Analysis of mortality rates and various environmental factors

Yifu Dong

Oct 23, 2018

Data analysis

Analysis of mortality rates and various environmental factors

The folder `pollution` contains mortality rates and various environmental factors from 60 U.S. metropolitan areas from McDonald, G.C. and Schwing, R.C. (1973) 'Instabilities of regression estimates relating air pollution to mortality', Technometrics, vol.15, 463-482.

Variables, in order:

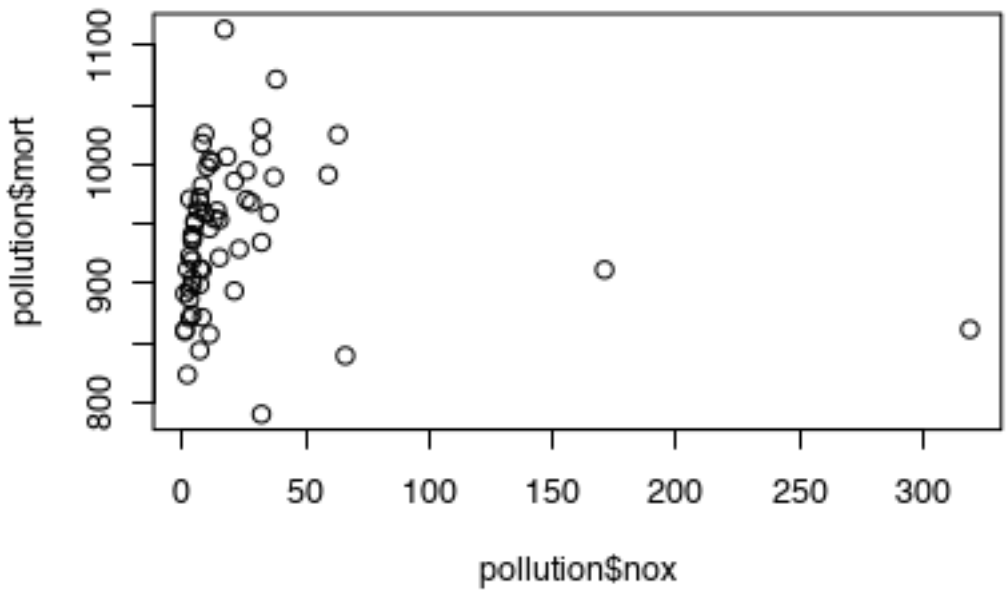
- PREC Average annual precipitation in inches
- JANT Average January temperature in degrees F
- JULT Same for July
- OVR65 % of 1960 SMSA population aged 65 or older
- POPN Average household size
- EDUC Median school years completed by those over 22
- HOUS % of housing units which are sound & with all facilities
- DENS Population per sq. mile in urbanized areas, 1960
- NONW % non-white population in urbanized areas, 1960
- WWDRK % employed in white collar occupations
- POOR % of families with income < $3000
- HC Relative hydrocarbon pollution potential
- NOX Same for nitric oxides
- SO@ Same for sulphur dioxide
- HUMID Annual average % relative humidity at 1pm
- MORT Total age-adjusted mortality rate per 100,000

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 transformations in regression.

```
gelman_dir   <- "http://www.stat.columbia.edu/~gelman/arm/examples/"
pollution    <- read.dta (paste0(gelman_dir,"pollution/pollution.dta"))
```

1. 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.

```
plot(pollution$nox,pollution$mort)
```

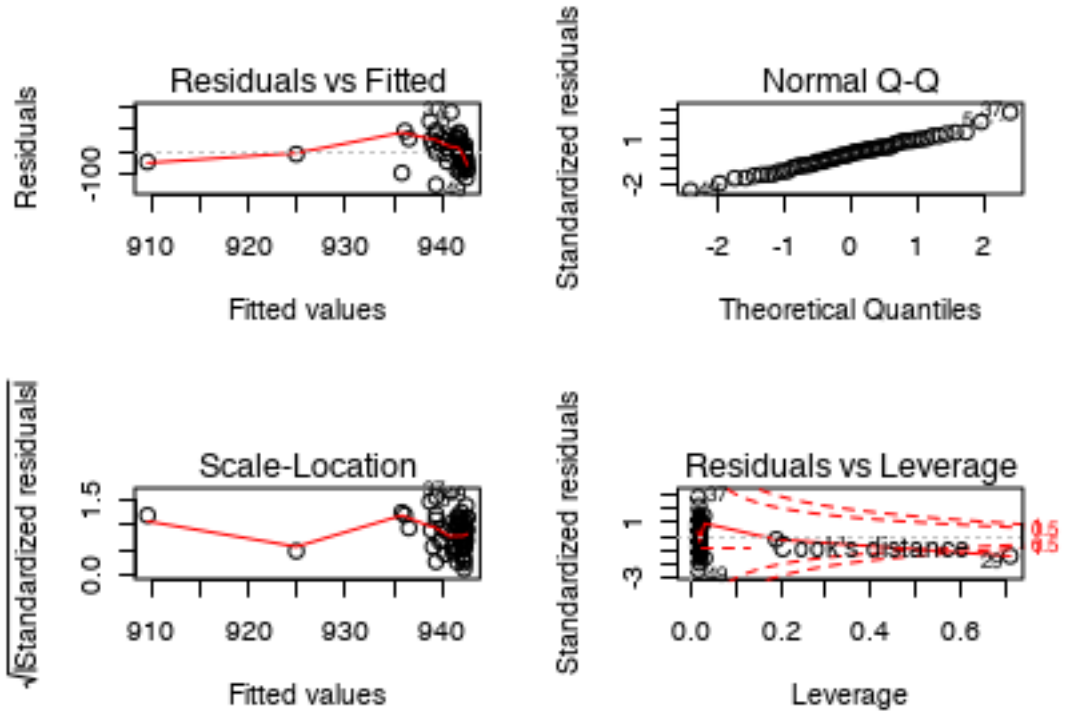


We can know from the figure above that linear regression modelmay be a good model for there data.

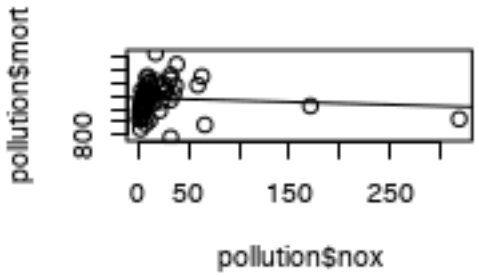
```
regout2 <- lm(pollution$mort~pollution$nox, data=pollution)
display(regout2)
```

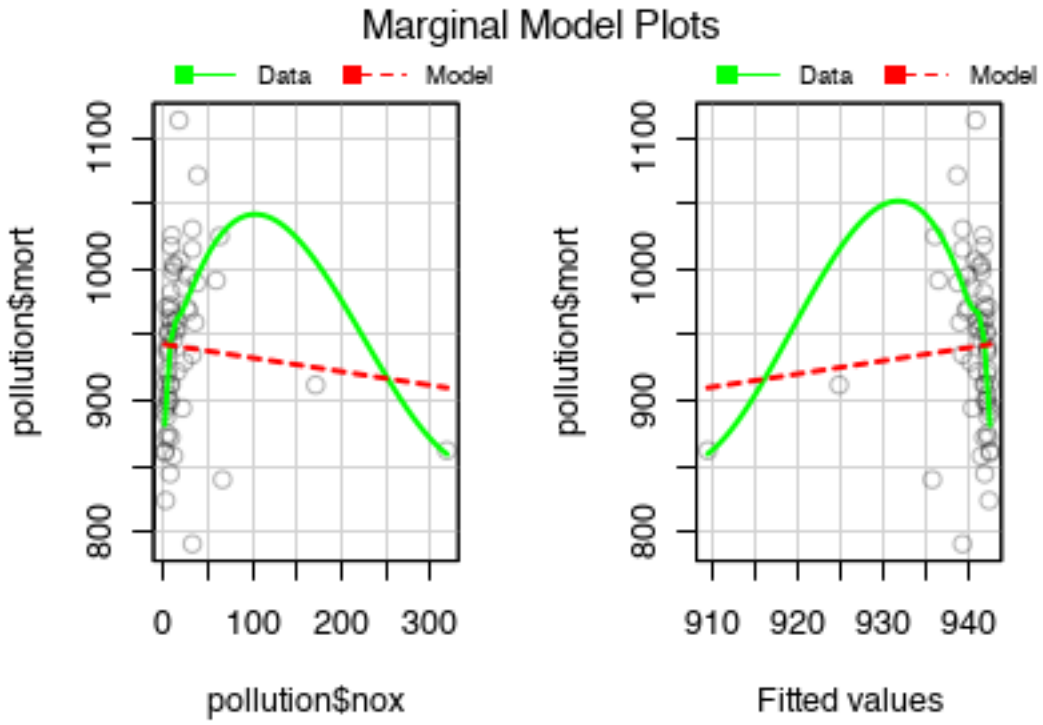
```
## lm(formula = pollution$mort ~ pollution$nox, data = pollution)
##                coef.est coef.se
## (Intercept)    942.71     9.00
## pollution$nox   -0.10     0.18
## ---
## n = 60, k = 2
## residual sd = 62.55, R-Squared = 0.01
```

```
gelman_dir   <- "http://www.stat.columbia.edu/~gelman/arm/examples/"
pollution    <- read.dta (paste0(gelman_dir,"pollution/pollution.dta"))
pollution_clean <- pollution
par(mfrow=c(2,2))
plot(regout2)
```




```
plot(pollution$nox,pollution$mort)
regout2 <- lm(pollution$mort~pollution$nox, data=pollution)
abline(regout2)
#overall fit
marginalModelPlots(regout2,col=rgb(0,0,0,alpha=0.3),col.line = c("green","red"))
```





From the plots of residuals and the relation between this two variables, we cannot say it fits very well. Residuals suffer from heteroschedasticity.**It seems that outliers exist.But we cannot remove them straightly.**

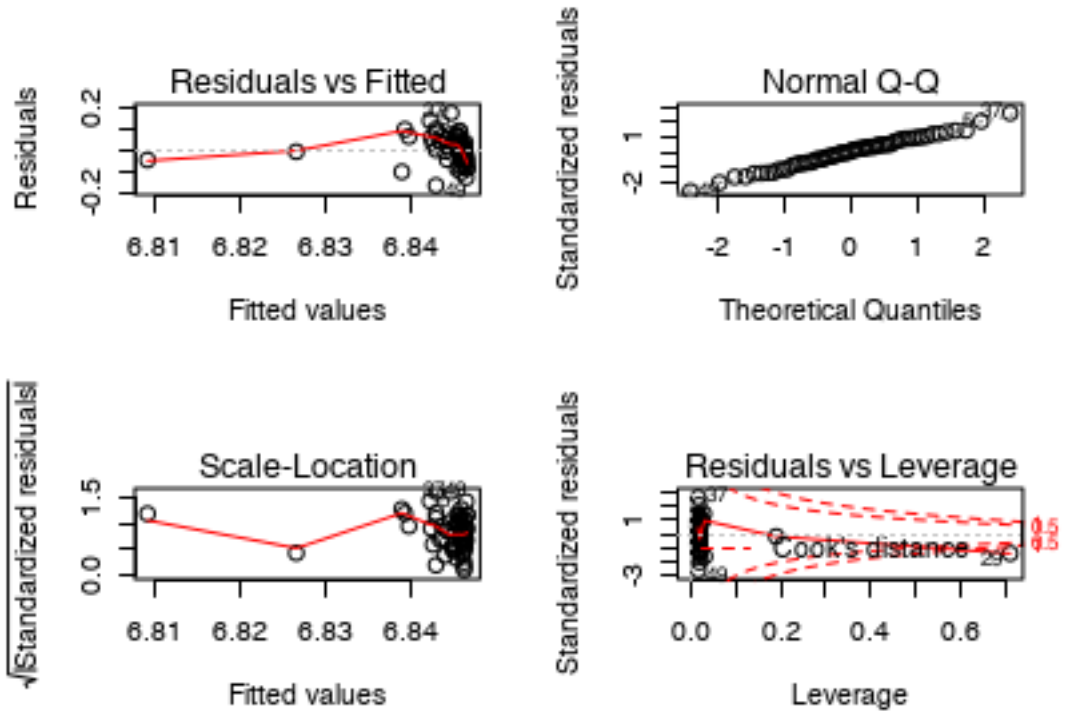
2. Find an appropriate transformation that will result in data more appropriate for linear regression. Fit a regression to the transformed data and evaluate the new residual plot.

Actually, the R^square of the model above is only 0.01, which means the relation of two variables is posibly not linear. So we use log to see what happens.

```
regout2_2 <- lm(log(pollution_clean$mort) ~ (pollution_clean$nox), data=pollution_clean)
display(regout2_2)
```

```
## lm(formula = log(pollution_clean$mort) ~ (pollution_clean$nox),
##     data = pollution_clean)
##                     coef.est coef.se
## (Intercept)         6.85     0.01
## pollution_clean$nox 0.00     0.00
## ---
## n = 60, k = 2
## residual sd = 0.07, R-Squared = 0.01
```

```
par(mfrow=c(2,2))
plot(regout2_2)
```



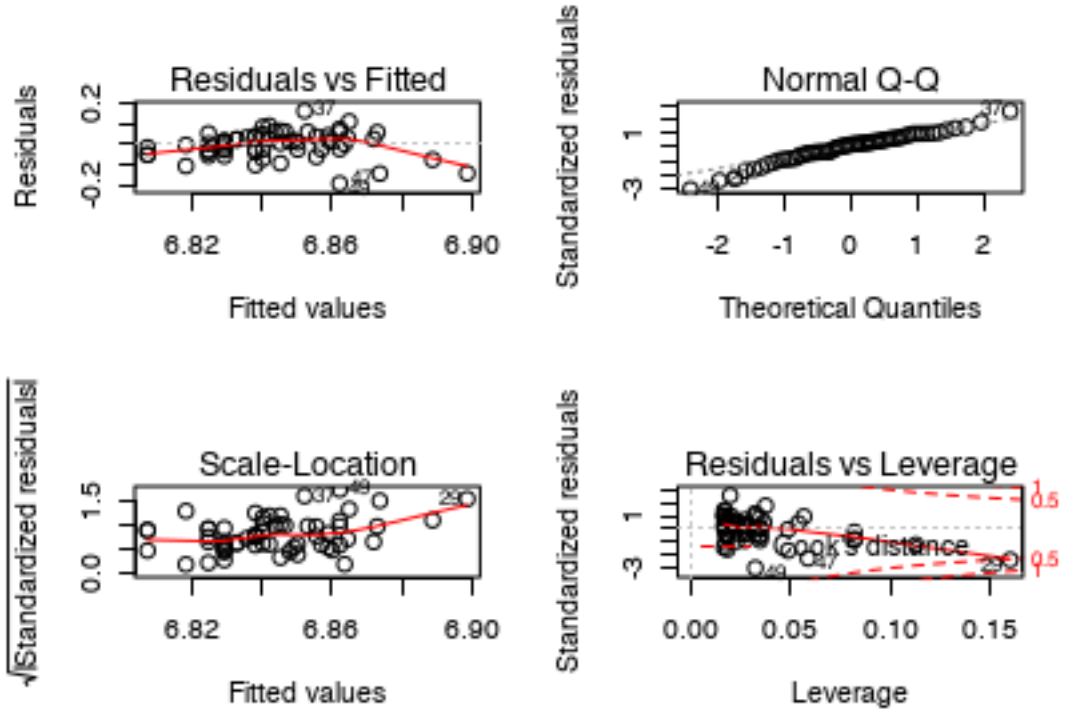
The R-Squared is still 0.01, without and improvement.

```
regout2_2 <- lm(log(pollution_clean$mort) ~ log(pollution_clean$nox), data=pollution_clean)
display(regout2_2)
```

```
## lm(formula = log(pollution_clean$mort) ~ log(pollution_clean$nox),
##     data = pollution_clean)
##                           coef.est coef.se
## (Intercept)               6.81     0.02
## log(pollution_clean$nox) 0.02     0.01
```

```
## ---
## n = 60, k = 2
## residual sd = 0.06, R-Squared = 0.08
```

```
par(mfrow=c(2,2))
plot(regout2_2)
```

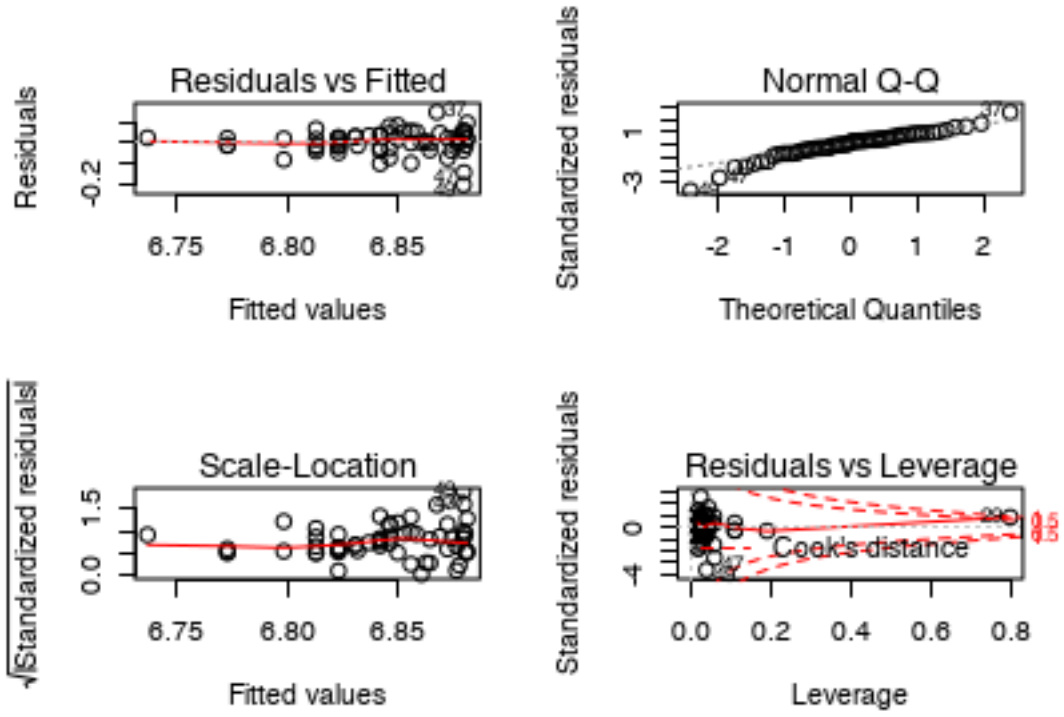


From the figures above, the log model fits better, but residuals still suffer from heteroschedasticity. Then we try to regress log(nox) on mort, the result is worse. Now we try another model:

```
regout2_2 <- lm(log(pollution_clean$mort) ~ log(pollution_clean$nox)+pollution_clean$nox, data=pollutio
display(regout2_2)
```

```
## lm(formula = log(pollution_clean$mort) ~ log(pollution_clean$nox) +
##     pollution_clean$nox, data = pollution_clean)
##                           coef.est coef.se
## (Intercept)               6.77     0.02
## log(pollution_clean$nox)  0.04     0.01
## pollution_clean$nox       0.00     0.00
## ---
## n = 60, k = 3
## residual sd = 0.06, R-Squared = 0.24
```

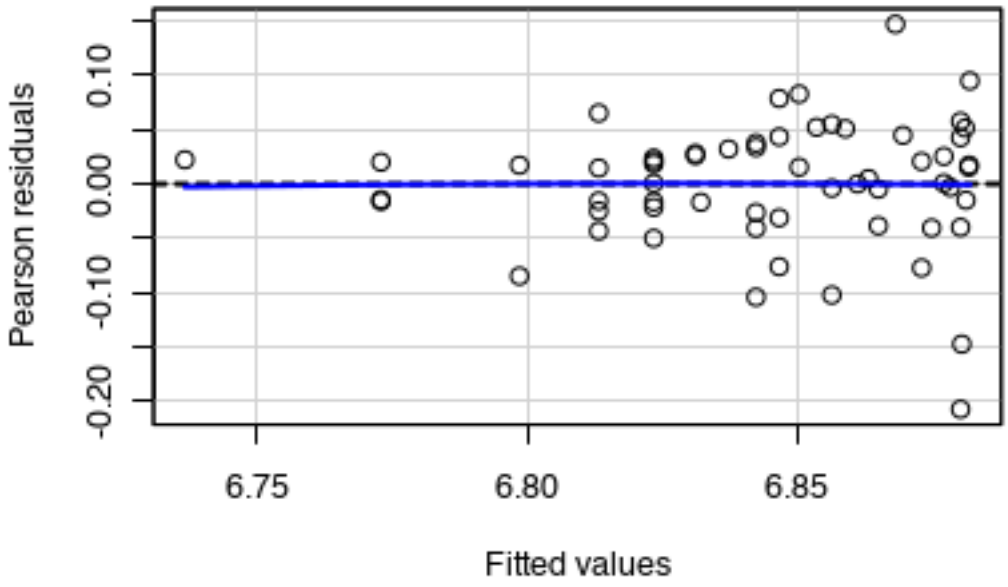
```
par(mfrow=c(2,2))
plot(regout2_2)
```



This model fits much better, although it's still not significant enough for this two variables.

Now we normalize the variables, however,when we normalize the data, it's hard to use log transformation since NaNs exist easier. So we don't choose to normalize data.

```
residualPlots(regout2_2, terms= ~ 1, fitted=TRUE)
```




```
##            Test stat Pr(>|Test stat|)
## Tukey test   -0.0963           0.9233
```

Tukey test is not significant, which means that this model is actually still not good enough.

Residuals still suffer from heteroschedasticity.

3. Interpret the slope coefficient from the model you chose in 2.

Intercept: The average morality rate when NO equals 0 is $exp(6.77) =871.3119 $ log(nox):For each 1 difference in nitric oxide , the predicted difference in morality rate is +0.04%

4. Construct 99% confidence interval for slope coefficient from the model you chose in 2 and interpret them.

```
confint(regout2_2,'log(pollution_clean$nox)',level = 0.99)
```

```
##                                0.5 %     99.5 %
## log(pollution_clean$nox) 0.01378255 0.06240595
```

This means that if we fit the model and calculate the slope over and over again, 99% true value of the slope coefficient will be in the interval (0.01378255, 0.06240595)

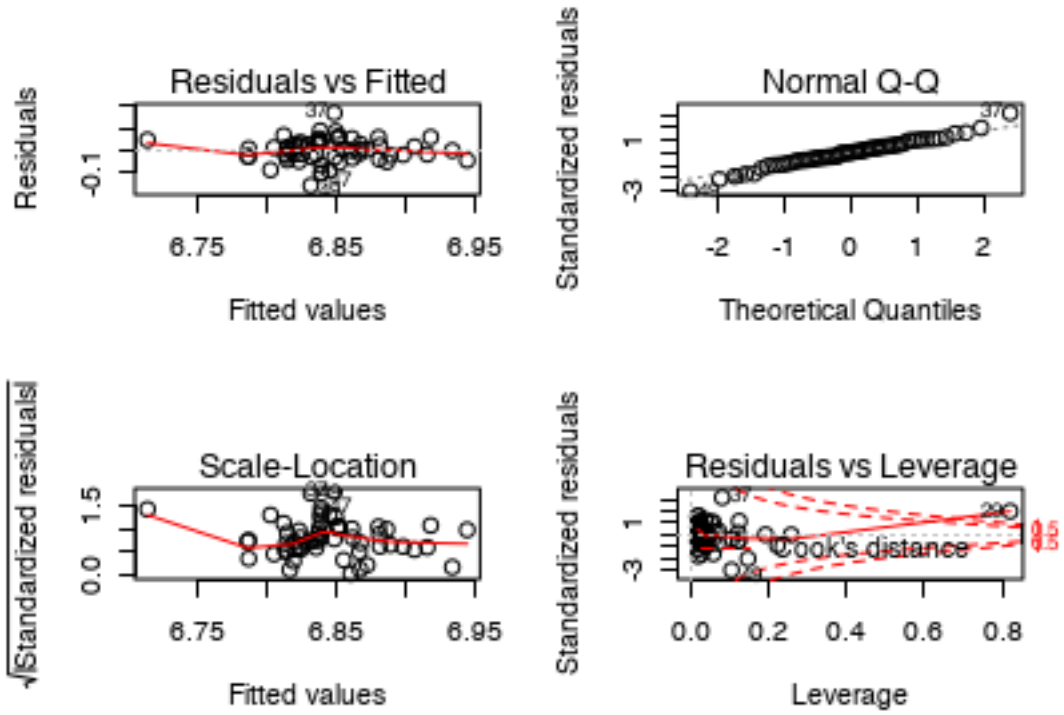
5. Now fit a model predicting mortality rate using levels of nitric oxides, sulfur dioxide, and hydrocarbons as inputs. Use appropriate transformations when helpful. Plot the fitted regression model and interpret the coefficients.

```
#normalize
so2n<- (pollution_clean$so2 - mean(pollution_clean$so2)) / (2*sd(pollution_clean$so2))
hcn<- (pollution_clean$hc - mean(pollution_clean$hc)) / (2*sd(pollution_clean$hc))

#regression
regout2_5 <- lm(log(pollution_clean$mort)~log(pollution_clean$nox)+so2n+hcn, data=pollution_clean)
display(regout2_5)
```

```
## lm(formula = log(pollution_clean$mort) ~ log(pollution_clean$nox) +
##     so2n + hcn, data = pollution_clean)
##                           coef.est coef.se
## (Intercept)                6.79     0.03
## log(pollution_clean$nox)   0.02     0.01
## so2n                       0.04     0.02
## hcn                       -0.07     0.02
## ---
## n = 60, k = 4
## residual sd = 0.06, R-Squared = 0.34
```

```
par(mfrow=c(2,2))
plot(regout2_5)
```



Before the model above, we tried not to normalize the predictors and tried to add "log" relatively. Finally we found that the model above fits best. So we choose this model.

Interpretation:

Intercept: The mortality rate for an individual exposed to average levels of nitric oxides, sulfur dioxide, and hydrocarbons is $exp(6.73) = 837.1473$

log(pollution_clean$nox): 1 standard deviation difference for nitric oxides corresponds to a mortality rate 5% higher.

so2n: 1 standard deviation difference for sulfur dioxide corresponds to $exp(0.03) = 1.030455$ increase in mortality rate.

hcn: 1 standard deviation difference in hydrocarbons corresponds to a mortality rate $exp(-0.10) = 0.948374$ times lower, which is a decrease of about 6%.

6. 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 4, so this is not really cross-validation, but it gives a sense of how the steps of cross-validation can be implemented.)

```
#divide the dataset into 2 part: train dataset and predict dataset
train <-  pollution_clean[1:(nrow(pollution_clean)/2),]
pred <- pollution_clean[((nrow(pollution_clean)/2)+1):nrow(pollution_clean),]

#normalize choosing the data from training dataset.
so2n<- (train$so2 - mean(train$so2)) / (2*sd(train$so2))
hcn<- (train$hc - mean(train$hc)) / (2*sd(train$hc))
regout2_6 <- lm(log(train$mort)~log(train$nox)+so2n+hcn, data=train)
display(regout2_6)
```

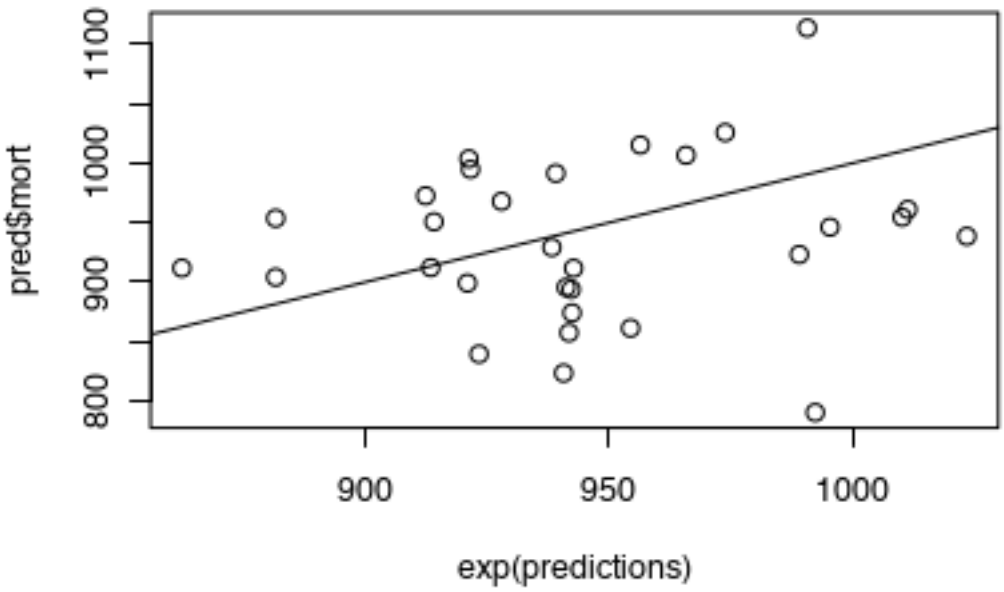
```
## lm(formula = log(train$mort) ~ log(train$nox) + so2n + hcn, data = train)
##                coef.est coef.se
## (Intercept)     6.78     0.03
## log(train$nox)  0.03     0.01
## so2n            0.02     0.03
## hcn            -0.08     0.02
## ---
## n = 30, k = 4
## residual sd = 0.05, R-Squared = 0.48
```

```
#predict
predictions <- predict(regout2_6, pred)
cbind(predictions=exp(predictions), observed=pred$mort)
```

```
##    predictions observed
## 31    965.7792 1006.490
## 32    954.4248  861.439
## 33    938.3258  929.150
## 34    941.8281  857.622
## 35   1011.1601  961.009
## 36    988.9615  923.234
## 37    990.5861 1113.156
## 38    921.6717  994.648
## 39    956.4211 1015.023
## 40    939.1293  991.290
## 41    942.2755  893.991
## 42   1023.2452  938.500
## 43    995.2379  946.185
## 44    973.7902 1025.502
```

```
## 45    942.4432  874.281
## 46    881.8057  953.560
## 47    923.4241  839.709
## 48    942.7623  911.701
## 49    992.2236  790.733
## 50    921.0422  899.264
## 51    881.8057  904.155
## 52    914.1987  950.672
## 53    912.4943  972.464
## 54    913.5440  912.202
## 55    928.0109  967.803
## 56    940.7267  823.764
## 57    921.3416 1003.502
## 58    941.2520  895.696
## 59    862.6143  911.817
## 60   1010.0055  954.442
```

```
plot(exp(predictions), pred$mort)
abline(a=0, b=1)
```




```
# compute RMSE
sqrt(mean((pred$mort-exp(predictions))^2))
```

```
## [1] 72.86453
```

```
#compute R Squared
summary(regout2_6)["r.squared"]
```

```
## $r.squared
## [1] 0.4788315
```