MA678 Homework 2

2025-09-26

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.

```
df <- read.table("pyth.txt",header=TRUE)
head(df)
```

```
      y   x1    x2
1 15.68 6.87 14.09
2  6.18 4.40  4.35
3 18.10 0.43 18.09
4  9.07 2.73  8.65
5 17.97 3.25 17.68
6 10.04 5.30  8.53
```

```
data_points <- df[1:40, ]
no_data_points <-df[41:60,]
fit<- lm(y ~ x1+x2,data= data_points)
summary(fit)
```

```
Call:
lm(formula = y ~ x1 + x2, data = data_points)

Residuals:
    Min      1Q  Median      3Q     Max
-0.9585 -0.5865 -0.3356  0.3973  2.8548

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.31513    0.38769   3.392  0.00166 **
x1           0.51481    0.04590  11.216 1.84e-13 ***
x2           0.80692    0.02434  33.148  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9 on 37 degrees of freedom
Multiple R-squared:  0.9724,    Adjusted R-squared:  0.9709
F-statistic: 652.4 on 2 and 37 DF,  p-value: < 2.2e-16
```

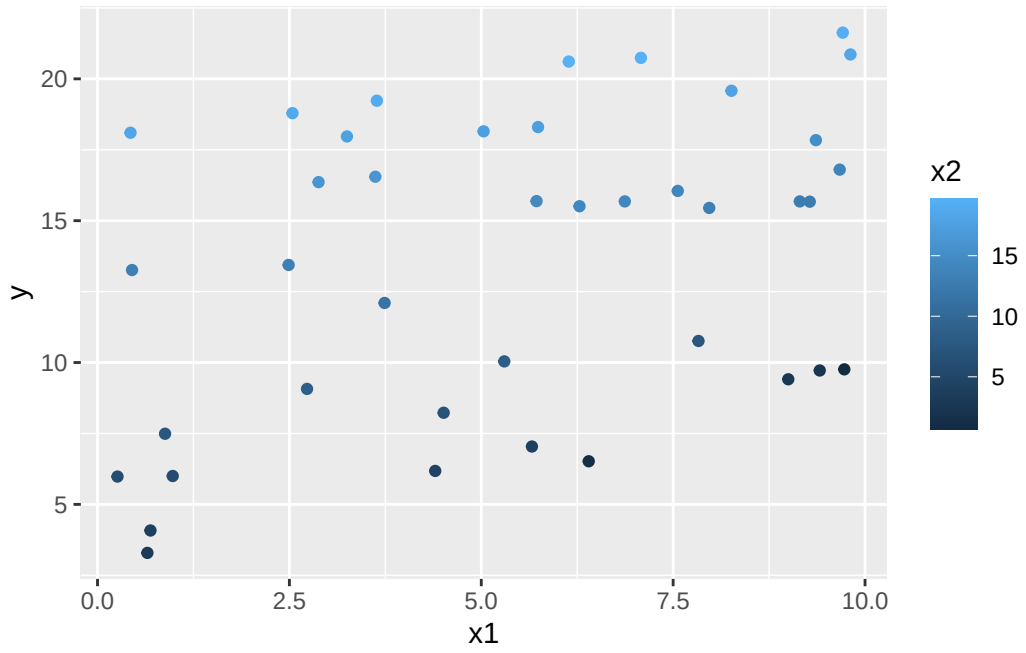
```
# the result shows that p value for intercept, coefficient of x1 and x2 are all very small, t
```

(b)

Display the estimated model graphically as in Figure 10.2

```
library(ggplot2)
ggplot(data_points,aes(x=x1,y=y,color=x2))+
  geom_point()+
  geom_smooth(method="lm",formula= y~x1+x2,se=FALSE)
```

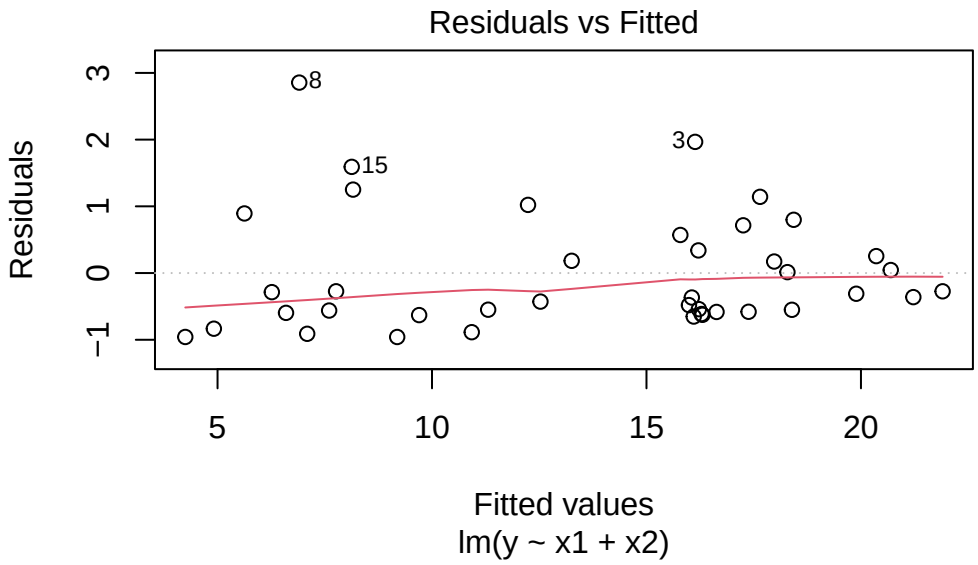
```
Warning: Failed to fit group -1.
Caused by error:
! object 'x1' not found
```



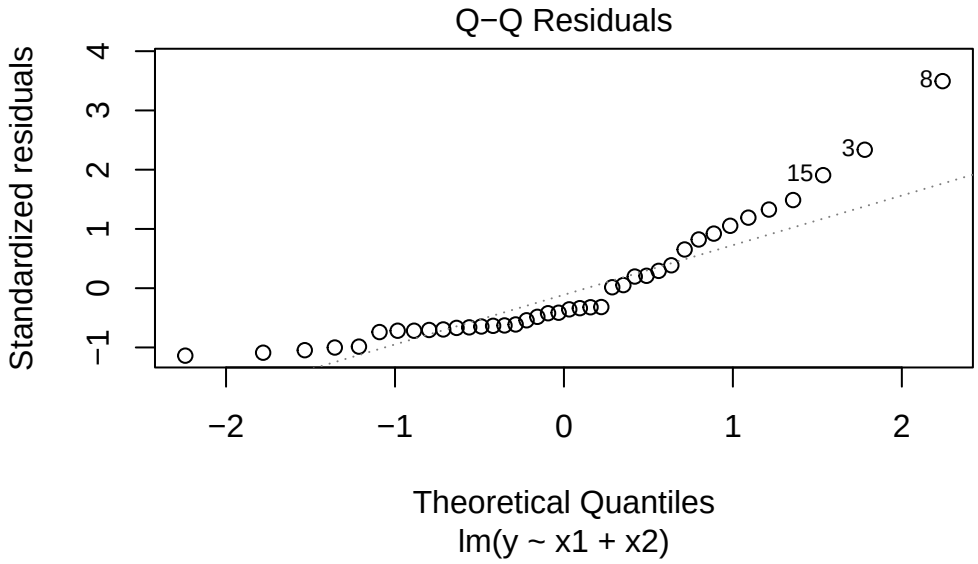
(c)

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

```
plot(fit, which = 1)  # Residuals vs Fitted
```




```r
plot(fit, which = 2)  # Normal Q-Q
```



```
# most residuals are around 0 and the smooth line is almost flat
#there a few outliers with high residuals but not so extreme
#the R^2 = 0.97 which is really good
#overall, the assumptions is met and the model is good
```

(d)

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

```
pred <- predict(fit, newdata = no_data_points, interval = "prediction")
head(pred,20)
```

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

```
# showing the result of 95% confidence interval of fitted value and it's looking good
```

12.5

Logarithmic transformation and regression: Consider the following regression:

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

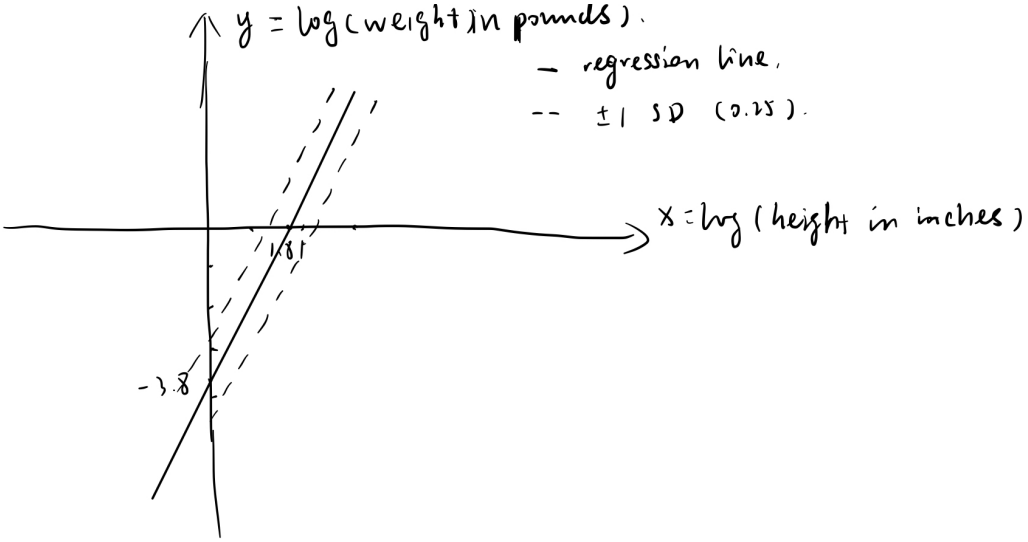
with errors that have standard deviation 0.25. Weights are in pounds and heights are in inches.

(a)

Fill in the blanks: Approximately 68% of the people will have weights within a factor of ***e^(-0.25)=0.78*** and ***e^(0.25)=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.



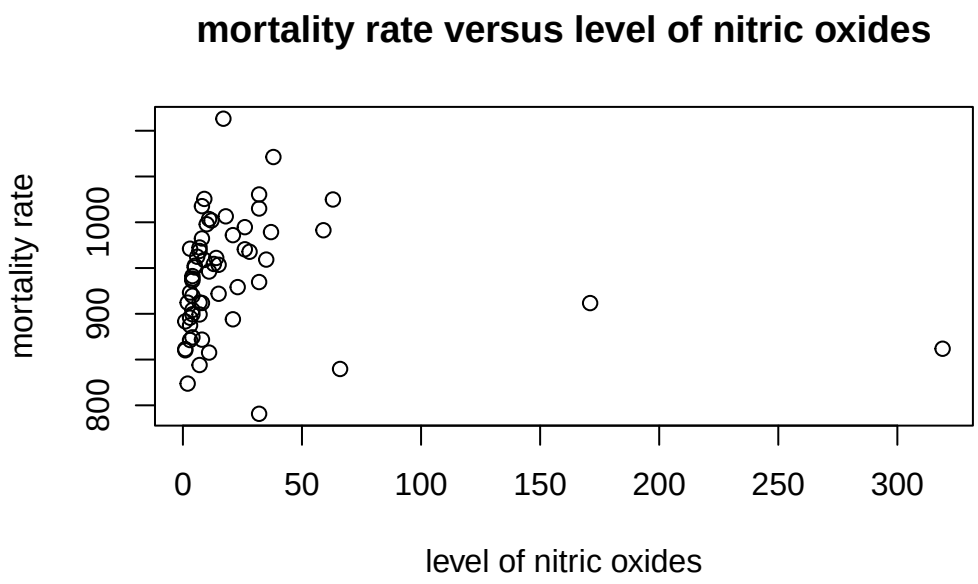
12.6

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.

```
df <- read.csv("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/Pollution/data
plot(df$nox,df$mort,xlab = "level of nitric oxides",ylab = "mortality rate",main = "mortality
```

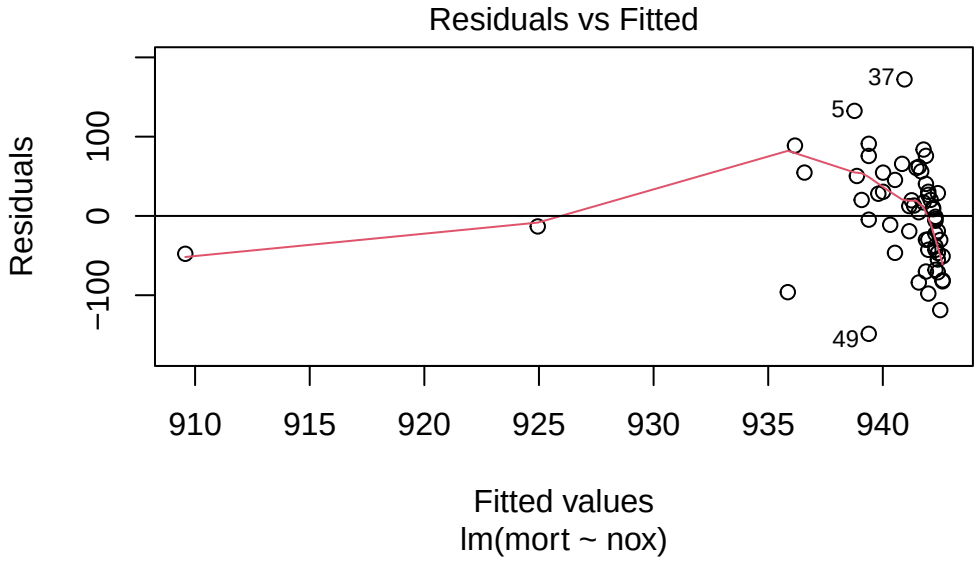



```
fit1<- lm(mort~nox,data = df)
summary(fit1)
```

```
Call:
```

```
lm(formula = mort ~ nox, data = df)

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
```

```r
###the p_value of nox is really high (>0.5),which means the linear regression doesn't fit we

plot(fit1,which=1)
abline(h = 0)
```



```
#residuals are wide spread and are not evenly scattered around 0, which means the model is no
```

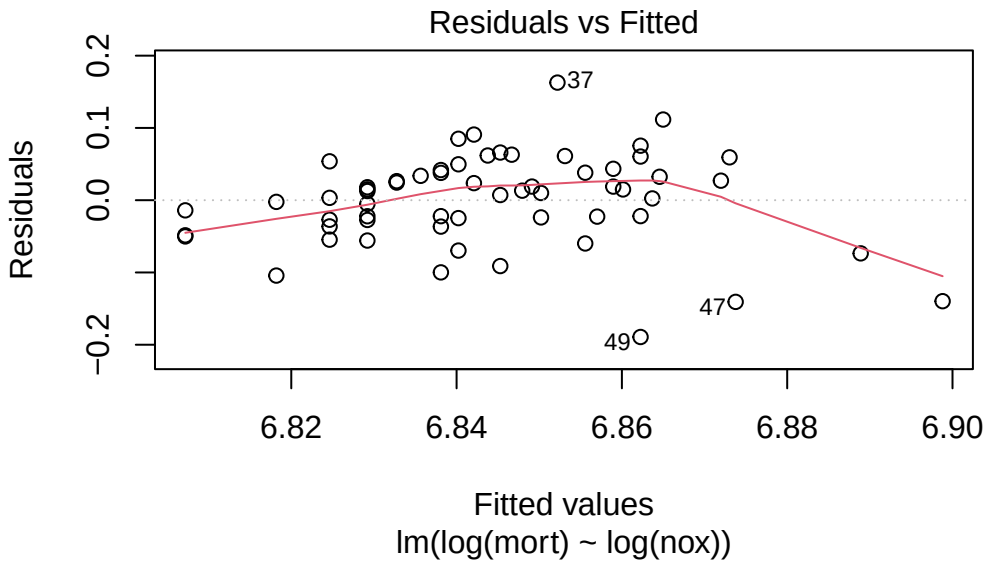
(b)

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

```
fit2 <- lm(log(mort)~ log(nox),data=df)
summary(fit2)
```

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

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(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
```

```
#the p_value for coefficient of log(nox) is pretty low(<0.05),there \n is statistical evidenc
plot(fit2,which=1)
```




```
#Residuals scattered around 0, and the red line is fairly flat.
#Although there are three outliers that may influence the trend, but generally it's good fit
```

(c)

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

```
cat("Intercept = 6.807 (baseline when log(NOx) = 0 → not very meaningful in practice since NO
```

```
Intercept = 6.807 (baseline when log(NOx) = 0 → not very meaningful in practice since NOx > 0
```

```
cat("the slope coefficient 0.0159 means that every 1% increase in x, there is about 0,0159% i
```

```
the slope coefficient 0.0159 means that every 1% increase in x, there is about 0,0159% increa
```

(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.

```r
fit3<- lm(mort~nox+so2+hc,data = df )
summary(fit3)
```

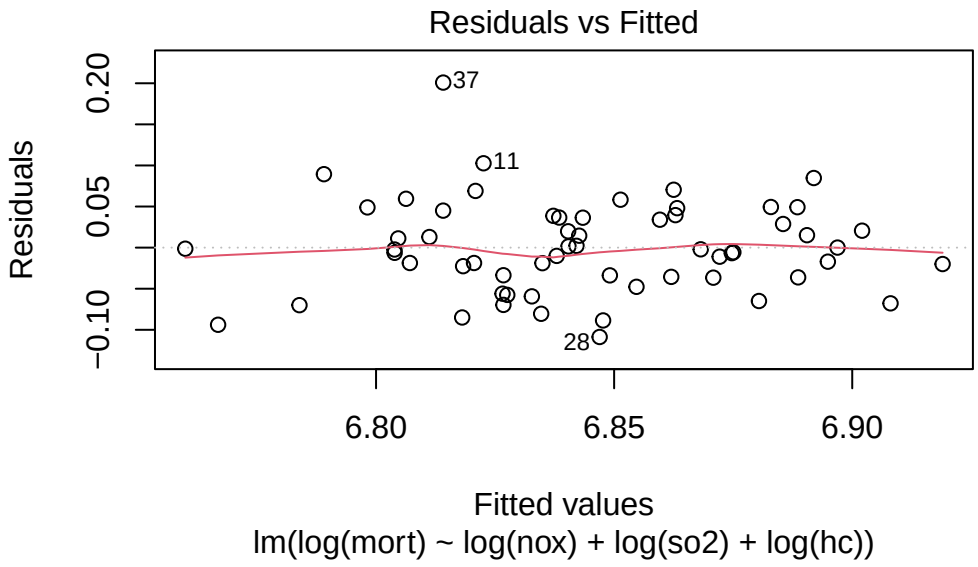
```
Call:
lm(formula = mort ~ nox + so2 + hc, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-100.020  -33.058   -5.287   38.398  171.163

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 924.1670     8.9731 102.993   <2e-16 ***
nox           2.9350     1.2668   2.317   0.0242 *
so2           0.2006     0.1728   1.161   0.2507
hc           -1.6135     0.6069  -2.659   0.0102 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 51.84 on 56 degrees of freedom
Multiple R-squared:  0.3407,    Adjusted R-squared:  0.3054
F-statistic: 9.647 on 3 and 56 DF,  p-value: 3.131e-05
```

```r
#from the result ,the linear regression is fitting nox and hc well, but doesn't fit so2 well
#so try to use log transformation
fit4<- lm(log(mort)~log(nox)+log(so2)+log(hc),data = df)
summary(fit4)
```

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

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
```

```
#interpretation of coefficient:
#log(NOx) (  = 0.060, p = 0.012): a 1% increase in NOx is associated with about a 0.06% incre
#log(SO2) (  = 0.014, p = 0.064): weak effect, marginally significant at 0.1 but not at 0.05.
#log(HC) (  = -0.061, p = 0.0045): statistically significant negative coefficient. For every
#same as the above, the linear regression after log transformation is fitting even better fo
#and it's getting better for so2, too. however, the p_value is still slightly more than 0.05
plot(fit4, which=1)
```

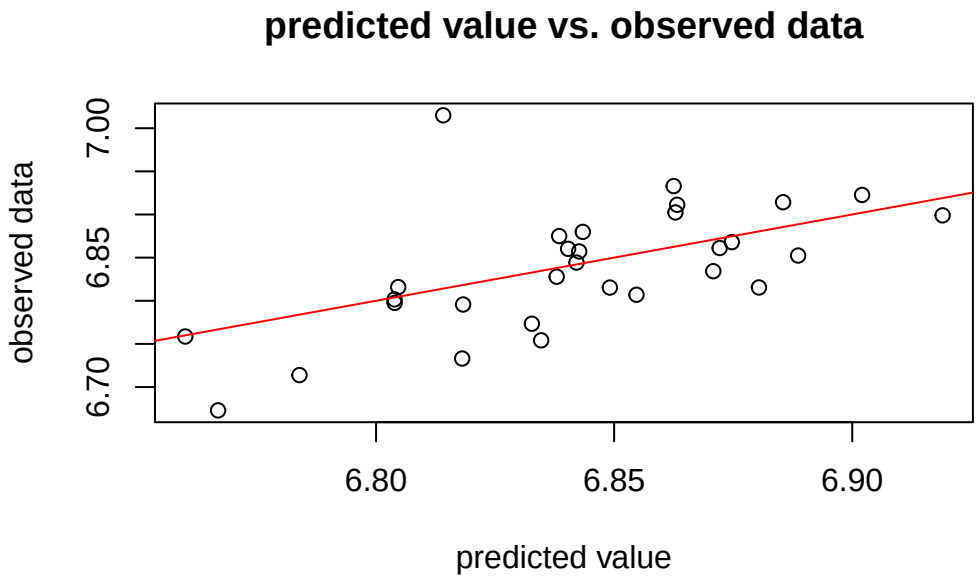



```
#the red smooth line is flat around 0,which suggests the model captures the main trend pretty
#residuals are scattered roughly evenly above and below 0
#in conclusion:Adding SO2 and HC improved the model: residuals look more balanced and the re
```

(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.

```
n<-nrow(df)
fit_data<- df[1:n/2,]
pre_data<-df[(n/2+1):n,]
pre<- predict(fit4,pre_data)
plot(pre,log(pre_data$mort),xlab="predicted value",ylab="observed data",main="predicted value
abline(0, 1, col = "red")
```




```
#Predictions aligned reasonably well with observed values, though some scatter remained.
```

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.

```
df <- read.csv("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/Earnings/data/
df <- subset(df, is.finite(earnk) & earnk > 0 & is.finite(height) & !is.na(male))
df$male <- factor(df$male, levels = c(0, 1),
                  labels = c("female", "male"))
m_raw <- lm(earnk ~ height + male, data = df)
m_log <- lm(log(earnk) ~ height + male, data = df)

summary(m_raw)
```

```
Call:
lm(formula = earnk ~ height + male, data = df)

Residuals:
   Min     1Q Median     3Q    Max
-31.46 -12.19  -3.92   6.08 368.12

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -18.5344    12.6635  -1.464  0.14349
height        0.5762     0.1959   2.942  0.00331 **
malemale      8.9325     1.5430   5.789 8.48e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.71 on 1626 degrees of freedom
Multiple R-squared:  0.07465,   Adjusted R-squared:  0.07352
F-statistic: 65.59 on 2 and 1626 DF,  p-value: < 2.2e-16
```

```
summary(m_log)
```

```
Call:
lm(formula = log(earnk) ~ height + male, data = df)

Residuals:
```

```
    Min      1Q  Median      3Q     Max
-4.2749 -0.4243  0.1140  0.5767  2.8353

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.06858    0.50624   2.111  0.03494 *
height       0.02383    0.00783   3.044  0.00237 **
malemale     0.37151    0.06168   6.023 2.11e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8679 on 1626 degrees of freedom
Multiple R-squared:  0.08007,   Adjusted R-squared:  0.07893
F-statistic: 70.76 on 2 and 1626 DF,  p-value: < 2.2e-16
```

```r
b_h   <- coef(m_log)["height"]
b_m <- coef(m_log)[grep("^male", names(coef(m_log)))]
b_m   <- if (!is.na(b_m)) b_m else coef(m_log)[grep("^male", names(coef(m_log)))[1]]

inch_pct   <- 100 * b_h
male_ratio <- exp(b_m)
male_pct   <- 100 * (male_ratio - 1)
cat(sprintf("Log model: +1 inch  %.2f%% higher earnings; male vs female  %.1f%% higher (sam
            inch_pct, male_pct))
```

```
Log model: +1 inch  2.38% higher earnings; male vs female  45.0% higher (same height).
```

```r
set.seed(615)
k <- 10
n <- nrow(df)
folds <- sample(rep(1:k, length.out = n))

pred_raw  <- rep(NA_real_, n)
pred_logB <- rep(NA_real_, n)
for (i in 1:k) {
  tr <- df[folds != i, ]
  te <- df[folds == i, ]
  mA <- lm(earnk ~ height + male, data = tr)
  pred_raw[folds == i] <- predict(mA, newdata = te)
```

```
  mB <- lm(log(earnk) ~ height + male, data = tr)
  smear <- mean(exp(residuals(mB)), na.rm = TRUE)
  muhat <- predict(mB, newdata = te)
  pred_logB[folds == i] <- exp(muhat) * smear
}


rmse <- function(a, b) sqrt(mean((a - b)^2))
mae  <- function(a, b) mean(abs(a - b))

cv_table <- data.frame(
  Model = c("Raw: earnk ~ height + male",
            "Log: log(earnk) ~ height + male (back-transformed w/ Duan smearing)"),
  RMSE  = c(rmse(df$earnk, pred_raw),  rmse(df$earnk, pred_logB)),
  MAE   = c(mae(df$earnk,  pred_raw),  mae(df$earnk,  pred_logB))
)

cat("\n=== 10-fold CV (errors on ORIGINAL earnk scale, in $1000s) ===\n")
```

```
=== 10-fold CV (errors on ORIGINAL earnk scale, in $1000s) ===
```

```
print(cv_table)
```

```
                                                                  Model     RMSE
1                                          Raw: earnk ~ height + male 21.73155
2 Log: log(earnk) ~ height + male (back-transformed w/ Duan smearing) 21.72445
       MAE
1 13.26874
2 13.26969
```

```
#Prediction accuracy: Both models perform almost identically on the original scale (RMSE/MAE
```

(b)

Compare models from other exercises in this chapter.

```r
df_poll <- read.csv("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/Pollutio
df_poll <- subset(df_poll, is.finite(mort) & mort > 0 & is.finite(nox)  & is.finite(so2) & is
set.seed(615)
k <- 10
n <- nrow(df_poll)
folds <- sample(rep(1:k, length.out = n))
pred_raw  <- rep(NA_real_, n)
pred_logB <- rep(NA_real_, n)
for (i in 1:k) {
  tr <- df_poll[folds != i, , drop = FALSE]
  te <- df_poll[folds == i, , drop = FALSE]
  fit_raw <- lm(mort ~ nox + so2 + hc, data = tr)
  pred_raw[folds == i] <- predict(fit_raw, newdata = te)
  fit_log <- lm(log(mort) ~ log(nox) + log(so2) + log(hc), data = tr)
  smear <- mean(exp(residuals(fit_log)), na.rm = TRUE)
  muhat <- predict(fit_log, newdata = te)
  pred_logB[folds == i] <- exp(muhat) * smear
}


rmse_raw  <- sqrt(mean((df_poll$mort - pred_raw)^2))
mae_raw   <- mean(abs(df_poll$mort - pred_raw))

rmse_logB <- sqrt(mean((df_poll$mort - pred_logB)^2))
mae_logB  <- mean(abs(df_poll$mort - pred_logB))

data.frame(
  Model = c("mort ~ nox + so2 + hc",
            "log(mort) ~ log(nox) + log(so2) + log(hc) [back-transformed]"),
  RMSE  = c(rmse_raw,  rmse_logB),
  MAE   = c(mae_raw,   mae_logB)
)
```

```
                                                  Model     RMSE
1                                  mort ~ nox + so2 + hc 57.29587
2 log(mort) ~ log(nox) + log(so2) + log(hc) [back-transformed] 65.55690
       MAE
1 45.82344
2 46.86143
```

```
#For the Pollution data, cross-validation indicates that the raw outcome model predicts morta
```

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. ln(weight_kg)= -5.521 + ln(height_cm) 1.96*sigma = ln(1.1) -> sigma = 0.0486 ### (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?

$$R^2 = 1 - \frac{\text{Var(residuals)}}{\text{Var(ln(weight))}}.$$

$$R^2 = 1 - \frac{0.00236}{0.04} \approx 1 - 0.059 \approx 0.94.$$

```
cat("the value of R^2 is 0.94, which is very high. It suggests the relationship between weig
```

```
the value of R^2 is 0.94, which is very high. It suggests the relationship between weight an
```

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: it keeps the original unit in dollars, one can easily see the direct difference between the two groups. it's also easier to interpret the difference in dollar terms like "How many more dollars did the Democrat raise than the Republican?" disadvantage: Doesn't scale: \$100k difference is huge in a small race but trivial in a big-money Senate race. Distribution may be very skewed if some candidates raise much more money overall. Not comparable across districts with different fundraising levels. ### (b) The ratio, D_i/R_i advantage:Captures relative fundraising strength: a Democrat who raises twice as much as the opponent has the same ratio whether it's \$20k vs \$10k or \$2M vs \$1M. Dimensionless (no units), so it's more comparable across districts. disadvantage:Ratios can be unstable when R^i is small (division by tiny numbers $\rightarrow$ huge values).Highly skewed distribution (e.g., ratios of 10, 20, 100). Harder to interpret directly in terms of vote share. ### (c) The difference on the logarithmic scale, $\log D_i - \log R_i$
advantage:similar to D_i/R_i , which compresses extreme ratios. Symmetric: if Democrats raise half as much, value = $-$log2; if they raise twice as much, value = +log2. Works well in regression because it stabilizes variance and makes effect sizes more interpretable. disadvantage:it;s harder to interpret the number directly. there is no unit. ### (d) The relative proportion, $D_i/(D_i + R_i)$. advantage:Normalized between 0 and 1 (proportion of total funds raised by Democrats). Easy to interpret: 0.6 means Democrats raised 60% of the money. Comparable across races of any size. disadvantage:Nonlinear effects: a change from 0.4 to 0.5 may matter differently than 0.8 to 0.9. Skewed distribution if many races are lopsided. Still undefined if both candidates raise zero.

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. $\beta = 0.3$: for every 1% increase in the average price of cigarettes, there is a 0.3% increase in the quantity purchased. this is weird because generally the higher the price, the less the purchase

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(rstanarm)
```

```
Loading required package: Rcpp
```

```
This is rstanarm version 2.32.1
```

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

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

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

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

```
library(loo)
```

```
This is loo version 2.8.0
```

```
- 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 'co
```

```
options(mc.cores = parallel::detectCores())
set.seed(123)


df <- read.csv("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/Beauty/data/be


df$female     <- factor(df$female)
df$minority   <- factor(df$minority)
df$nonenglish <- factor(df$nonenglish)
df$lower      <- factor(df$lower)
df$course_id  <- factor(df$course_id)
df <- subset(df, is.finite(eval) & is.finite(beauty) & is.finite(age) & age > 0)
df$log_age <- log(df$age)
```

```
m1 <- stan_glm(eval ~ beauty, data=df, family=gaussian())
m2 <- stan_glm(eval ~ beauty + female, data=df, family=gaussian())
m3 <- stan_glm(eval ~ beauty*(female + minority + nonenglish + lower) + age,
               data=df, family=gaussian())
```

```
m4 <- stan_lmer(eval ~ beauty + female + minority + nonenglish + lower + log_age +
                  (1 | course_id),
                data=df)

mods <- list(m1=m1, m2=m2, m3=m3, m4=m4)


K <- 10
folds <- loo::kfold_split_random(K = K, N = nrow(df))

kfs <- lapply(mods, function(fit) loo::kfold(fit, folds = folds, save_fits = FALSE))
```

```
Fitting K = 10 models distributed over 10 cores
```

```
Fitting K = 10 models distributed over 10 cores
Fitting K = 10 models distributed over 10 cores
Fitting K = 10 models distributed over 10 cores
```

```
stopifnot(all(sapply(kfs, function(x) inherits(x, "kfold"))))
cmp <- loo::loo_compare(kfs)
print(cmp)
#Because of the highest elpd_diff, ther result shows that m4(:eval = 4.7 + 0.1*beauty -0.2*f
```

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.

code: pred <- predict(fit_4, newdata = new_trees, se.fit = TRUE) estimate<- mean(pred$fit) $se_avg <- -sqrt(mean(pred$se.fit^2))$
list(estimate = estimate, se = se_avg)