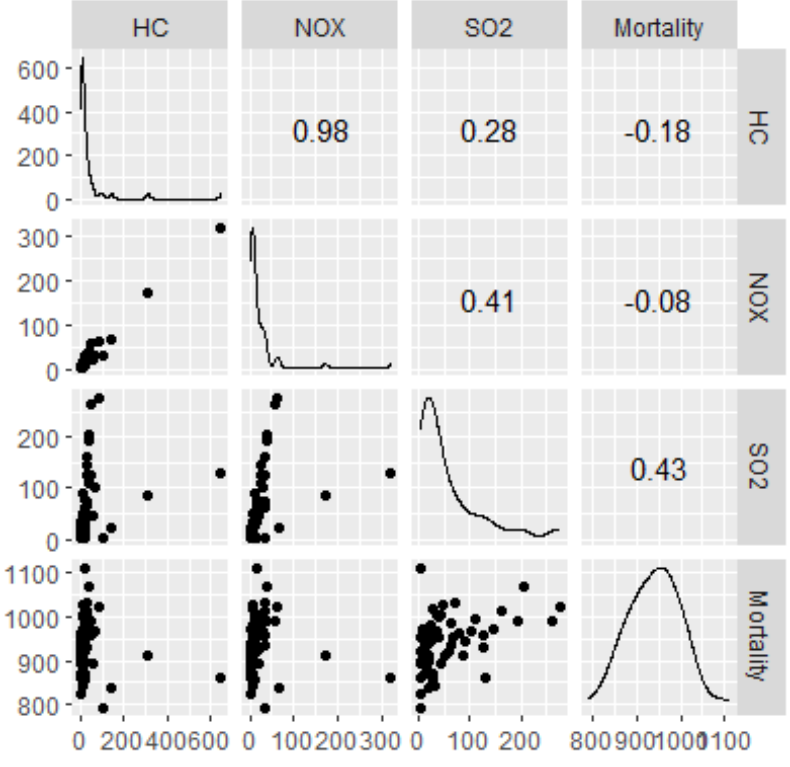
Code Supplement for Case Study 2

Dean Gladish and Nate Isbell

May 8, 2018

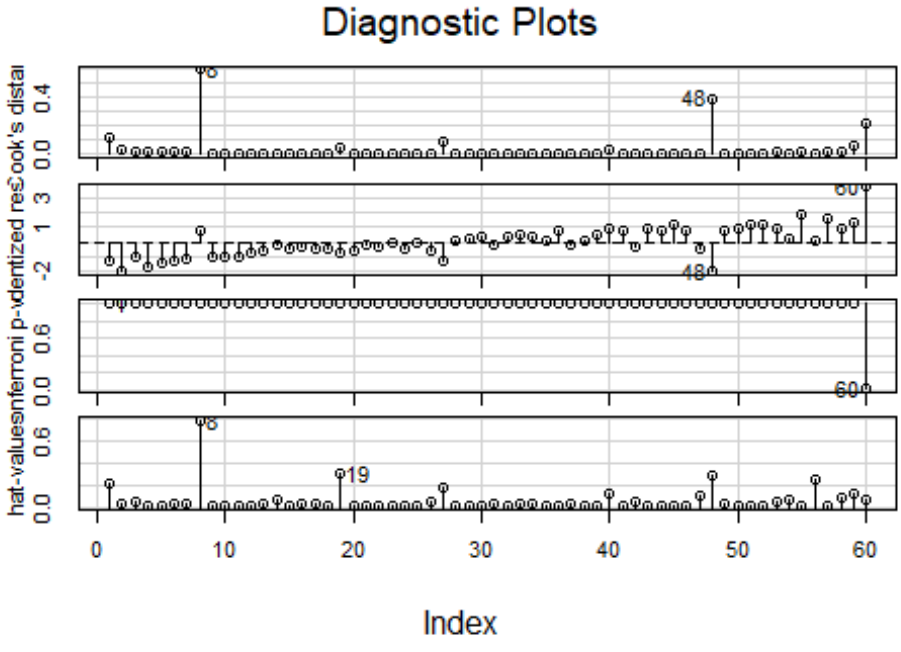
```
library(Sleuth3)
library(GGally)
library(broom)
library(car)
library(ggformula)


data(ex1217, package = "Sleuth3")

#  Constructs a scatterplot matrix
ggscatmat(ex1217, columns = c(15:17, 2))
```



```r
#  A basic SLR model.
model1 <- lm(Mortality ~ HC + NOX + SO2, data = ex1217)

#  This puts our information into a more readable format.
tidy(model1)
```

```
##          term    estimate std.error  statistic      p.value
## 1 (Intercept) 924.1631054 8.9720848 103.004276 1.557282e-65
## 2          HC  -1.6132989 0.6068211  -2.658607 1.021017e-02
## 3         NOX   2.9345822 1.2666352   2.316833 2.419511e-02
## 4         SO2   0.2006535 0.1727938   1.161231 2.504740e-01
```

```r
#  This portion allows us to see Cook's distance (which governs
influence),
#  Studentized residuals (which indicate deviation from the model
itself),
#  the Bonferroni p-value, and the hat-values.
influenceIndexPlot(model1)
```

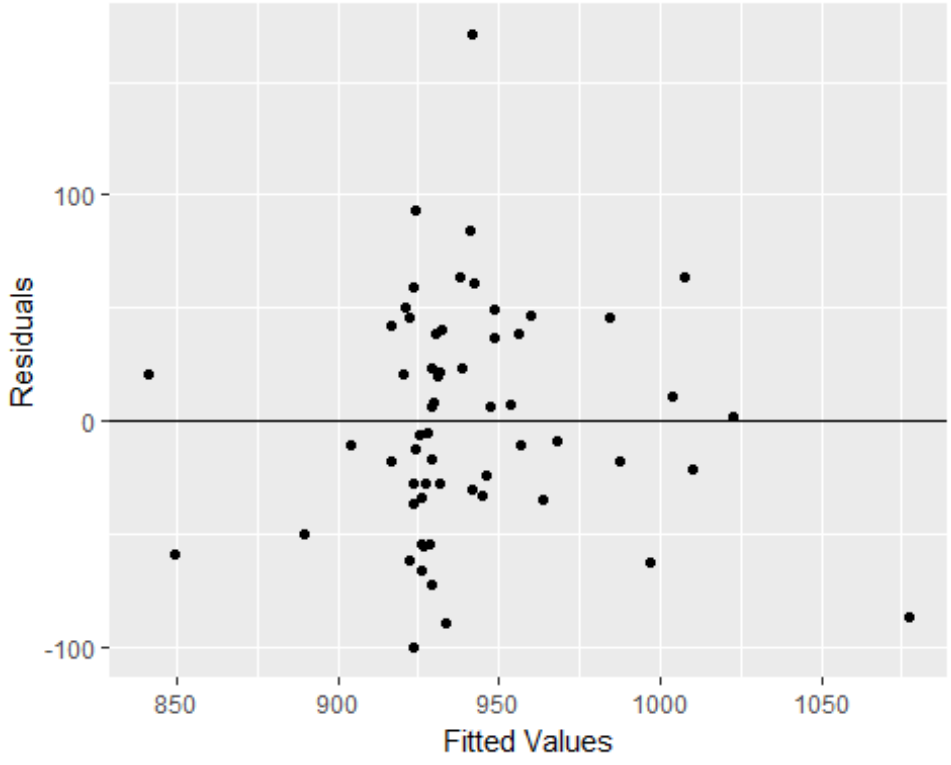


We determined that the cut-off point for leverage here is 2*((3+1)/60) ~ 0.133333. As shown on the plots, case 8 (LA) is influential as it is close to 1 in terms of Cook's Distance. This case (8) also has high leverage as indicated on the hat-values plot. This means that we should exclude case 8 as well as any other outliers that we come across.

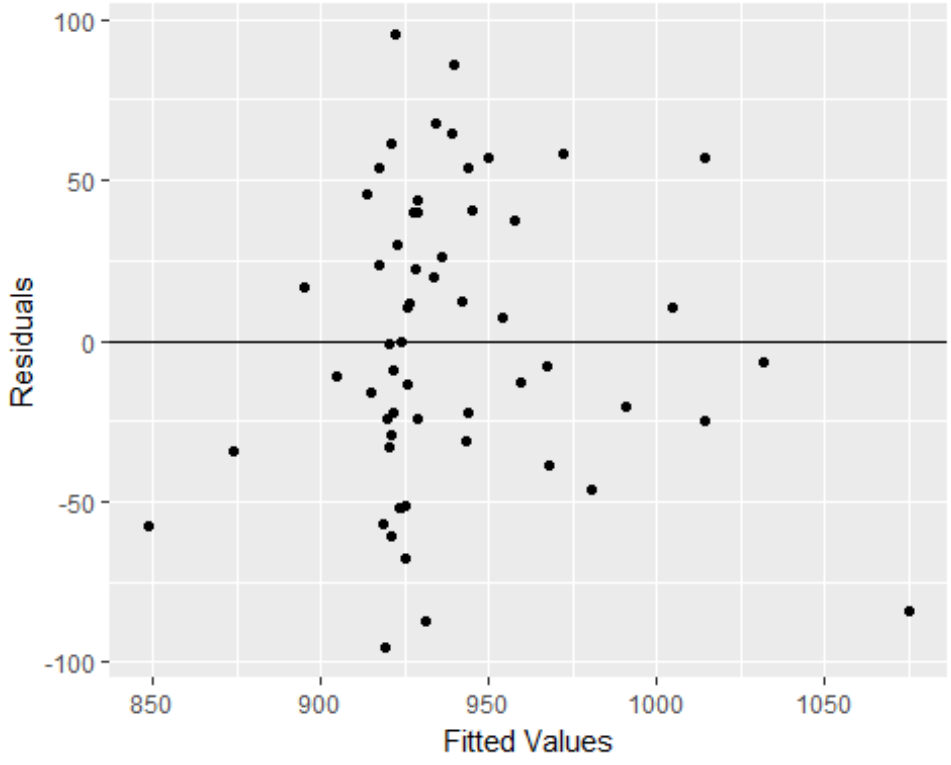
City 60 does not have a high leverage value but it deviates from the model as indicated by its high residual value. Thus we decided to remove cases 8 and 60 (New Orleans).

The following code is where we fit a new model that excludes these two cities.

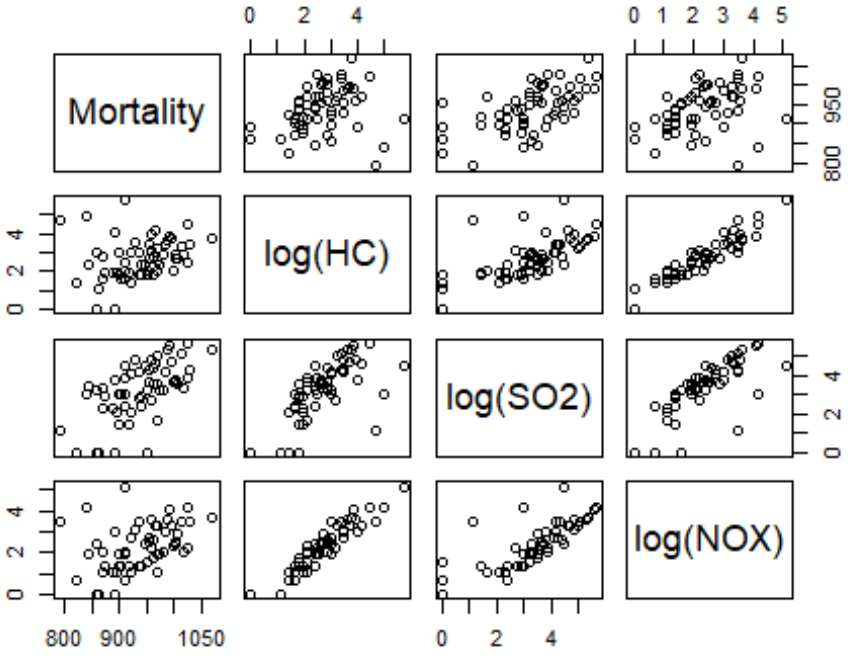
```
#  Fit a new model without case 8.
model2 <- lm(Mortality ~ HC + NOX + SO2, data = ex1217, subset = -c(8,
60))

#  The following code gives us information about the model used to
#  generate residual plots below.
model1_aug <- augment(model1)
model2_aug <- augment(model2)

# Residual plot for first model
gf_point(.resid ~ .fitted, data = model1_aug) %>%
  gf_hline(yintercept = 0, color = "blue", linetype = 2) %>%
  gf_labs(x = "Fitted Values", y = "Residuals")
```




```
# Residual plot after removals.
gf_point(.resid ~ .fitted, data = model2_aug) %>%
  gf_hline(yintercept = 0, color = "blue", linetype = 2) %>%
  gf_labs(x = "Fitted Values", y = "Residuals")
```




```
# To assess potential multicollinearity we created a correlation
matrix.
pairs(Mortality ~ log(HC) + log(SO2) + log(NOX), data = ex1217, subset
= -c(8, 60))
```



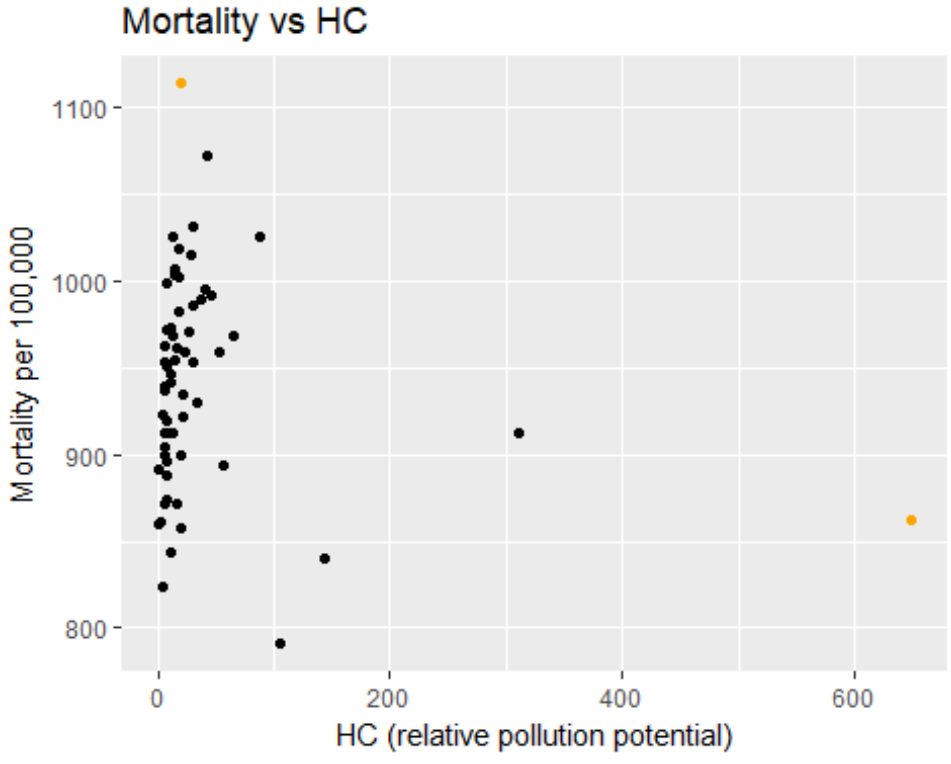
The residual plots shown above indicate that after removing the two outliers we are left with less variance and a more smooth pattern among the residuals. The correlation matrix that we created essentially indicates that there is a problem with multicollinearity; specifically, log(HC) and log(NOX) appear to be highly correlated, as do log(SO2) and log(NOX). The result of this correlation between explanatory variables needs to be addressed in order for our model in order not to result in seemingly significant results due to inflated p-values.

As a result, we elected to add a composite variable. This variable is the average of the logs of two columns, HC and NOX, to the dataset ex1217.

```
library(dplyr)

# This code updates the dataset with a column consisting of the
composite
# variable evaluated at each value of HC and NOX.
ex1217 <- mutate(ex1217, logHC_logNOX_Average = (log(HC) + log(NOX))/
2)
```

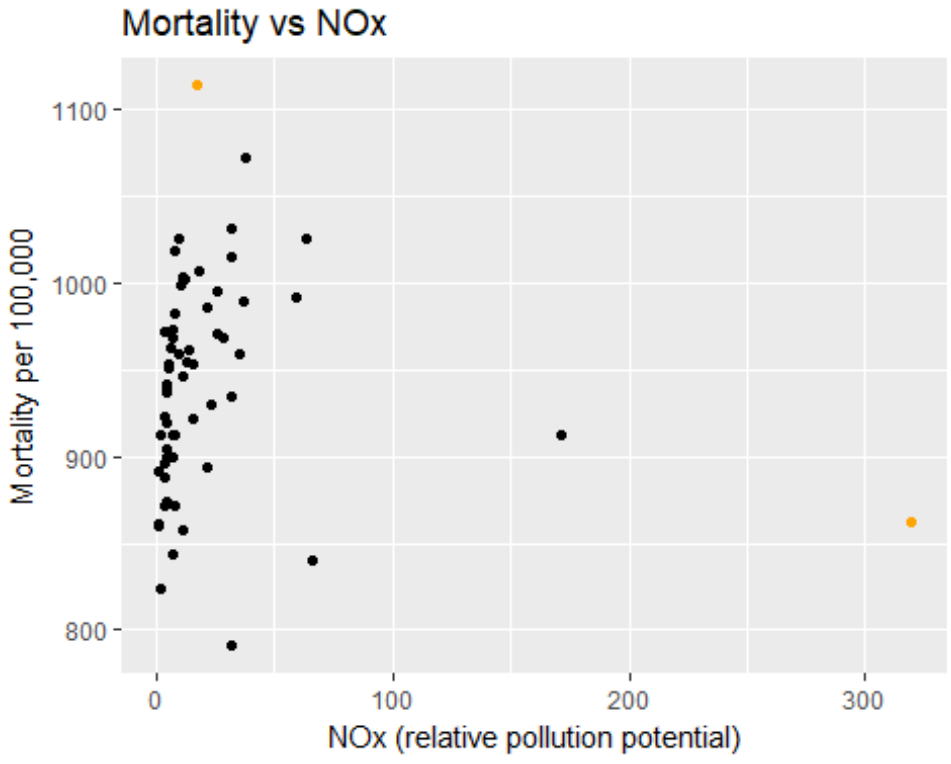
Ultimately we determined that the two variables HC and NOX did not matter very much and that the SO2 value was generally the sole variable of our interest that was associated with the mortality rate. We also looked at scatterplots of the original variables versus the transformed variables, and included our plots of the log-transformed variables in our final report to indicate that the transformation effectively solved the non-linearity (which resembles a plot following the sqrt function) problem of our data.

```
# Scatterplots of original (not transformed) variables with
# outliers highlighted in orange.
gf_point(Mortality ~ (HC), data = ex1217) %>%
  gf_point(Mortality ~ (HC), data = filter(ex1217, CITY == "Los
Angeles, CA"), color = "orange") %>%
  gf_point(Mortality ~ (HC), data = filter(ex1217, CITY == "New
Orleans, LA"), color = "orange") %>%
  gf_labs(x = "HC (relative pollution potential)", y = "Mortality per
100,000", title = "Mortality vs HC")
```

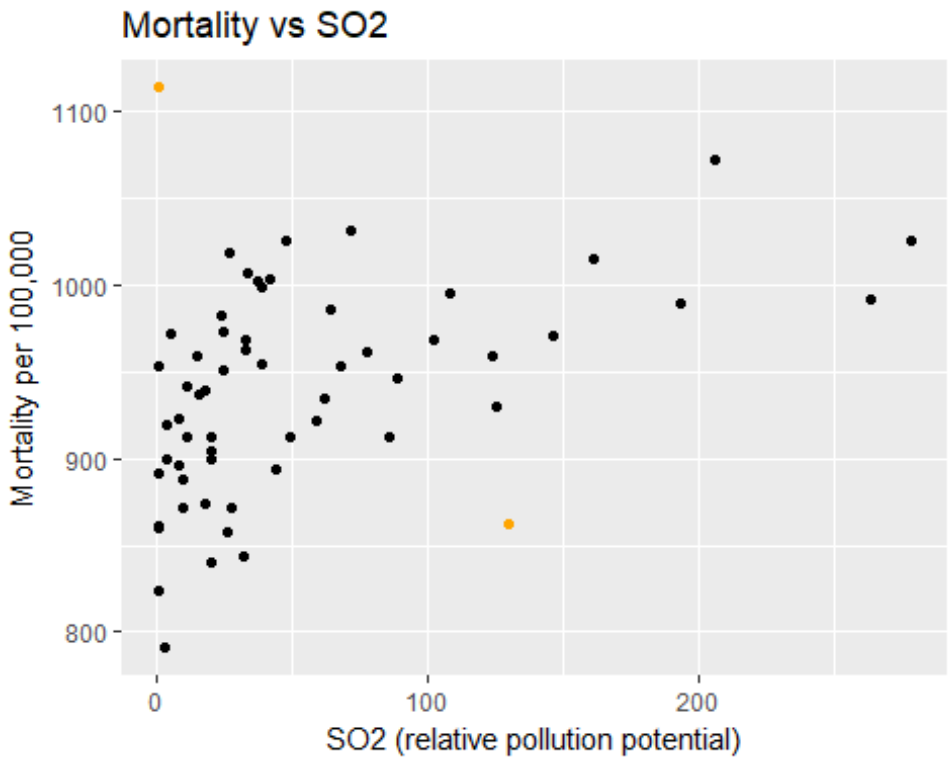



```
gf_point(Mortality ~ (NOX), data = ex1217) %>%
  gf_point(Mortality ~ (NOX), data = filter(ex1217, CITY == "Los
```

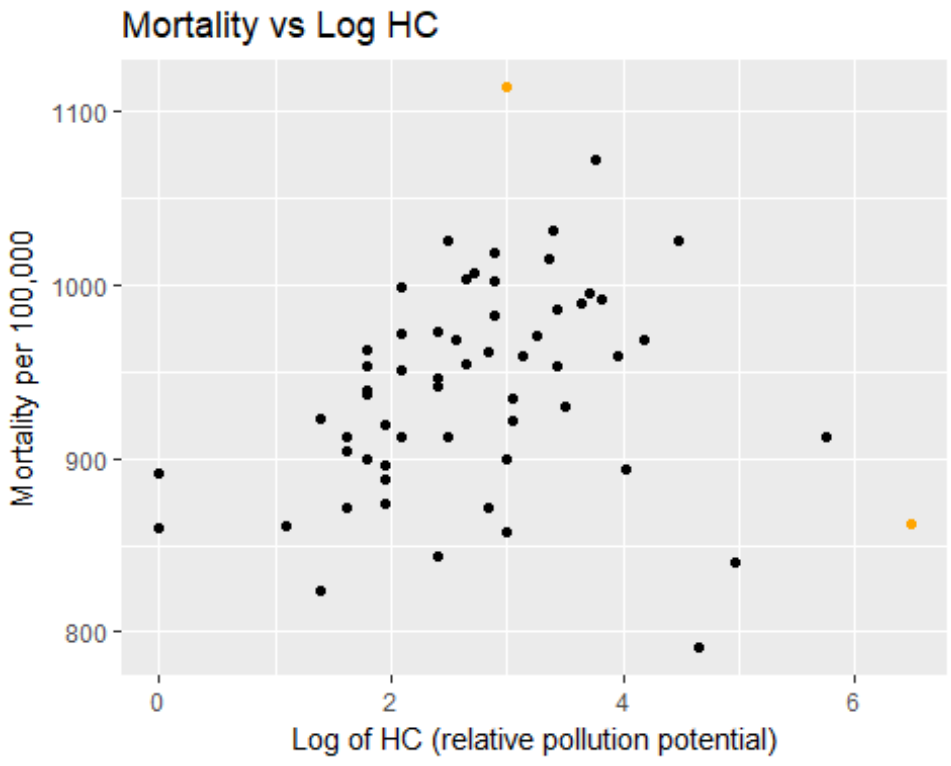
```
Angeles, CA"), color = "orange") %>%
  gf_point(Mortality ~ (NOX), data = filter(ex1217, CITY == "New
Orleans, LA"), color = "orange") %>%
  gf_labs(x = "NOx (relative pollution potential)", y = "Mortality per
100,000", title = "Mortality vs NOx")
```



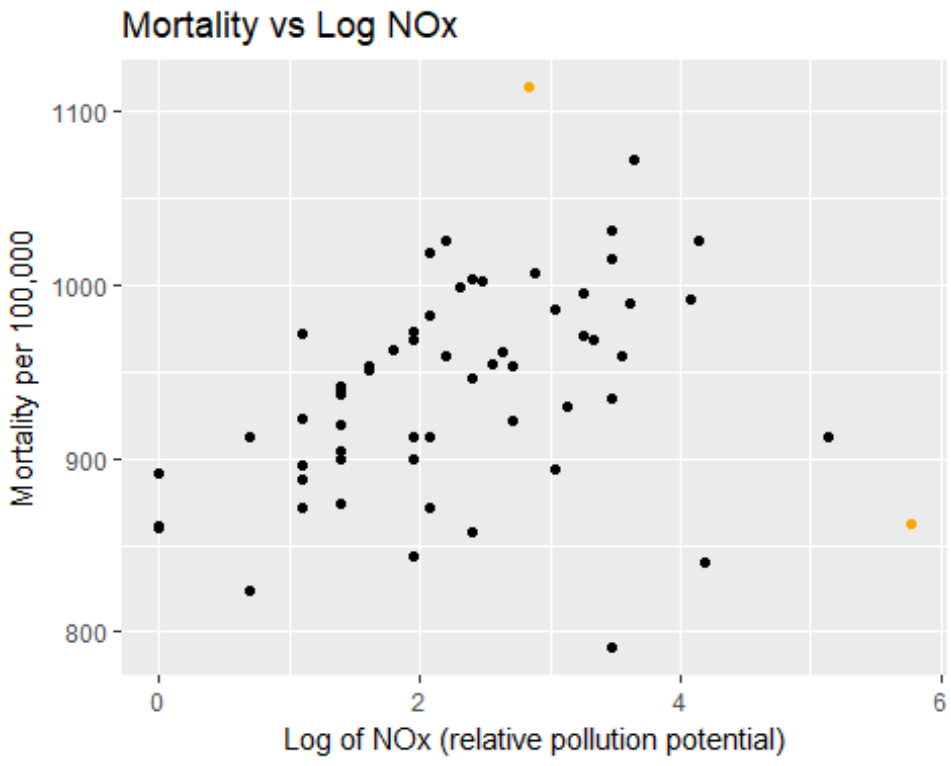

```
gf_point(Mortality ~ (SO2), data = ex1217) %>%
  gf_point(Mortality ~ (SO2), data = filter(ex1217, CITY == "Los
Angeles, CA"), color = "orange") %>%
  gf_point(Mortality ~ (SO2), data = filter(ex1217, CITY == "New
Orleans, LA"), color = "orange") %>%
  gf_labs(x = "SO2 (relative pollution potential)", y = "Mortality per
100,000", title = "Mortality vs SO2")
```



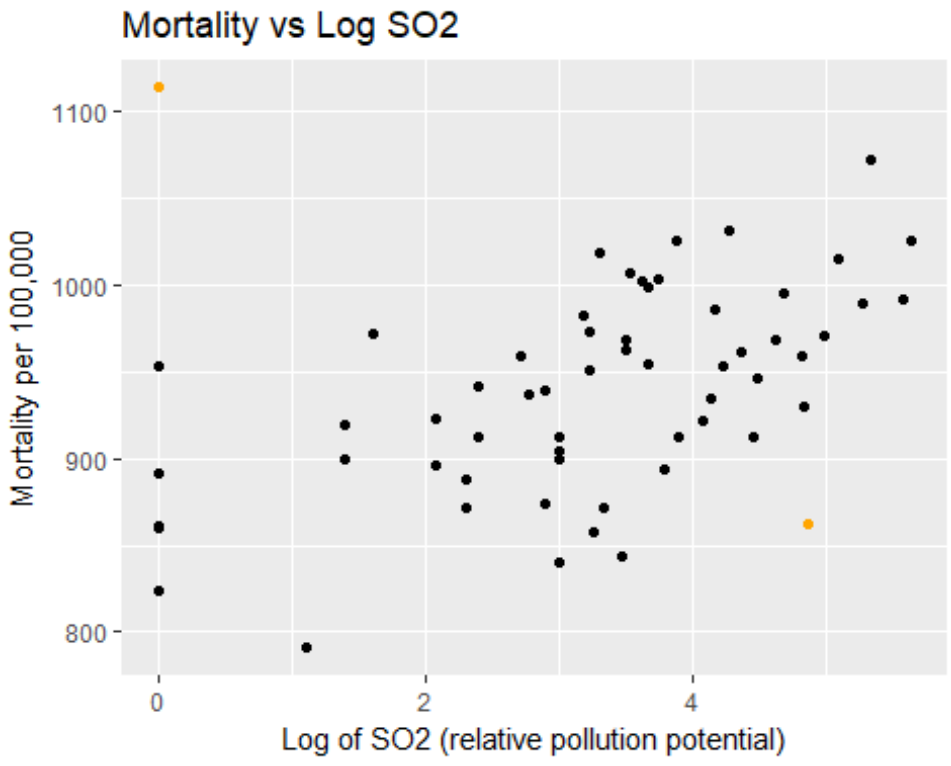

```
# Scatterplots of log-transformed variables with outlier highlighted.
gf_point(Mortality ~ log(HC), data = ex1217) %>%
  gf_point(Mortality ~ log(HC), data = filter(ex1217, CITY == "Los
Angeles, CA"), color = "orange") %>%
  gf_point(Mortality ~ log(HC), data = filter(ex1217, CITY == "New
Orleans, LA"), color = "orange") %>%
  gf_labs(x = "Log of HC (relative pollution potential)", y =
"Mortality per 100,000", title = "Mortality vs Log HC")
```




```
gf_point(Mortality ~ log(NOX), data = ex1217) %>%
  gf_point(Mortality ~ log(NOX), data = filter(ex1217, CITY == "Los
Angeles, CA"), color = "orange") %>%
  gf_point(Mortality ~ log(NOX), data = filter(ex1217, CITY == "New
Orleans, LA"), color = "orange") %>%
  gf_labs(x = "Log of NOx (relative pollution potential)", y =
"Mortality per 100,000", title = "Mortality vs Log NOx")
```




```
gf_point(Mortality ~ log(SO2), data = ex1217) %>%
  gf_point(Mortality ~ log(SO2), data = filter(ex1217, CITY == "Los
Angeles, CA"), color = "orange") %>%
  gf_point(Mortality ~ log(SO2), data = filter(ex1217, CITY == "New
Orleans, LA"), color = "orange") %>%
  gf_labs(x = "Log of SO2 (relative pollution potential)", y =
"Mortality per 100,000", title = "Mortality vs Log SO2")
```



After deciding on our composite variable we fit our final model.

```
# Fit a third model based on our composite:
model3 <- lm(Mortality ~ logHC_logNOX_Average + log(SO2), data =
ex1217, subset = -c(8, 60))

# We computed values for a 95% confidence
# interval for the coefficient
# of our composite variable.
-15.7+2*8.44
```

```
## [1] 1.18
```

```
-15.7-2*8.44
```

```
## [1] -32.58
```

```
# and for a 95% CI of our log SO2 variable.
33.2+2*6.32
```

```
## [1] 45.84
```

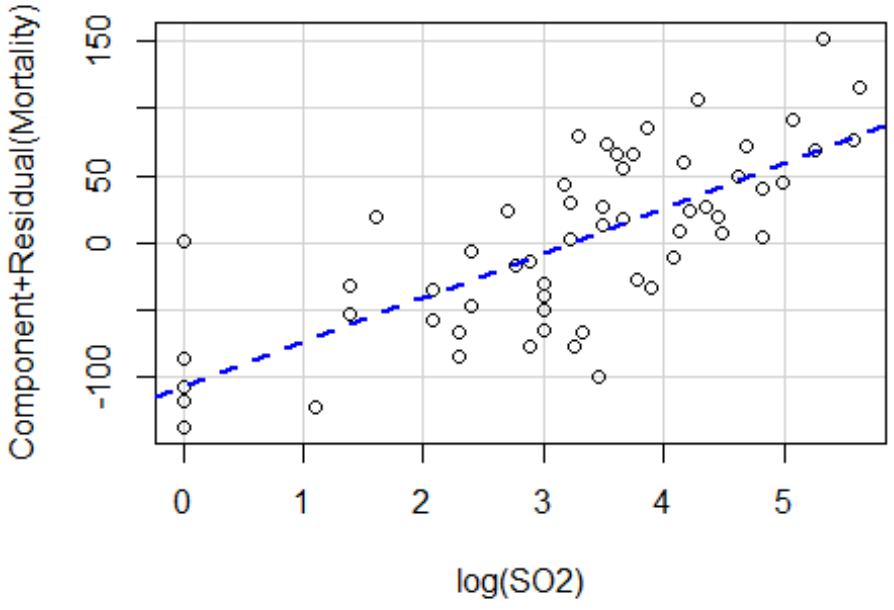
```r
33.2-2*6.32
```

```
## [1] 20.56
```

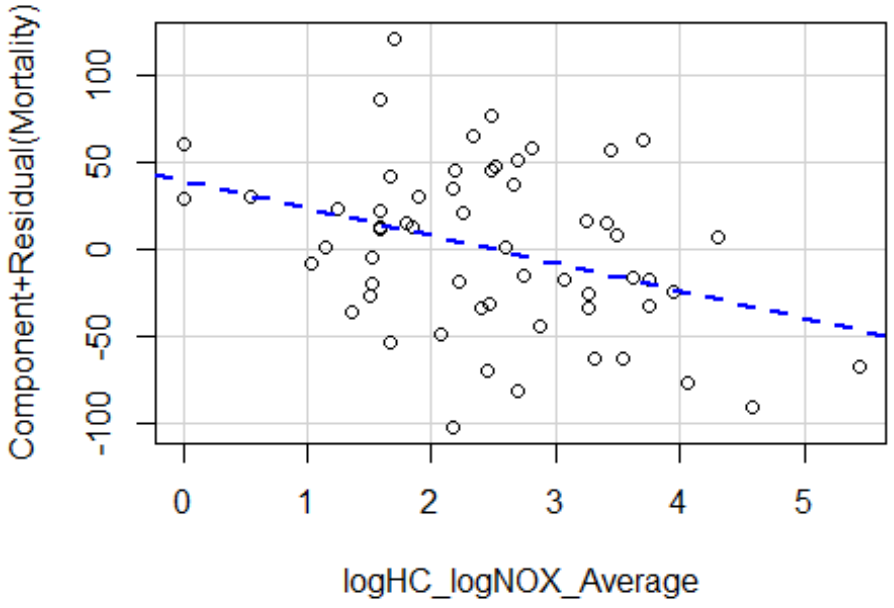
```r
# The Adjusted R-Squared value indicated
# below merits some justification.
summary(model3)
```

```
##
## Call:
## lm(formula = Mortality ~ logHC_logNOX_Average + log(SO2), data =
ex1217,
##     subset = -c(8, 60))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -107.41  -31.58   -1.02   29.25  108.57
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            870.708     15.748  55.289  < 2e-16 ***
## logHC_logNOX_Average  -15.730      8.439  -1.864   0.0677 .
## log(SO2)               33.156      6.315   5.251 2.52e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 45.59 on 55 degrees of freedom
## Multiple R-squared:  0.4049, Adjusted R-squared:  0.3833
## F-statistic: 18.71 on 2 and 55 DF,  p-value: 6.316e-07
```

```r
# These partial residual plots allow us to analyze
# potential non-linearity concerns.
crPlot(model3, variable = "log(SO2)", smooth = FALSE)
```




```
crPlot(model3, variable = "logHC_logNOX_Average", smooth = FALSE)
```



Given that our partial residual plots do not indicate any obvious patterns we can say that the model does fit the log-transformed data despite not accounting for all of the variance as shown by our low adjusted R^2 value.