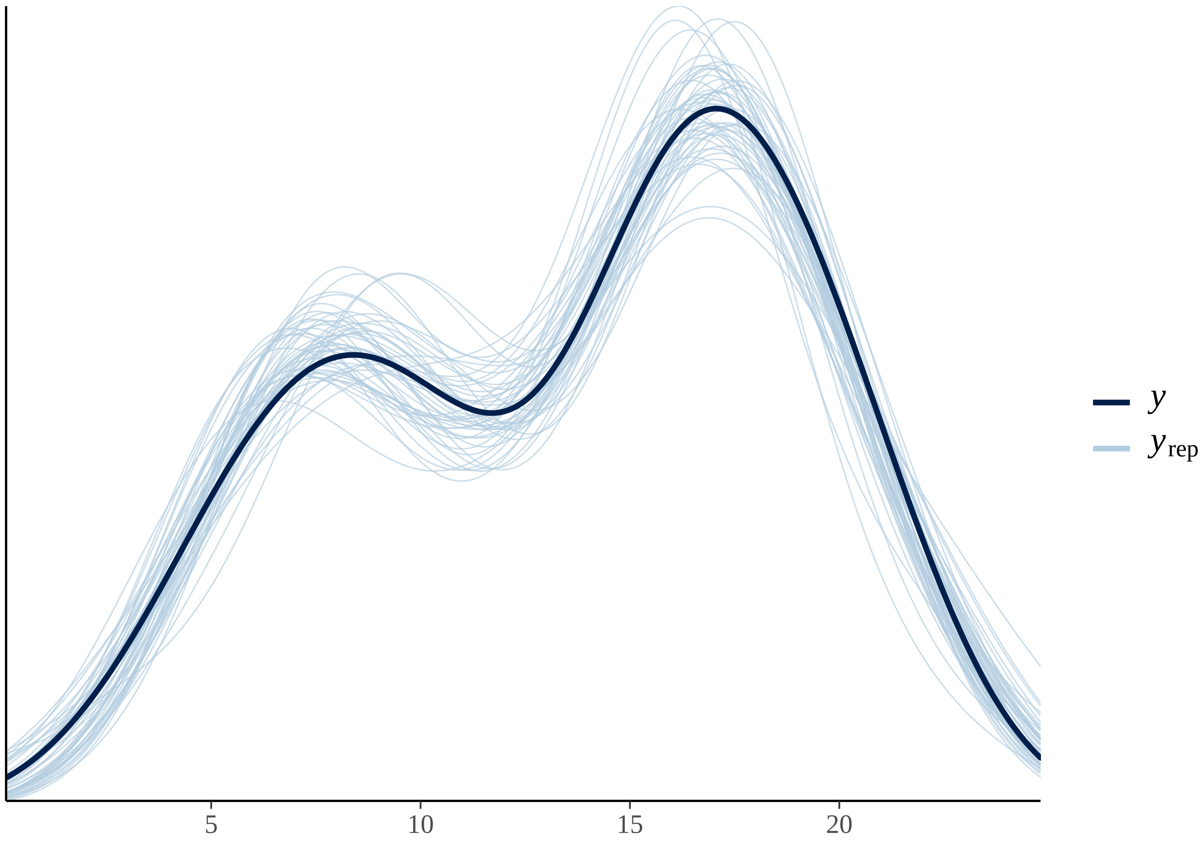
```
f2 <- function(x1, x2){
  sqrt(x1^2+x2^2)
}
n <- 200
fit2 <- lm(I(y^2)~I(x1^2)+I(x2^2)-1, data = Pyth[1:40,])
summary(fit2)
```

```
##
## Call:
## lm(formula = I(y^2) ~ I(x1^2) + I(x2^2) - 1, data = Pyth[1:40,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25966 -0.05413 -0.00175  0.06525  0.35803
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## I(x1^2) 0.9999889  0.0005356    1867   <2e-16 ***
## I(x2^2) 0.9998752  0.0001266    7900   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1326 on 38 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 7.171e+07 on 2 and 38 DF,  p-value: < 2.2e-16
```

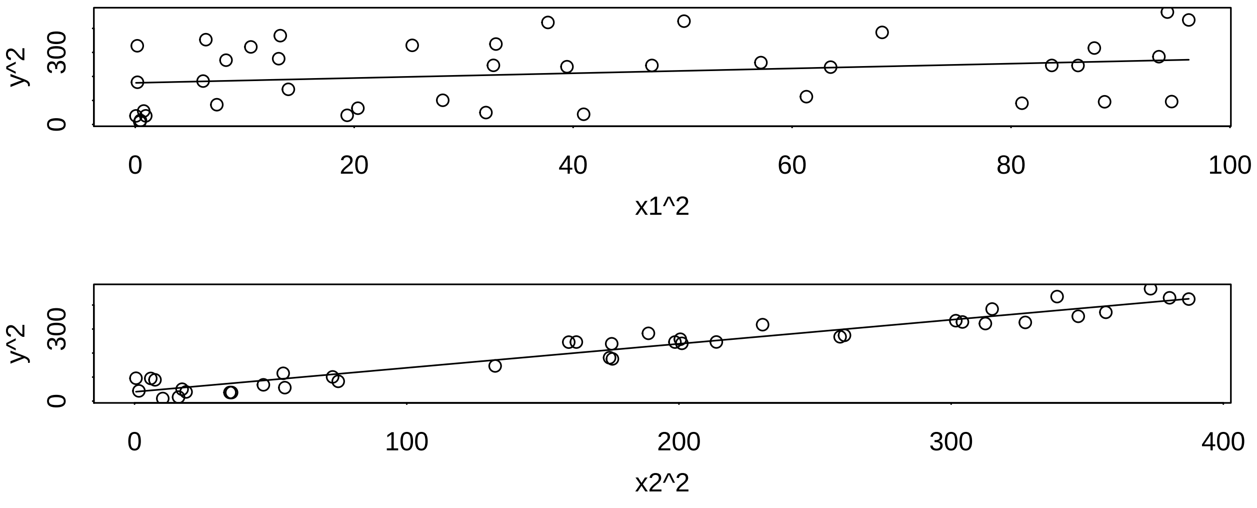
```
plot(fitted(fit2),fit2$residuals)
abline(0,0, col="red")
```



```
fit_11.5d=stan_glm(formula = y~x1^2+x2^2, data = Pyth, subset = 1:40, refresh=0)
pp_check(fit_11.5d)
```



```
par (mar=c(3,3,2,1), mgp=c(2,.7,0), tck=-.01)
par(mfrow=c(2,1))
plot ( Pyth[1:40,]$x1^2,  Pyth[1:40,]$y^2, xlab = "x1^2", ylab="y^2")
curve (coef(fit2)[1]*(x)+ coef(fit2)[2]*(mean(Pyth[1:40,]$x2^2)), add=TRUE)
#abline(lm(y~x1,data=Pyth[1:40,]),lty=2)
plot ( Pyth[1:40,]$x2^2,  Pyth[1:40,]$y^2, xlab = "x2^2", ylab="y^2")
curve ( coef(fit2)[1]*mean(Pyth[1:40,]$x1^2)+ coef(fit2)[2]*(x), add=TRUE)
```



```
#abline(lm(y~x2,data=Pyth[1:40,]),lty=2)
```

12.5

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

(a)

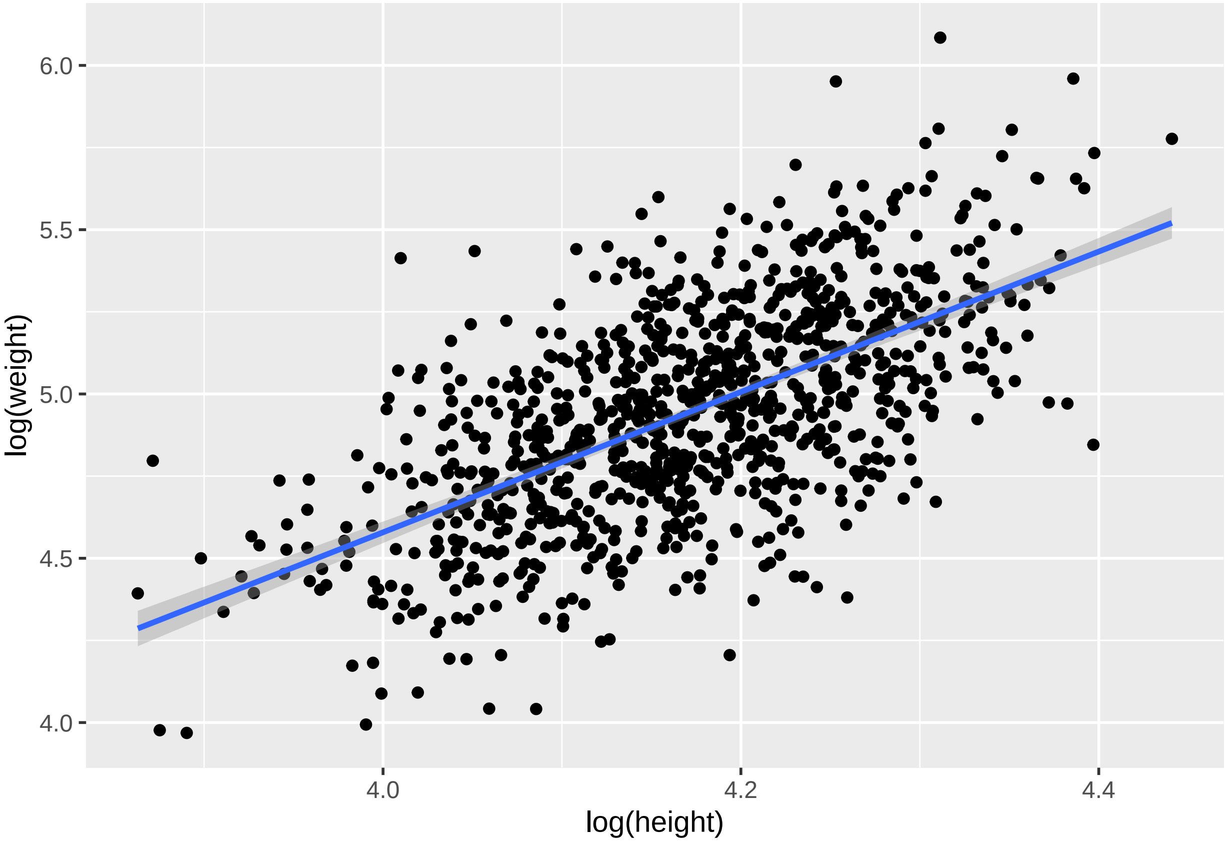
Fill in the blanks: Approximately 68% of the people will have weights within a factor of _________ and _________ of their predicted values from the regression.

within a factor of $exp(\sigma) = 1.284$
and 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.

```r
## Simulate height data
## According to the internet, world adult average height is 165cm (65 inches)
height=rnorm(n = 1000, mean = 65, sd = 6)
weight=exp(-3.8+2.1*log(height)+rnorm(n = 1000, mean = 0, sd = 0.25))
w_h=data.frame(weight=weight, height=height)
ggplot(w_h, aes(x=log(height), y=log(weight)))+
  geom_point()+
  geom_smooth(formula = 'y~x', method = "lm")
```



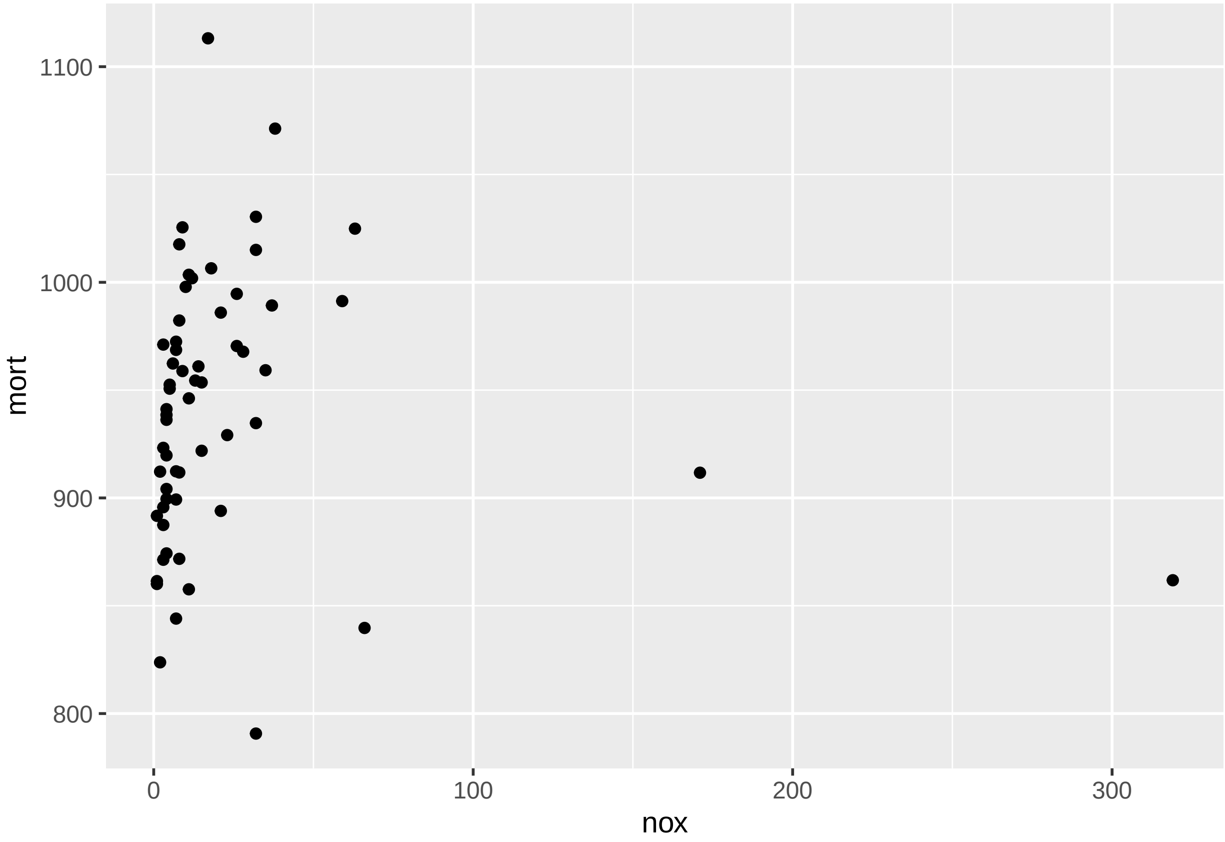
12.6

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 oversimplification, 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=read.csv("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/Pollution/data/poll
ghv_data_dir <- "https://raw.githubusercontent.com/avehtari/ROS-Examples/master/"
Pollution <- read.csv (paste0(ghv_data_dir,"Pollution/data/pollution.csv"), header=TRUE)
ggplot(data = Pollution)+
  geom_point(aes(x=nox,y = mort))
```



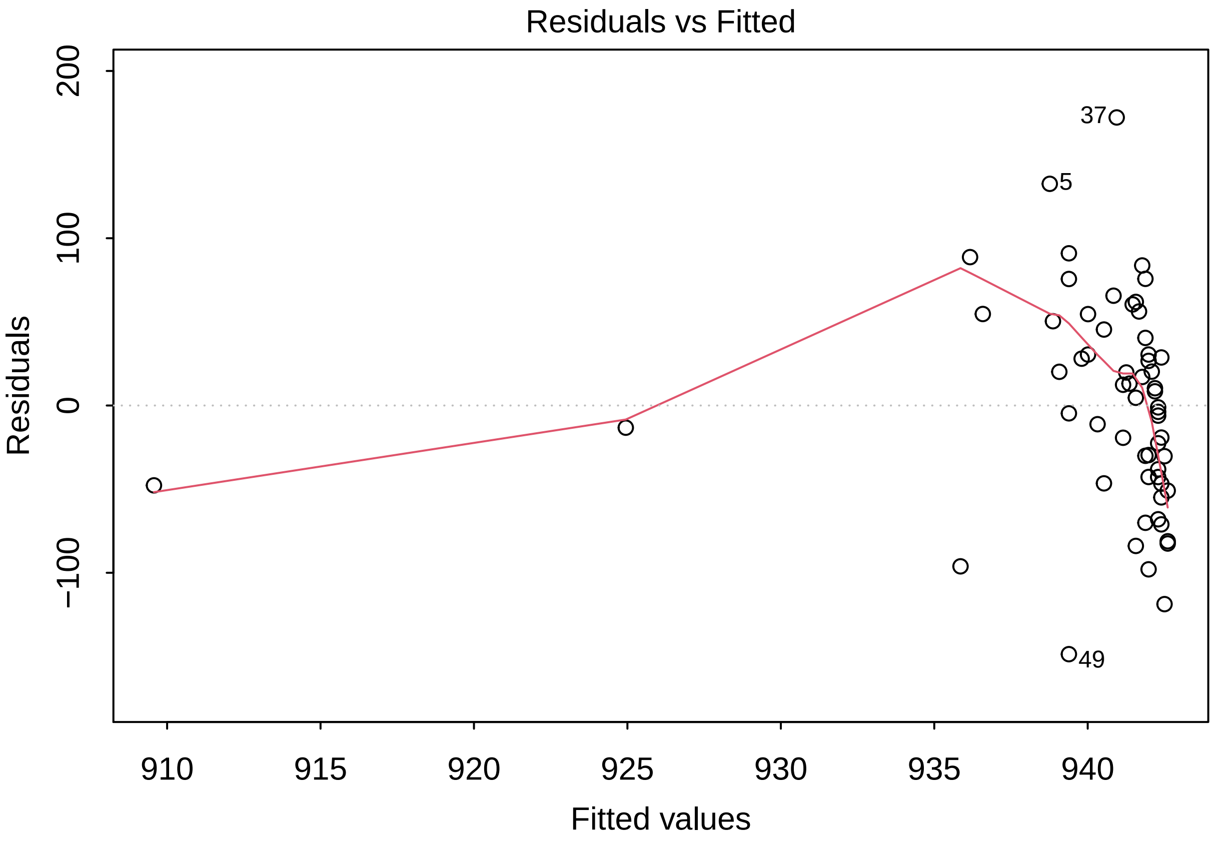
The scatterplot shows that although there's a clear linear trend for most data points where nox<100, points on the bottom right half of the graph might deviate a linear model from the general trend.

```
fit_12.6a=lm(mort~nox, data = Pollution)
summary(fit_12.6a)
```

```
##
## Call:
## lm(formula = mort ~ nox, data = Pollution)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -148.654  -43.710    1.751   41.663  172.211
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 942.7115     9.0034 104.706   <2e-16 ***
## nox          -0.1039     0.1758  -0.591    0.557
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.55 on 58 degrees of freedom
## Multiple R-squared:  0.005987,   Adjusted R-squared:  -0.01115
## F-statistic: 0.3494 on 1 and 58 DF,  p-value: 0.5568
```

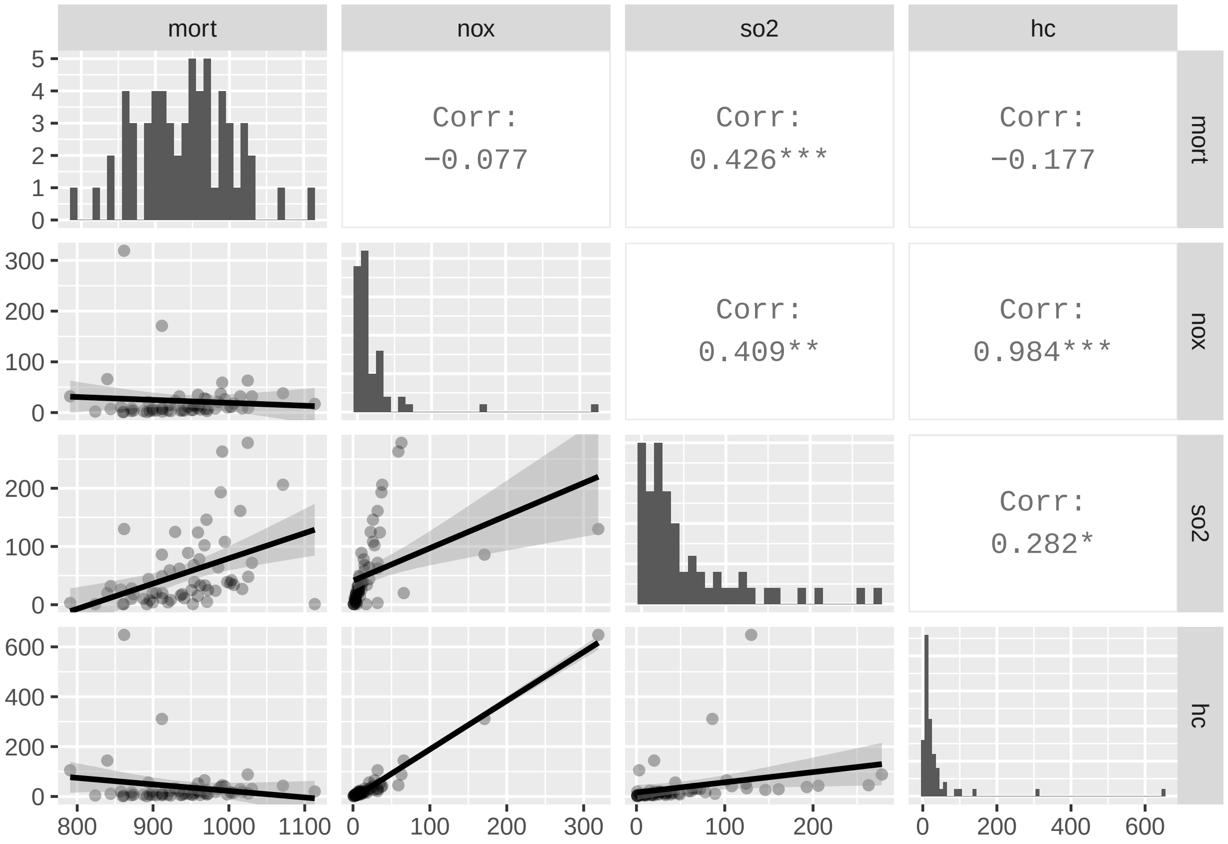
```
par(mar=c(3,3,2,1), mgp=c(2,.7,0), tck=-.01)
plot(fit_12.6a, which = 1)
```



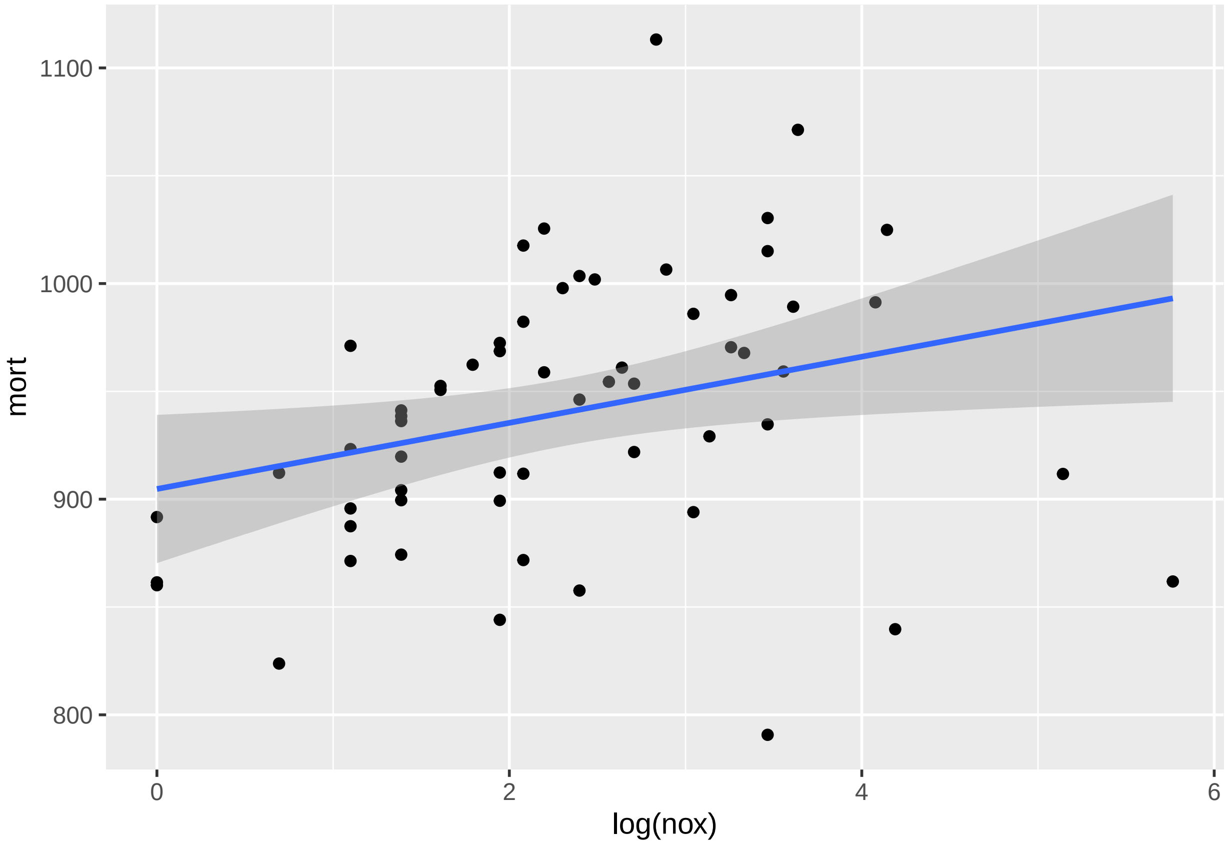
(b)

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

```
## based on the histograms I decided to log transform nox, so2, and hc
## because they are heavily right-skewed
Pollution %>% select (mort, nox, so2, hc) %>%
  ggpairs(upper = list(continuous = wrap("cor")),
          lower = list(continuous = wrap("smooth", alpha = 0.3)),
          diag  = list(continuous = wrap("barDiag", binwidth = 10)))
```



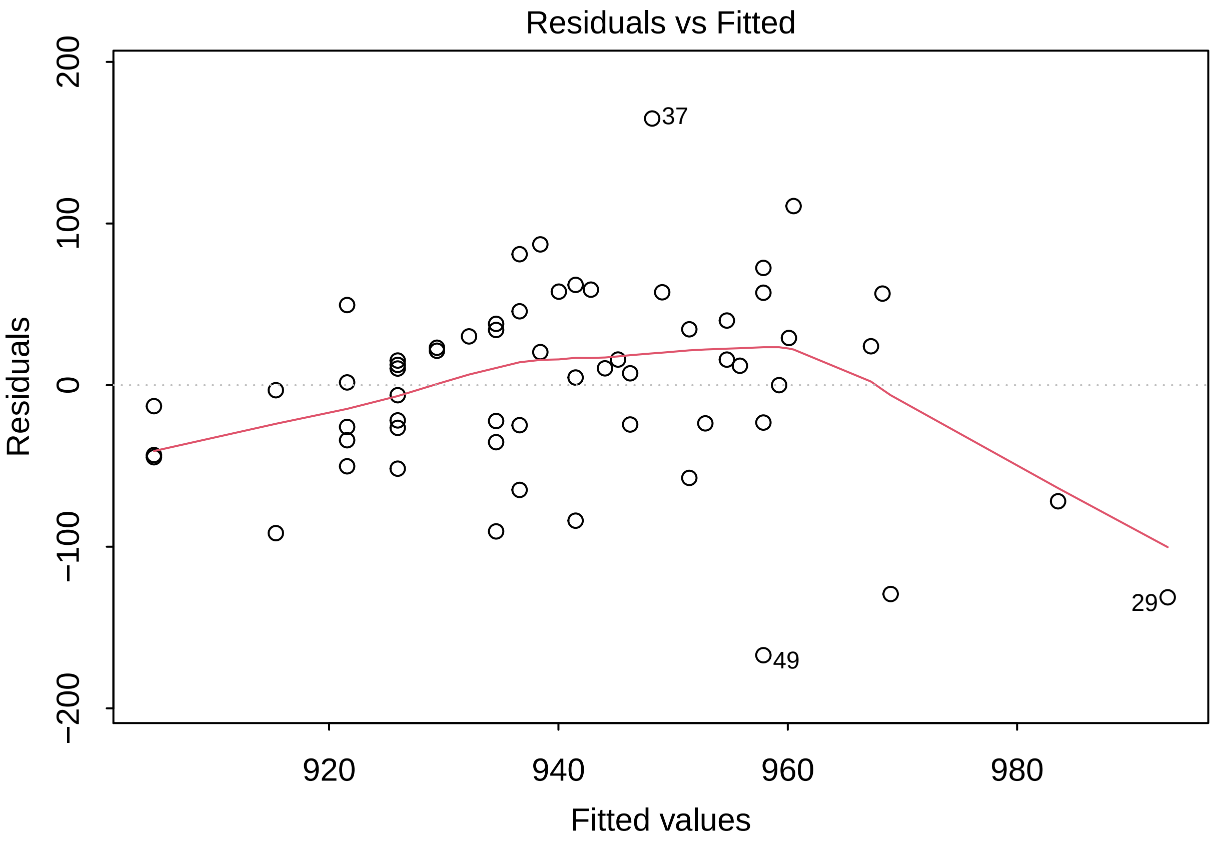
```
## can see a more obvious trend after log transforming nox
ggplot(data = Pollution, aes(x=log(nox),y = mort))+
  geom_point()+
  geom_smooth(formula = "y~x", method = "lm")
```



```
fit_12.6b=lm(mort~log(nox), data = Pollution)
summary(fit_12.6b)
```

```
##
## Call:
## lm(formula = mort ~ log(nox), data = Pollution)
##
## 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
```

```
# residual plot
par(mar=c(3,3,2,1), mgp=c(2,.7,0), tck=-.01)
plot(fit_12.6b, which = 1)
```



(c)

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

On average, every unit increase in nitric oxides on the log scale is associated with 15.3354967 increase in mortality

(d)

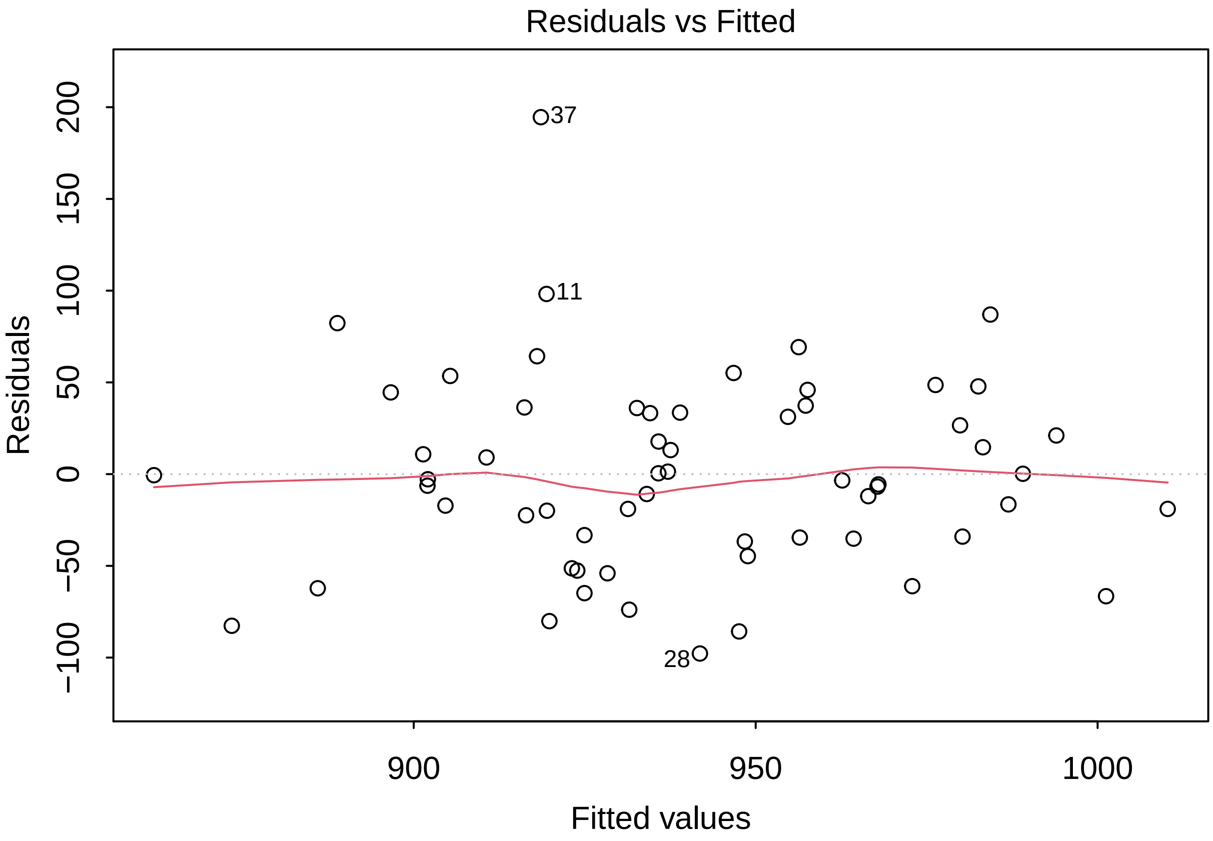
Now fit a model predicting mortality rate using levels of nitric oxides, sulfur dioxide, and hydrocarbons as inputs. Use appropriate transformation when helpful. Plot the fitted regression model and interpret the coefficients.

```
fit_12.6d=lm(mort~log(nox)+log(so2)+log(hc), data = Pollution)
summary(fit_12.6d)
```

```
##
## Call:
## lm(formula = mort ~ log(nox) + log(so2) + log(hc), data = Pollution)
##
## 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
```

On average, every unit increase in nitric oxides, sulfur dioxide, and hydrocarbons on the log scale is associated with 58.3363988, NA, NA increase in mortality respectively

```
# residual plot
par(mar=c(3,3,2,1), mgp=c(2,.7,0), tck=-.01)
plot(fit_12.6d, which = 1)
```



(e)

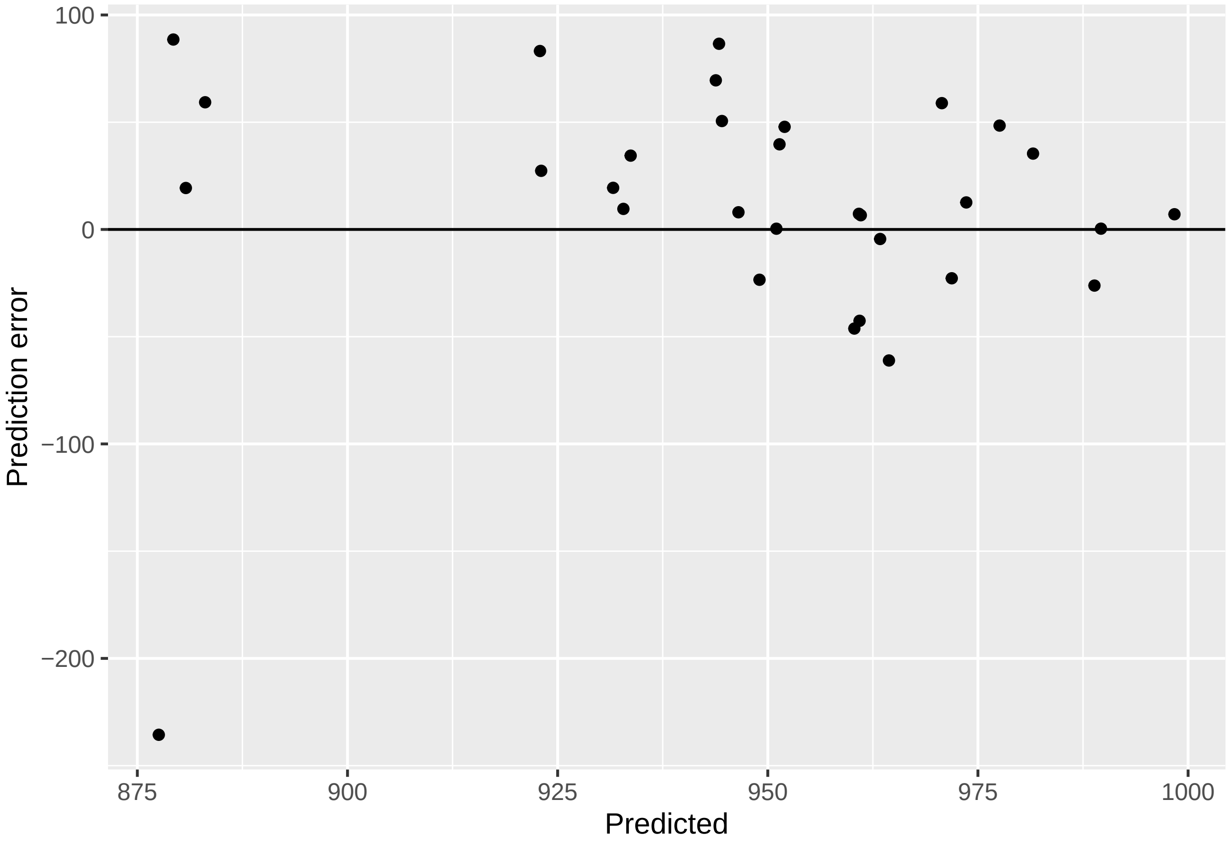
Cross validate: fit the model you chose above to the first half of the data and then predict for the second half. You used all the data to construct the model in (d), so this is not really cross validation, but it gives a sense of how the steps of cross validation can be implemented.

```
## fit the model on the trainning set (using subset = 1:n)
n=dim(Pollution)[1]/2
fit_12.6e=lm(mort~log(nox)+log(so2)+log(hc), data = Pollution, subset = 1:n)
summary(fit_12.6e)
```

```
## 
## Call:
## lm(formula = mort ~ log(nox) + log(so2) + log(hc), data = Pollution, 
##     subset = 1:n)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -110.358  -36.766   -1.032   35.049   82.107 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   899.97      25.71  35.009   <2e-16 ***
## log(nox)       10.57      29.59   0.357   0.7240    
## log(so2)       21.87      12.32   1.774   0.0877 .  
## log(hc)       -17.47      26.21  -0.667   0.5108    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 52.07 on 26 degrees of freedom
## Multiple R-squared:  0.2522, Adjusted R-squared:  0.1659 
## F-statistic: 2.922 on 3 and 26 DF,  p-value: 0.05277
```

```
pred_fit_12.6e=predict(fit_12.6e, newdata = Pollution[n:(2*n),])
## calculate mean squared error on the test set
mse=mean((Pollution$mort[n:(2*n)]-pred_fit_12.6e)**2)
```

```
## Residual plot on the test set
ggplot()+
  geom_point(aes(x=pred_fit_12.6e, y=pred_fit_12.6e-Pollution$mort[n:(2*n)]))+
  geom_hline(yintercept=0)+
  labs(x="Predicted", y="Prediction error")
```



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.

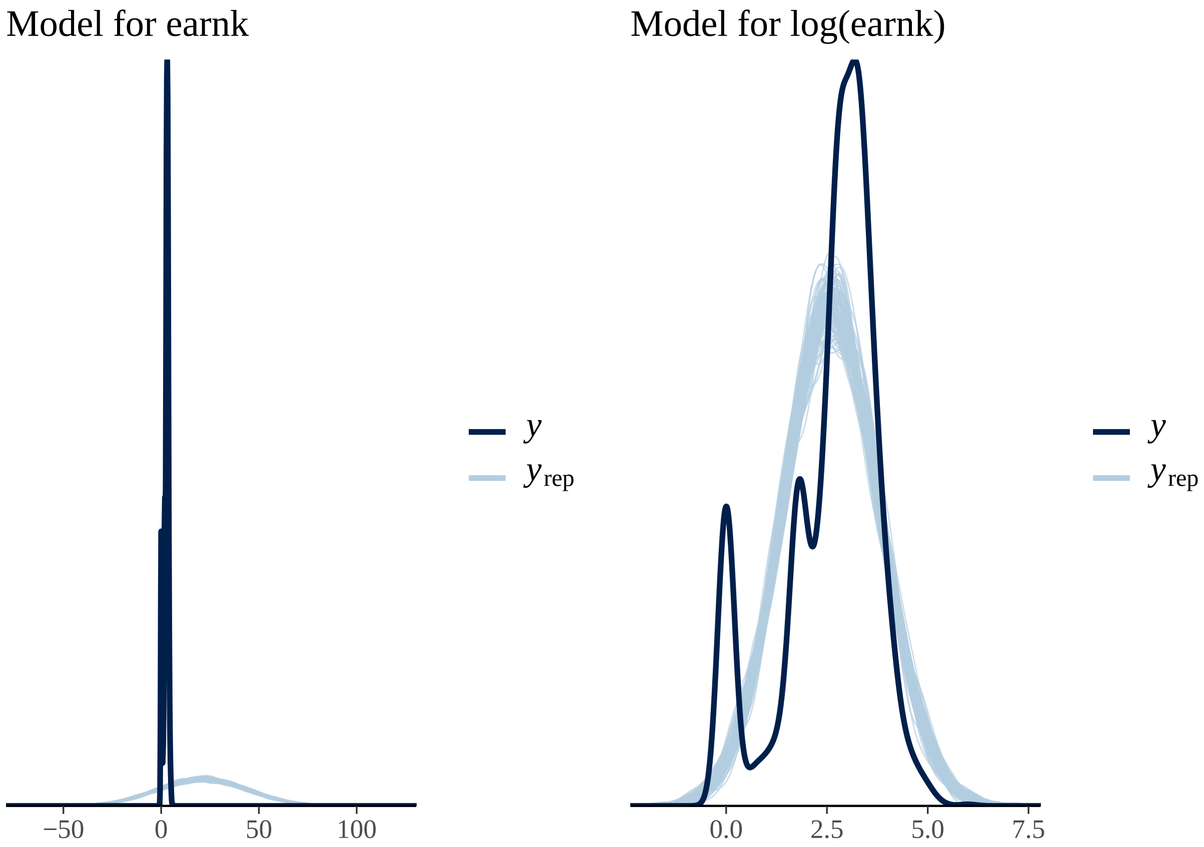
```
#earnings=read.csv("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/Earnings/data/earnin
ghv_data_dir <- "https://raw.githubusercontent.com/avehtari/ROS-Examples/master/"
earnings <- read.csv (paste0(ghv_data_dir,"Earnings/data/earnings.csv"), header=T)
fit_12.7a_1=stan_glm(earnk~height+male, data = earnings, refresh=0)
print(fit_12.7a_1)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      earnk ~ height + male
##  observations: 1816
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept) -25.7   12.0
## height        0.6    0.2
## male         10.7    1.4
##
## 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
```

```
fit_12.7a_2=stan_glm(log(1+earnk)~height+male, data = earnings, refresh=0)
print(fit_12.7a_2)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      log(1 + earnk) ~ height + male
##  observations: 1816
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept) 0.0    0.6
## height      0.0    0.0
## male        0.6    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 1.1    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
## posterior predictive check and comparison
yrep_fit_12.7a_1 <- posterior_predict(fit_12.7a_1)
n_sims <- nrow(yrep_fit_12.7a_1)
sims_display <- sample(n_sims, 100)
ppc_fit_12.7a_1 <- ppc_dens_overlay(log(1+earnings$earnk), yrep_fit_12.7a_1[sims_display,]) +
    theme(axis.line.y = element_blank())
yrep_fit_12.7a_2 <- posterior_predict(fit_12.7a_2)
ppc_fit_12.7a_2 <- ppc_dens_overlay(log(1+earnings$earnk), yrep_fit_12.7a_2[sims_display,]) +
  theme(axis.line.y = element_blank())
bayesplot_grid(
  ppc_fit_12.7a_1, ppc_fit_12.7a_2,
  grid_args = list(ncol = 2),
  titles = c("Model for earnk", "Model for log(earnk)")
)
```



```r
loo_12.7a_1=loo(fit_12.7a_1)
loo_12.7a_2=loo(fit_12.7a_2)
print(loo_12.7a_1)
```

```
##
## Computed from 4000 by 1816 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo  -8156.0 174.5
## p_loo        30.7  22.9
## looic     16311.9 348.9
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     1815  99.9%   788
##  (0.5, 0.7]   (ok)          0   0.0%   <NA>
##    (0.7, 1]   (bad)         0   0.0%   <NA>
##    (1, Inf)   (very bad)    1   0.1%   2
## See help('pareto-k-diagnostic') for details.
```

```r
print(loo_12.7a_2)
```

```
##
## Computed from 4000 by 1816 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2745.7 31.7
## p_loo         3.9  0.2
## looic      5491.3 63.3
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
## loo_compare favors the model with elpd_diff=0
## which corresponds to the model with the largest ELPD (and smallest LOOIC)
loo_compare(loo_12.7a_1, loo_12.7a_2)
```

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

```
##             elpd_diff se_diff
## fit_12.7a_2     0.0       0.0
## fit_12.7a_1 -5410.3     171.4
```

```
# kfold(fit_12.7a_1, K = 10)
```

(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.

$$\hat{\beta}_1 = 2$$
$$\hat{\beta}_0 = log(10) - \hat{\beta}_1 \times log(50) =$$

-5.52

$$log(weight) = -5.52 + 0.02 log(height) + \epsilon$$

and the residual standard deviation would be 0.0476551

(b)

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

$R^2 = 1 - \frac{\hat{\sigma}^2}{s_y^2} = 1 - (\frac{0.047^2}{0.2^2}) = 0.9448$

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, $D_i - R_i$

- advantage: symmetric and centered at zero, which is when $D_i = R_i$. The unit is in dollars so easy to interpret.
- disadvantage: You loose the scale since 7M and 5M will be the same as 2M and 0M. If absolute difference is what matters this may be a good transformation.

(b)

The ratio, D_i/R_i

- advantage: Some people might find this ratio useful (?)
- disadvantage: Center is at 1 when $D_i = R_i$. Asymmetric and unstable when $R_i \approx 0$. Unit is defined relative to R_i so 20M and 1M is the same as 0.2M and 0.01M overly emphasizing the smaller district.

(c)

The difference on the logarithmic scale, $log\ D_i - log\ R_i$

- advantages: this is same as $log(D_i/R_i)$ but if the outcome is on log scale the interpretation may be easier. You may be able to adjust the distribution of the ratio.
- disadvantages: hard to interpret

(d)

The relative proportion, $D_i/(D_i + R_i)$.

- advantages: Symmetric and stable unless $D_i = R_i = 0$.
- disadvantages: Center is at 0.5 and unit become hard to interpret. It hides the scale of money raised.

12.11

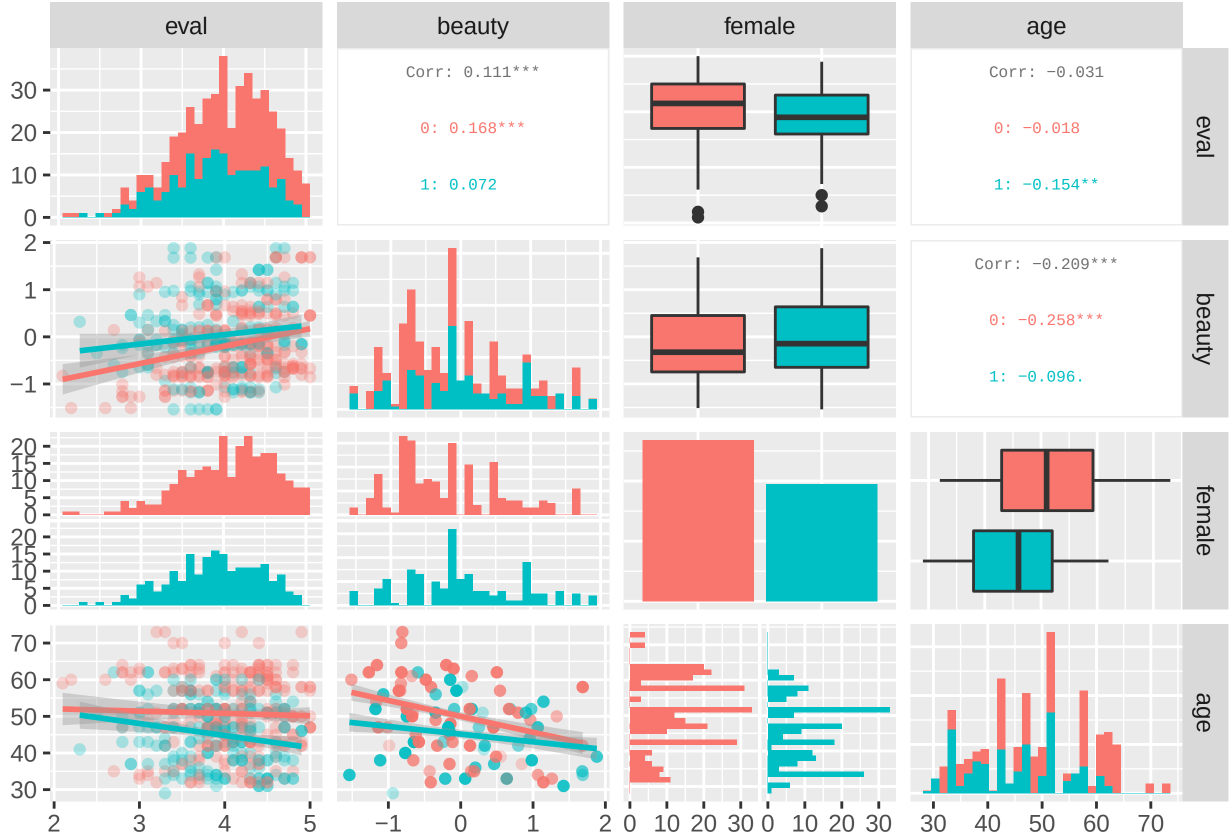
Elasticity: An economist runs a regression examining the relations between the average price of cigarettes, P, and the quantity purchased, Q, across a large sample of counties in the United States, assuming the functional form, $logQ = \alpha + \beta logP$. Suppose the estimate for β is 0.3. Interpret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

On average, every 1% increase in the cigarettes price is associated with 0.3% difference in the quantity purchased. Or, the price elasticity of cigarettes is 0.3

12.13

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

```
ghv_data_dir <- "https://raw.githubusercontent.com/avehtari/ROS-Examples/master/"
beauty <- read.csv (paste0(ghv_data_dir,"Beauty/data/beauty.csv"), header=T)
beauty= beauty %>% mutate(female=factor(female),
                          minority=factor(minority),
                          nonenglish=factor(nonenglish))
beauty %>% select (eval, beauty, female, age) %>%
  ggpairs(upper = list(continuous = wrap("cor", method = "kendall", size=2)),
          lower = list(continuous = wrap("smooth", alpha = 0.3)),
          diag  = list(continuous = wrap("barDiag"), binwidth = 1),
          mapping = aes(color = female))
```



```
fit_12.13=lm(eval~ beauty*female+age*female, data = beauty)
summary(fit_12.13)
```

```
##
## Call:
## lm(formula = eval ~ beauty * female + age * female, data = beauty)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.85649 -0.34566  0.04438  0.37690  1.05652
```

```
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      3.934956   0.175916  22.368  < 2e-16 ***
## beauty           0.214899   0.045597   4.713 3.25e-06 ***
## female1          0.498918   0.274074   1.820  0.06936 .  
## age              0.003374   0.003455   0.977  0.32925    
## beauty:female1 -0.152184   0.066027  -2.305  0.02162 *  
## female1:age    -0.015229   0.005735  -2.656  0.00819 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5328 on 457 degrees of freedom
## Multiple R-squared:  0.08785,    Adjusted R-squared:  0.07787 
## F-statistic: 8.803 on 5 and 457 DF,  p-value: 5.466e-08
```

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.

```
# ghv_data_dir <- "https://raw.githubusercontent.com/avehtari/ROS-Examples/master/"
# mesquite <- read.table(paste0(ghv_data_dir,"Mesquite/data/mesquite.dat"), header=T)
#
# 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)
# pred=posterior_predict(fit_4, newdata = new_trees, fun=exp)
# pred_mean=apply(pred, MARGIN = 2, FUN = mean)
# population_mean=mean(pred_mean)
# population_msd=sd(pred_mean)
```