MA678 Homework 2

9/22/2020

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.

```
library(rstanarm)
```

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

```
## This is rstanarm version 2.21.1
```

```
## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
```

```
## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.
```

```
## - For execution on a local, multicore CPU with excess RAM we recommend calling
```

```
##   options(mc.cores = parallel::detectCores())
```

```
pyth=read.table("C:/Users/dell/Documents/ROS/Pyth/pyth.txt",header = T)
pyth_train=pyth[1:40,]
fit_pyth=stan_glm(y~x1+x2,data=pyth_train,refresh=0)
print(fit_pyth)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x1 + x2
##  observations: 40
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept) 1.3    0.4
## x1          0.5    0.0
## x2          0.8    0.0
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.9    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

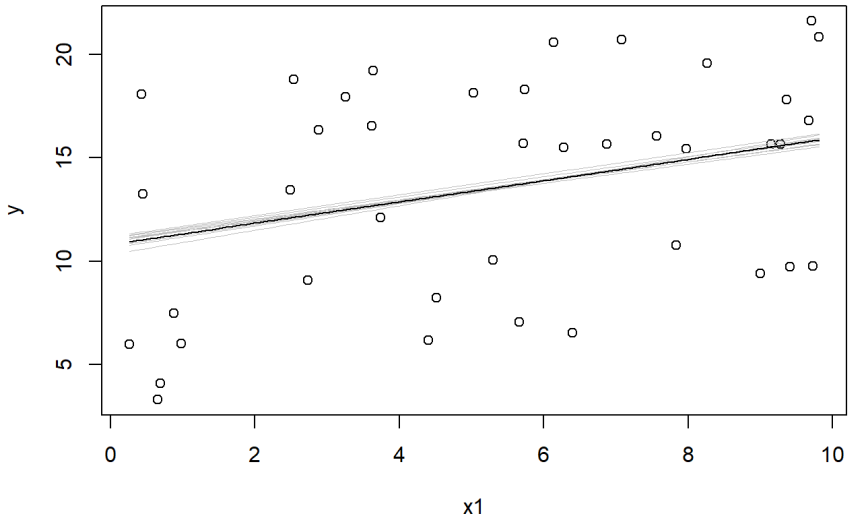
```
#fit_pyth=lm(y~x1+x2,data=pyth_train)
#summary(fit_pyth)
```

(b)

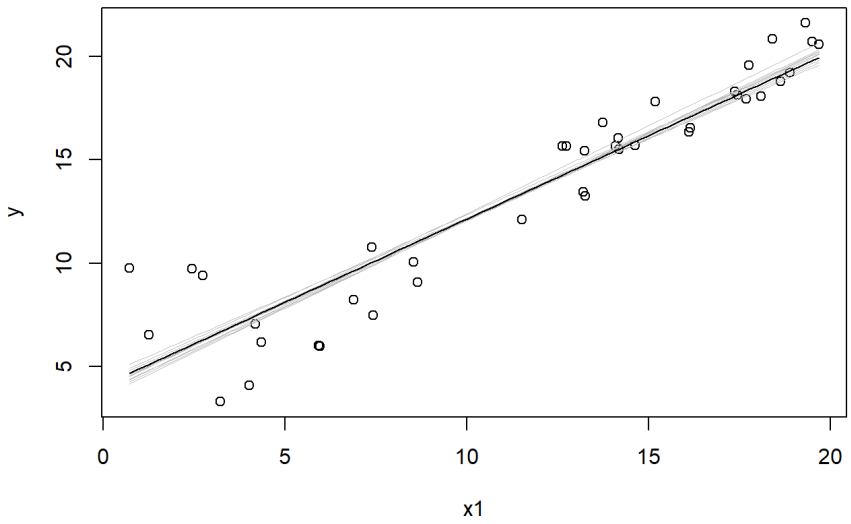
Display the estimated model graphically as in Figure 10.2

```
sim_pyth=as.matrix(fit_pyth)
nsim_pyth=nrow(sim_pyth)
display_pyth=sample(nsim_pyth,10)

par=(mfrow=c(1,2))
plot(pyth_train$x1,pyth_train$y,xlab="x1",ylab="y")
x2_bar=mean(pyth_train$x2)
for(i in display_pyth){
  curve(cbind(1,x,x2_bar) %*% sim_pyth[i,1:3],lwd=0.5,col="gray",add=TRUE)
}
 curve(cbind(1,x,x2_bar)%*%coef(fit_pyth),col="black",add=TRUE)
```



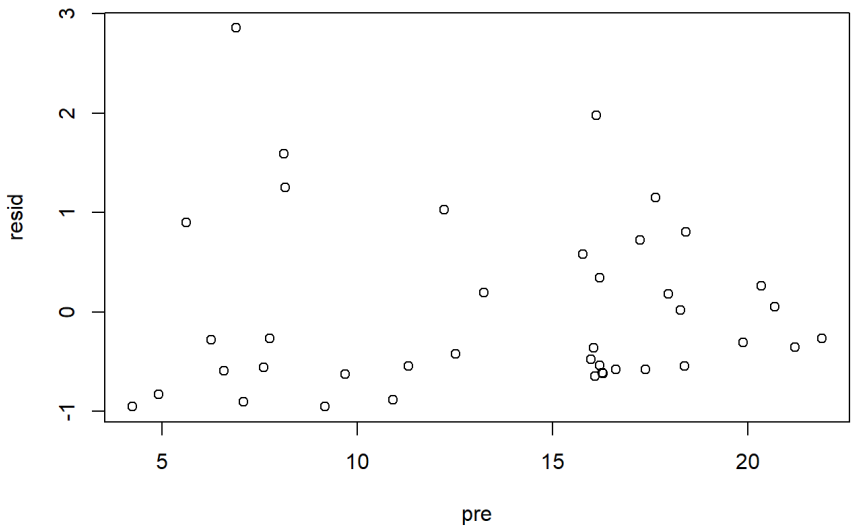

```
plot(pyth_train$x2,pyth_train$y,xlab="x1",ylab="y")
x1_bar=mean(pyth_train$x1)
for(i in display_pyth){
  curve(cbind(1,x1_bar,x)%*%sim_pyth[i,1:3],lwd=0.5,col="gray",add=TRUE)
}
 curve(cbind(1,x1_bar,x)%*%coef(fit_pyth),col="black",add=TRUE)
```



(c)

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

```
pre=predict(fit_pyth)
resid=pyth_train$y-pre
plot(pre,resid)
```



**The assumptions appear to not be met.*

(d)

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

```
pyth_test=pyth[41:60,2:3]
inter=coef(fit_pyth)[1]
a=coef(fit_pyth)[2]
b=coef(fit_pyth)[3]
pre_y=inter+a*pyth_test$x1+b*pyth_test$x2
cbind(pre_y,pyth_test)
```

I am not sure about the prediction since the error sd of this model is 0.9.

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.

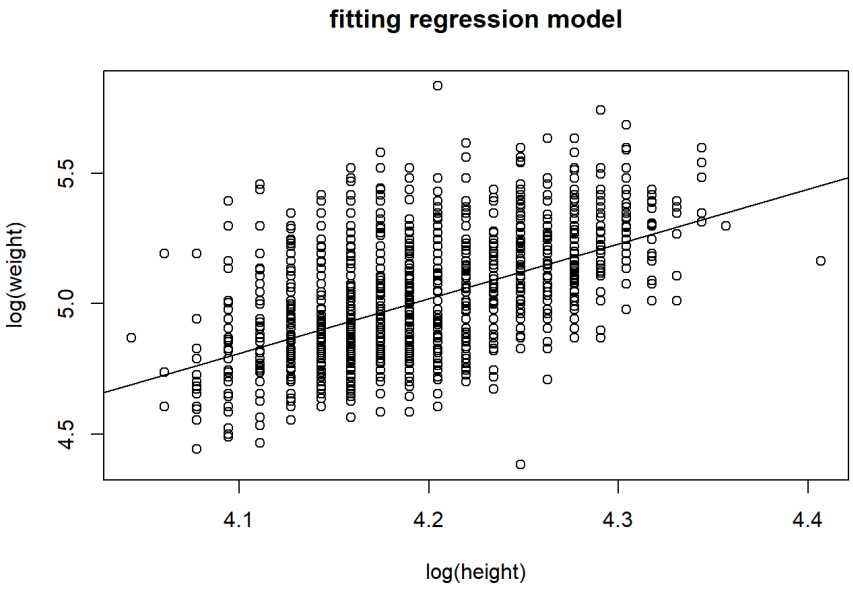
(a)

Fill in the blanks: Approximately 68% of the people will have weights within a factor of **1.28** and **-1.28** 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.

```
data=read.csv("C:/Users/dell/Documents/ROS/Earnings/data/earnings.csv")
plot(log(data$height),log(data$weight),xlab="log(height)",ylab="log(weight)",main="fitting regression model")
abline(a=-3.8,b=2.1)
```



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.

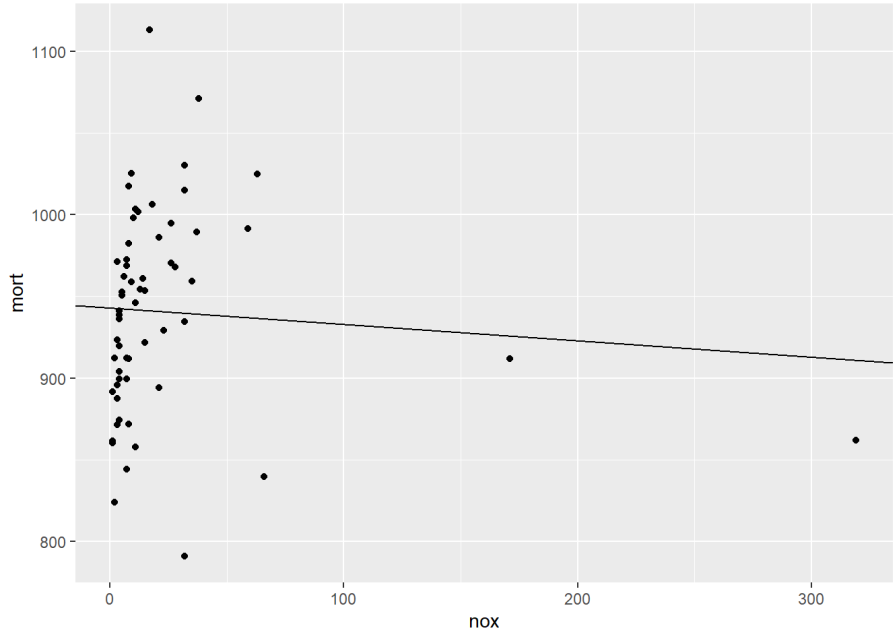
(a)

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.

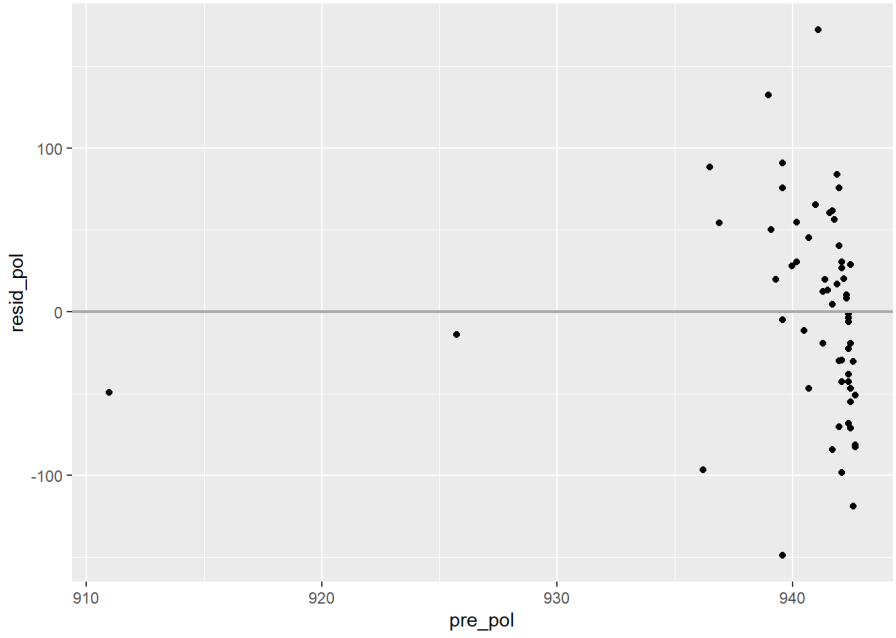
```
library(ggplot2)
pol=read.csv("C:/Users/dell/Documents/ROS/Pollution/data/pollution.csv")
# original regression model
fit1_pol=stan_glm(pol$mort~pol$nox,refresh=0)
print(fit1_pol)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      pol$mort ~ pol$nox
##  observations: 60
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 942.8    9.2
## pol$nox      -0.1    0.2
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 62.6    5.7
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
ggplot(pol,aes(nox,mort))+geom_point()+geom_abline(intercept = coef(fit1_pol)[1],slope =  coef(fit1_pol)[2])
```




```
#residual plot
pre_pol=predict(fit1_pol)
resid_pol=pol$mort-pre_pol
data=data.frame(pre_pol,resid_pol)
ggplot(data,aes(pre_pol,resid_pol))+geom_point()+geom_hline(yintercept = 0,lwd=0.8, color = "darkgray")
```



This linear regression fit these data not well

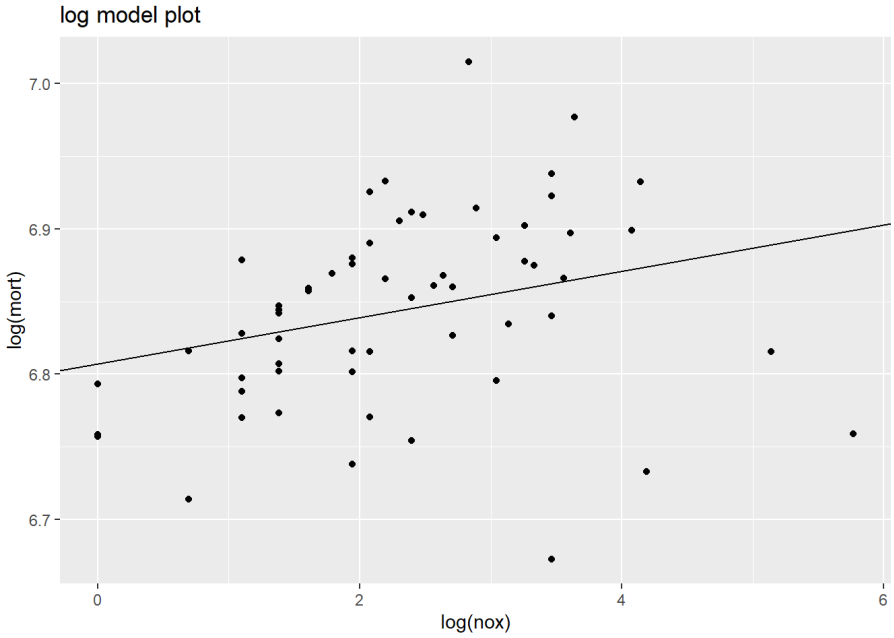
(b)

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.

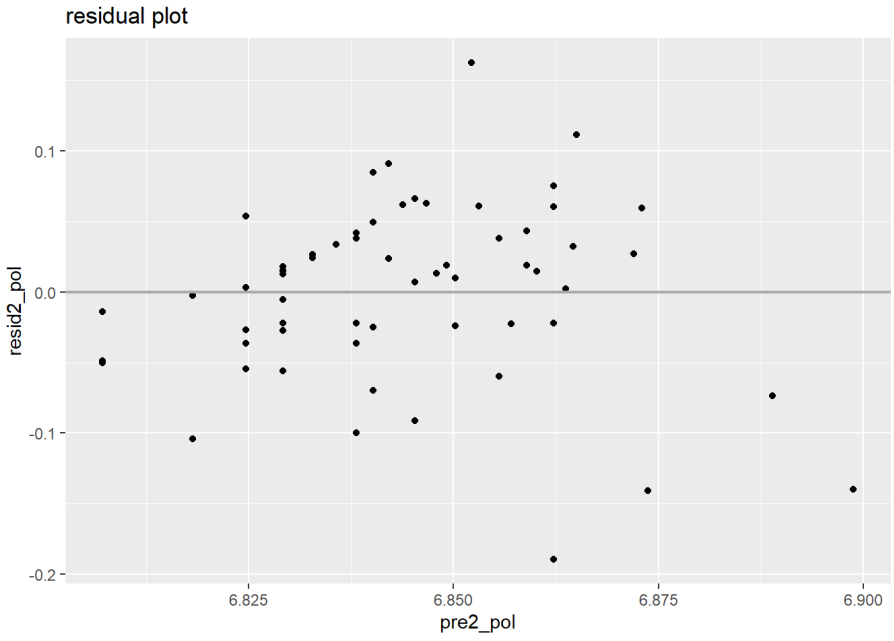
```
#log trans model
fit2_pol=lm(log(pol$mort)~log(pol$nox))
summary(fit2_pol)
```

```
##
## Call:
## lm(formula = log(pol$mort) ~ log(pol$nox))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18930 -0.02957  0.01132  0.03897  0.16275
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.807175   0.018349 370.975   <2e-16 ***
## log(pol$nox) 0.015893   0.007048   2.255   0.0279 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06412 on 58 degrees of freedom
## Multiple R-squared:  0.08061,    Adjusted R-squared:  0.06476
## F-statistic: 5.085 on 1 and 58 DF,  p-value: 0.02792
```

```
ggplot(pol,aes(log(nox),log(mort)))+geom_point()+geom_abline(intercept = coef(fit2_pol)[1],slope =  coef(fit2_pol)[2])+labs( title = "log model plot")
```




```
#residual plot
pre2_pol=predict(fit2_pol)
resid2_pol=log(pol$mort)-pre2_pol
data=data.frame(pre2_pol,resid2_pol)
ggplot(data,aes(pre2_pol,resid2_pol))+geom_point()+geom_hline(yintercept = 0,lwd=0.8, color = "darkgray")+  labs( title = "residual plot")
```



(c)

Interpret the slope coefficient from the model you chose in (b) **the intercept 6.807175:the predicted log mortality rate is 6.81 if level of nox is 1 the coef 0.015893: each 1% difference in nox,the difference in mortality rate is 0.016%**

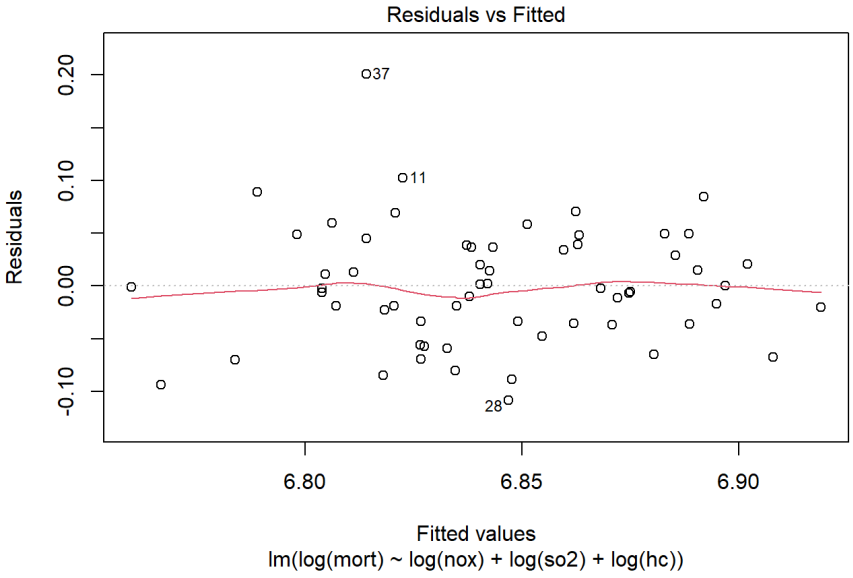
(d)

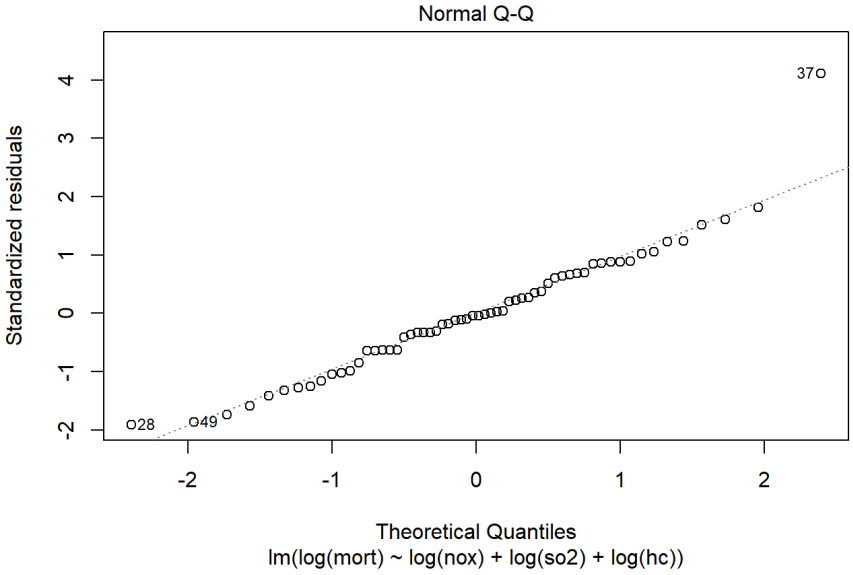
Now fit a model predicting mortality rate using levels of nitric oxides, sulfur dioxide, and hydrocarbons as inputs. Use appropriate transformation when helpful. Plot the fitted regression model and interpret the coefficients.

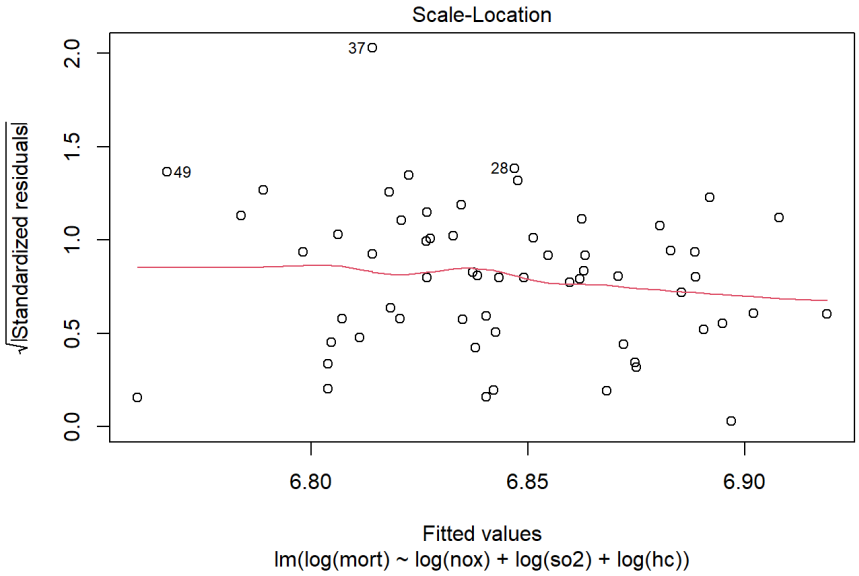
```
nox=pol$nox
so2=pol$so2
hc=pol$hc
mort=pol$mort
fit3_pol=lm(log(mort)~log(nox)+log(so2)+log(hc),data=pol)
summary(fit3_pol)
```

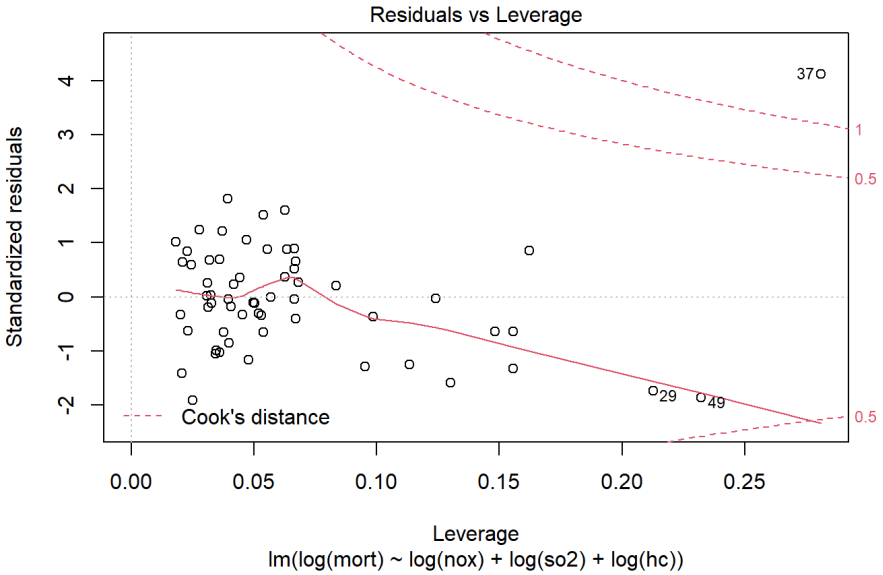
```
##
## Call:
## lm(formula = log(mort) ~ log(nox) + log(so2) + log(hc), data = pol)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.10874 -0.03574 -0.00218  0.03709  0.20085
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.826749   0.022701 300.726  < 2e-16 ***
## log(nox)     0.059837   0.023021   2.599  0.01192 *
## log(so2)     0.014309   0.007584   1.887  0.06436 .
## log(hc)     -0.060812   0.020553  -2.959  0.00452 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05753 on 56 degrees of freedom
## Multiple R-squared:  0.2852, Adjusted R-squared:  0.2469
## F-statistic: 7.449 on 3 and 56 DF,  p-value: 0.0002777
```

```
plot(fit3_pol)
```


Residuals vs Fitted
Residuals
0.20
0.10
0.00
-0.10
37
11
28
6.80
6.85
6.90
Fitted values
lm(log(mort) ~ log(nox) + log(so2) + log(hc))


Normal Q-Q
Standardized residuals
4
3
2
1
0
-1
-2
37
28
49
-2
-1
0
1
2
Theoretical Quantiles
lm(log(mort) ~ log(nox) + log(so2) + log(hc))


Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
37
49
28
6.80
6.85
6.90
Fitted values
lm(log(mort) ~ log(nox) + log(so2) + log(hc))



log(nox): each 1% difference in nox,the difference in mortality rate is 0.060% log(so2): each 1% difference in so2,the difference in mortality rate is 0.014 log(hc): each 1% difference in hc,the difference in mortality rate is 0.061% the intercept:the predicted log mortality rate is 6.83 if levels of nox so2 and hc are 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.

```
pol_train=pol[1:30,]
pol_test=pol[31:60,]
fit4_pol=lm(log(mort)~log(nox)+log(so2)+log(hc),data=pol_train)
summary(fit4_pol)
```

```
##
## Call:
## lm(formula = log(mort) ~ log(nox) + log(so2) + log(hc), data = pol_train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.121185 -0.036920  0.000443  0.036738  0.081083
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.80248    0.02735 248.684   <2e-16 ***
## log(nox)     0.01070    0.03149   0.340   0.7368
## log(so2)     0.02298    0.01311   1.752   0.0915 .
## log(hc)     -0.01815    0.02789  -0.651   0.5208
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05541 on 26 degrees of freedom
## Multiple R-squared:  0.2448, Adjusted R-squared:  0.1577
## F-statistic: 2.809 on 3 and 26 DF,  p-value: 0.05926
```

```
inter=coef(fit4_pol)[1]
a=coef(fit4_pol)[2]
b=coef(fit4_pol)[3]
c=coef(fit4_pol)[4]

pre_mort=inter+a*log(pol_train$nox)+b*log(pol_train$so2)+log(pol_train$hc)
#cbind(mort=pol_test$mort,pre_mort)
```

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.

```
earn=read.csv("C:/Users/dell/Documents/ROS/Earnings/data/earnings.csv")
model=stan_glm(earnk~height+male,data=earn,subset=earnk>0,refresh=0)
logmodel=stan_glm(log(earnk)~height+male,data=earn,subset=earnk>0,refresh=0)
print(model)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      earnk ~ height + male
##  observations: 1629
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept) -18.8   12.7
## height        0.6    0.2
## male          8.9    1.6
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 21.7    0.4
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
print(logmodel)
```

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

(b)

Compare models from other exercises in this chapter.

```
library(loo)
```

```
## This is loo version 2.3.1
```

```
## - Online documentation and vignettes at mc-stan.org/loo
```

```
## - As of v2.0.0 loo defaults to 1 core but we recommend using as many as possible. Use the 'cores' argument or set options(m
c.cores = NUM_CORES) for an entire session.
```

```
## - Windows 10 users: loo may be very slow if 'mc.cores' is set in your .Rprofile file (see https://github.com/stan-dev/loo/is
sues/94).
```

```
loo1=loo(model,k_threshold=0.7)
```

```
## 1 problematic observation(s) found.
## Model will be refit 1 times.
```

```
##
## Fitting model 1 out of 1 (leaving out observation 399)
```

```
loo2=loo(logmodel)
loo_compare(loo1,loo2)
```

```
##          elpd_diff se_diff
## logmodel     0.0       0.0
## model    -5264.4     169.5
```

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.

** $log(weigh)=-5.52+2log(height)$ $sd = log(1.1)/2 = 0.048$ **

(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?

** $R^2=1-sd^2/0.2^2=0.94$ **

12.9

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

(a)

The simple difference, $D_{i}-R_{i}$

The simple difference can directly reflect the difference between the amount of money raised by two parties.We can compare the simple difference with 0. But it cannot tell us how big the difference is, compared to the variable itself.

(b)

The ratio, D_{i}/R_{i}

The ratio can reflect the relative size of the difference compared with the variable. We can compare the ratio with 1. But it cannot show the specific value of the difference.

(c)

The difference on the logarithmic scale, $log\ D_{i}-log\ R_{i}$

The difference on the logarithmic scale equals to $log(D_{i}/R_{i})$. It has similar advantage annd disadvantage with the ratio D_{i}/R_{i} but can transform it to logarithmic scale.

(d)

The relative proportion, $D_{i}/(D_{i}+R_{i})$.

The value of relative proportion falls into $(0,1)$. We can compare it with 0.5. But we cannot know the specific value of the difference.

12.11

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

for each 1% difference in price of cigarettes, the difference in quantity purchased is 0.3%

12.13

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

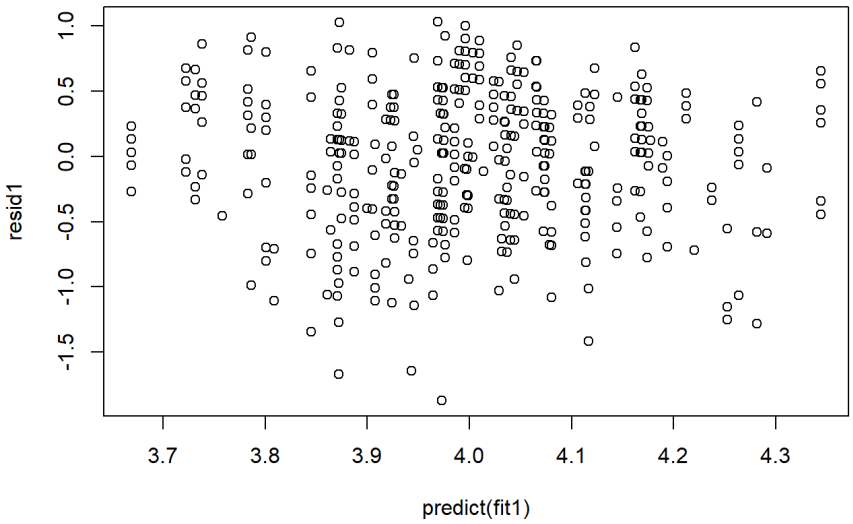
```
library(ggplot2)
beauty=read.csv("C:/Users/dell/Documents/ROS/Beauty/data/beauty.csv")
#1
fit1= stan_glm(eval ~ beauty + female, data=beauty, refresh=0)
print(fit1)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      eval ~ beauty + female
##  observations: 463
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept)  4.1    0.0
## beauty       0.1    0.0
## female      -0.2    0.0
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.5    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

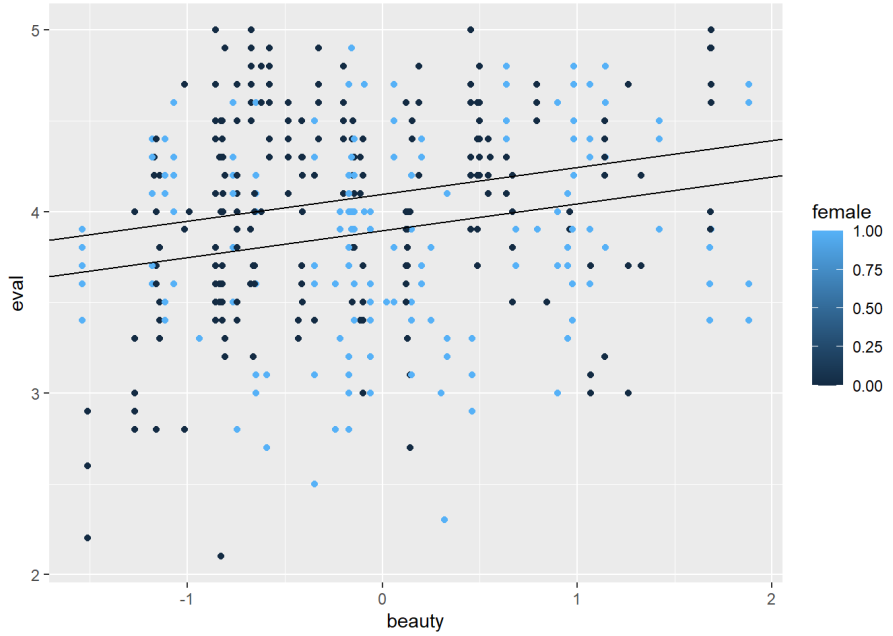
```
median(bayes_R2(fit1))
```

```
## [1] 0.06763786
```

```
resid1=beauty$eval-predict(fit1)
plot(predict(fit1),resid1)
```




```
ggplot(beauty,aes(beauty,eval,col=female))+geom_point()+geom_abline(intercept = coef(fit1)[1],slope=coef(fit1)[2])+geom_abline
(intercept = coef(fit1)[1]+coef(fit1)[3],slope=coef(fit1)[2])
```



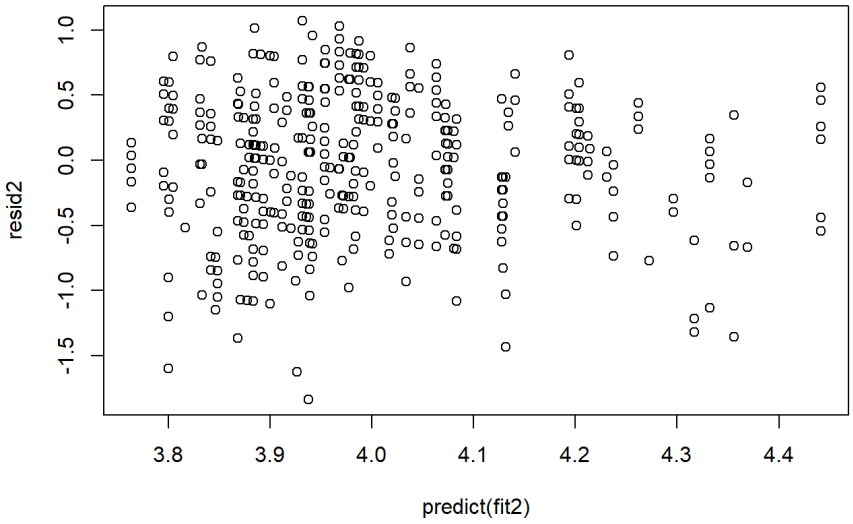

```
#2
fit2=stan_glm(eval ~ beauty + female + beauty:female, data=beauty, refresh=0)
print(fit2)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      eval ~ beauty + female + beauty:female
##  observations: 463
##  predictors:   4
## ------
##               Median MAD_SD
## (Intercept)    4.1    0.0
## beauty         0.2    0.0
## female        -0.2    0.1
## beauty:female -0.1    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.5    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

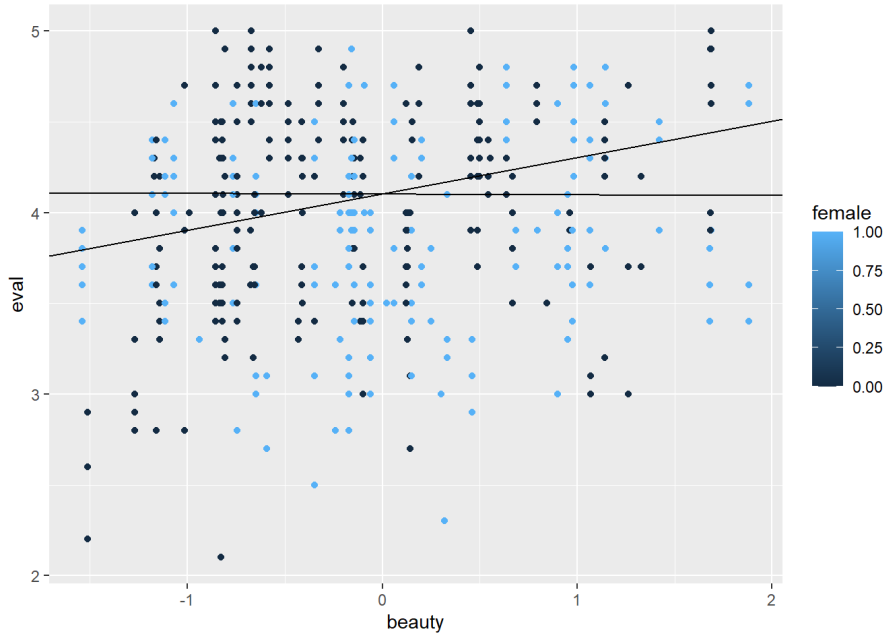
```
median(bayes_R2(fit2))
```

```
## [1] 0.07576873
```

```
resid2=beauty$eval-predict(fit2)
plot(predict(fit2),resid2)
```




```
ggplot(beauty,aes(beauty,eval,col=female))+geom_point()+geom_abline(intercept = coef(fit2)[1],slope=coef(fit2)[2])+geom_abline
(intercept = coef(fit2)[1],slope=coef(fit2)[2]+coef(fit2)[3])
```



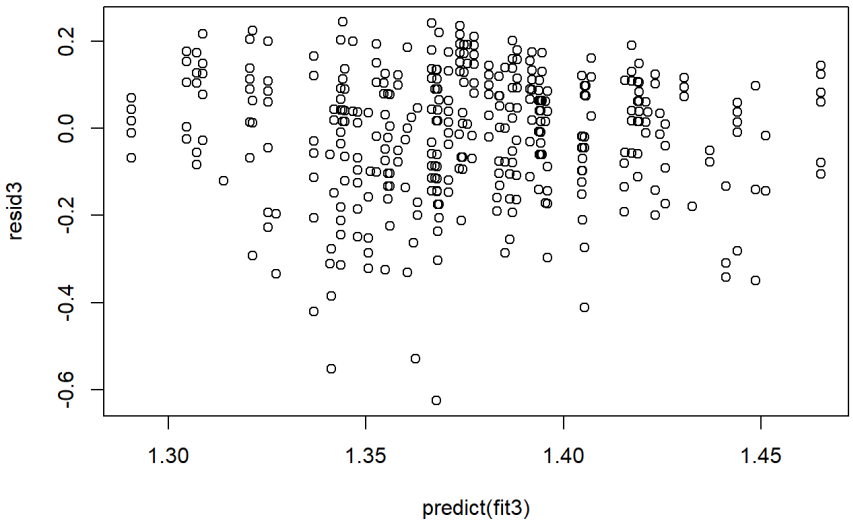

```
#3
fit3=stan_glm(log(eval) ~ beauty + female, data=beauty, refresh=0)
print(fit3)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      log(eval) ~ beauty + female
##  observations: 463
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept) 1.4    0.0
## beauty      0.0    0.0
## female      0.0    0.0
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.1    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

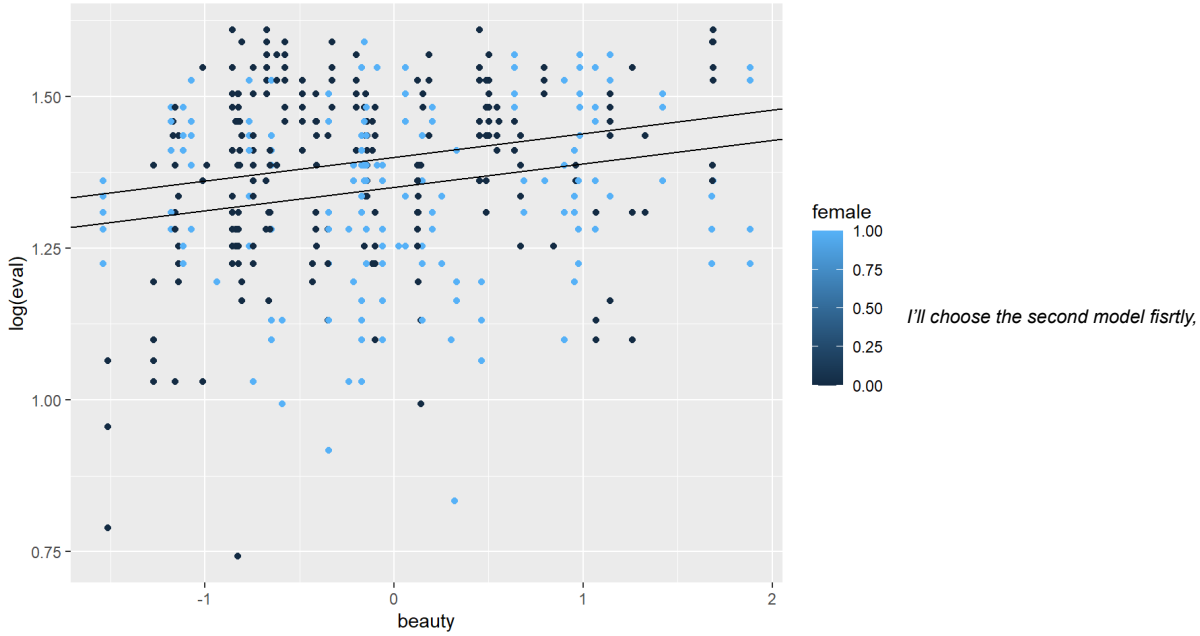
```
median(bayes_R2(fit3))
```

```
## [1] 0.06398194
```

```
resid3=log(beauty$eval)-predict(fit3)
plot(predict(fit3),resid3)
```




```
ggplot(beauty,aes(beauty,log(eval),col=female))+geom_point()+geom_abline(intercept = coef(fit3)[1],slope=coef(fit3)[2])+geom_ab
line(intercept = coef(fit3)[1]+coef(fit3)[3],slope=coef(fit3)[2])
```



I'll choose the second model fisrtly, it's bayes R^2 is the largest Secondly, it residual vs fitted value scartterplot is distributed around y=0

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.

the code is:

mes=read.table("C:/Users/dell/Documents/ROS/Mesquite/data/mesquite.dat",head=T)

mes\(canopy_volum=mes\)diam1*mes\(diam2*mes\)canopy_height

mes\(canopy_area=mes\)diam1*mes$diam2

mes\(canopy_shape=mes\)diam1/mes$diam2

fit4=stan_glm(log(weight)~log(diam1*diam2*canopy_height)+log(diam1*diam2)+log(diam1/diam2)+log(total_height)+log(density)+group,data=mes,subset=mes>0,

log(pre_mes)=predict(fit4,data=new_trees)

print(pre_mes,mean(pre_mes),sd(pre_mes)) ```