Gasbarro - Linear Regression - Exercise 0

February 18, 2019

Contents

Exercise 0: Linear Regression

Load the STATES data

```
## IMPORTING THE DATA

library(readxl)
library(tidyverse)
library(dplyr)
library(readxl)
library(stringr)
library(lubridate)
library(ggplot2)
library(xts)
library(scales)

# Load the data
statesData <- readRDS("dataSets/states.rds")
```

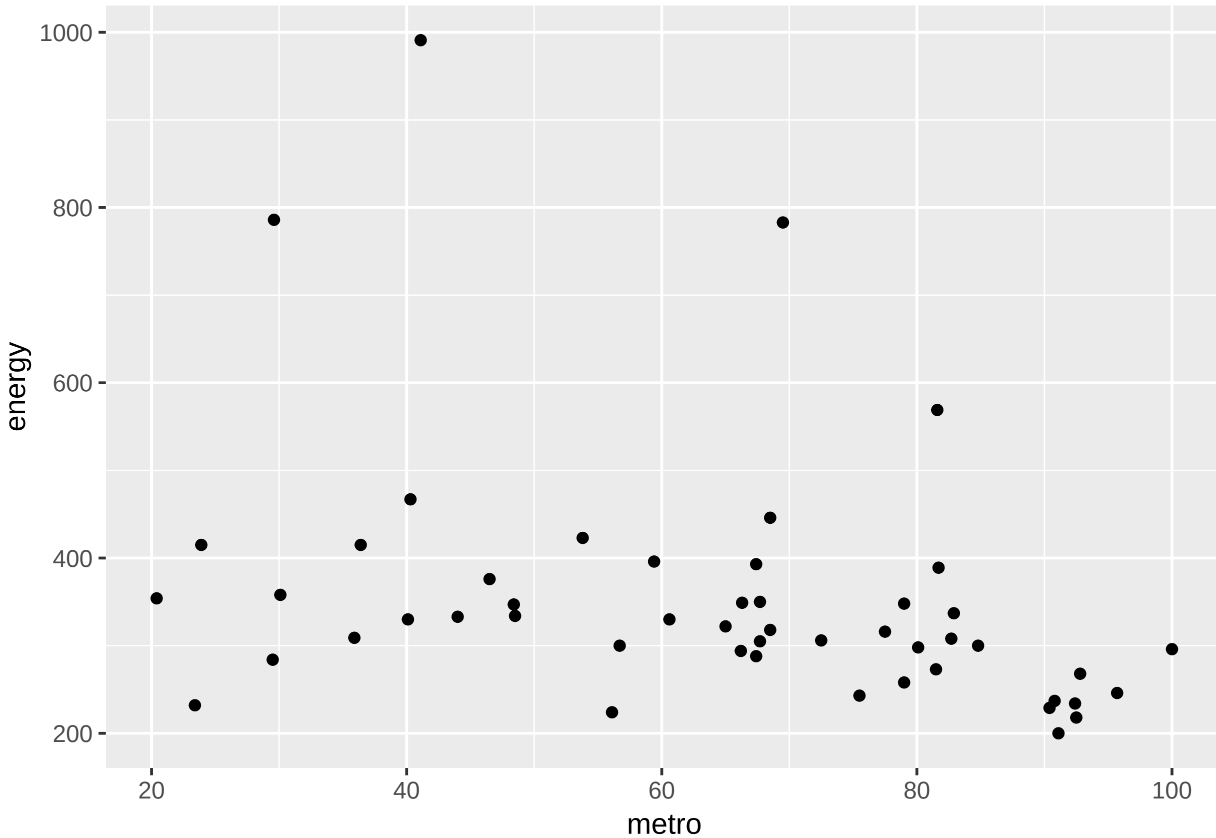
Examine and plot the data before fitting the model

```
# Select data for METRO areas and ENERGY consumption
statesMetroEnergy <- subset(statesData, select = c("metro", "energy"))

summary(statesMetroEnergy)
```

```
##      metro            energy
```

```
##  Min.   : 20.40   Min.   :200.0
##  1st Qu.: 46.98   1st Qu.:285.0
##  Median : 67.55   Median :320.0
##  Mean   : 64.07   Mean   :354.5
##  3rd Qu.: 81.58   3rd Qu.:371.5
##  Max.   :100.00   Max.   :991.0
##  NA's   :1        NA's   :1
```

```
# Plot X=METRO against Y=ENERGY
ggplot(statesMetroEnergy, aes(x=metro,y=energy)) +
  geom_point()
```



Print and interpret the model

```
cor(statesMetroEnergy, use="pairwise")
```

```
##             metro     energy
## metro   1.0000000 -0.3397445
## energy -0.3397445  1.0000000
```

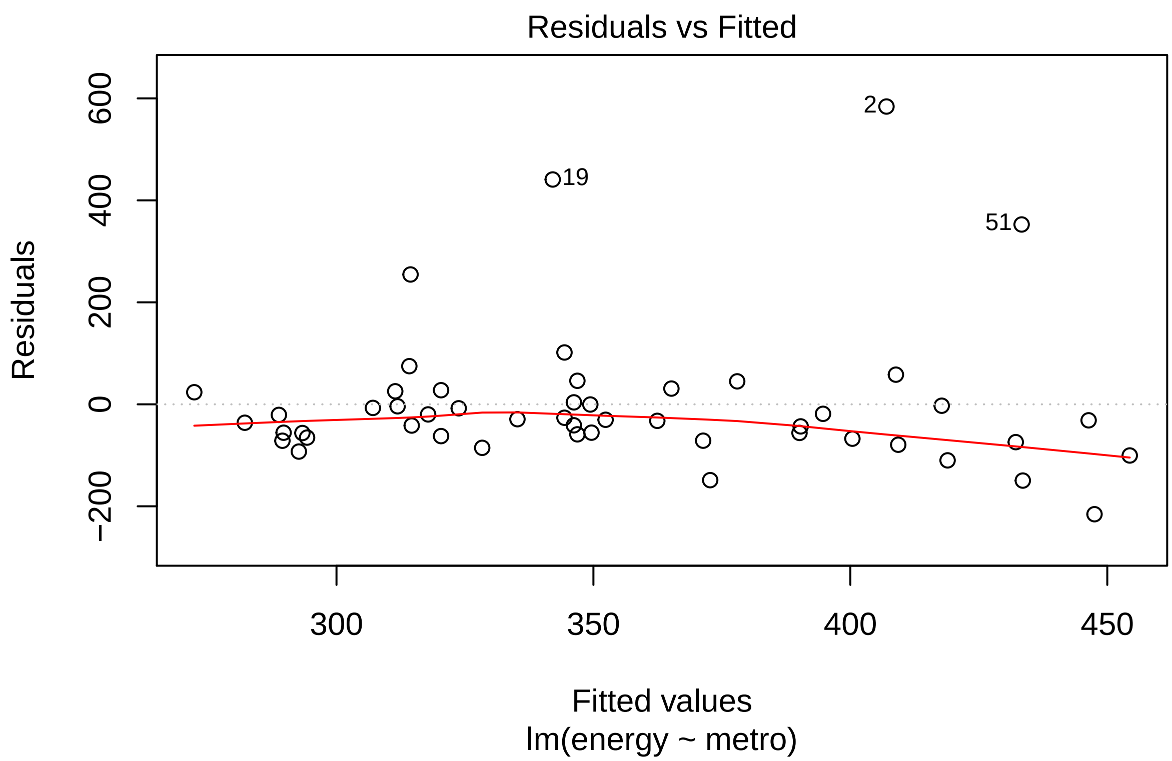
```
modelMetroEnergy <- lm(energy ~ metro, data = statesMetroEnergy)

summary(modelMetroEnergy)
```

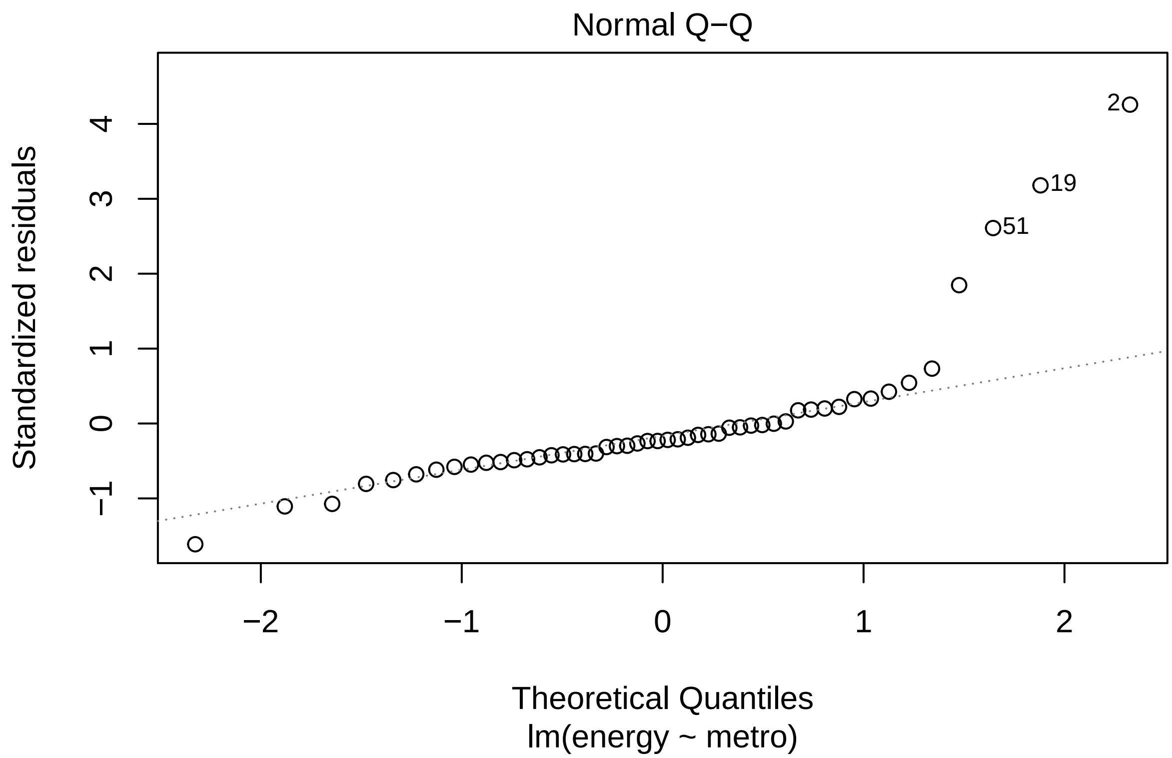
```
##
## Call:
## lm(formula = energy ~ metro, data = statesMetroEnergy)
```

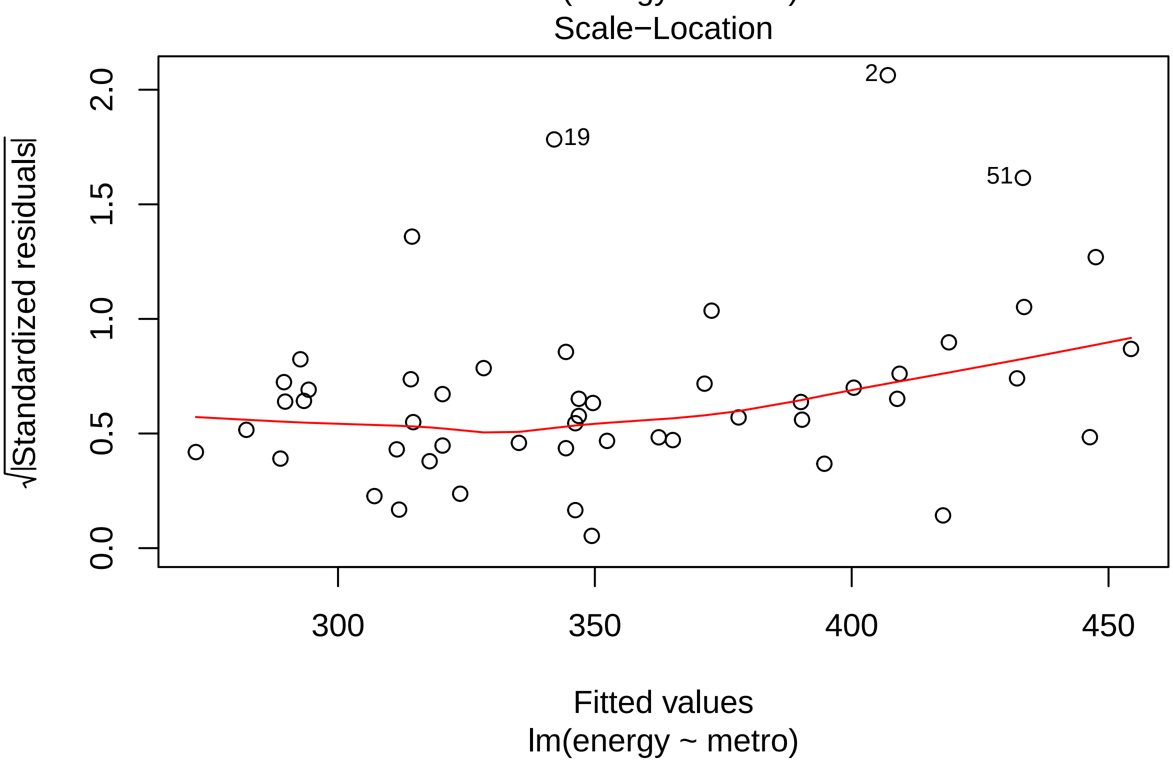
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -215.51  -64.54  -30.87   18.71  583.97
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 501.0292    61.8136   8.105 1.53e-10 ***
## metro        -2.2871     0.9139  -2.503   0.0158 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140.2 on 48 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1154, Adjusted R-squared:  0.097
## F-statistic: 6.263 on 1 and 48 DF,  p-value: 0.01578
```

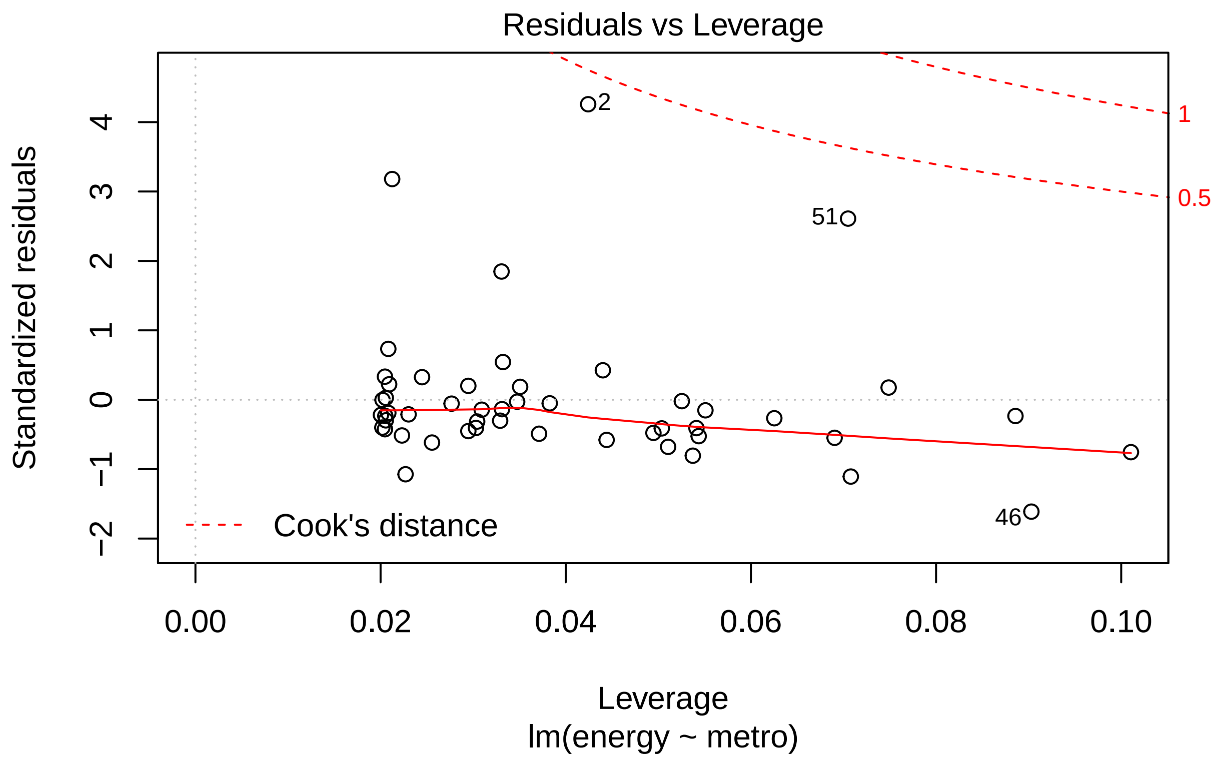
Plot the model to look for deviations from modeling assumptions

```r
#Assuming we use the lm dataframe we just generated

plot(modelMetroEnergy)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(energy ~ metro)
−2
−1
0
1
2
3
4
2
19
51


Scale−Location
√|Standardized residuals|
Fitted values
lm(energy ~ metro)
0.0
0.5
1.0
1.5
2.0
300
350
400
450
2
19
51




```r
# CAN YOU USE GGPLOT TO GENERATE THE SAME PLOTS?
```

Select one or more additional predictors

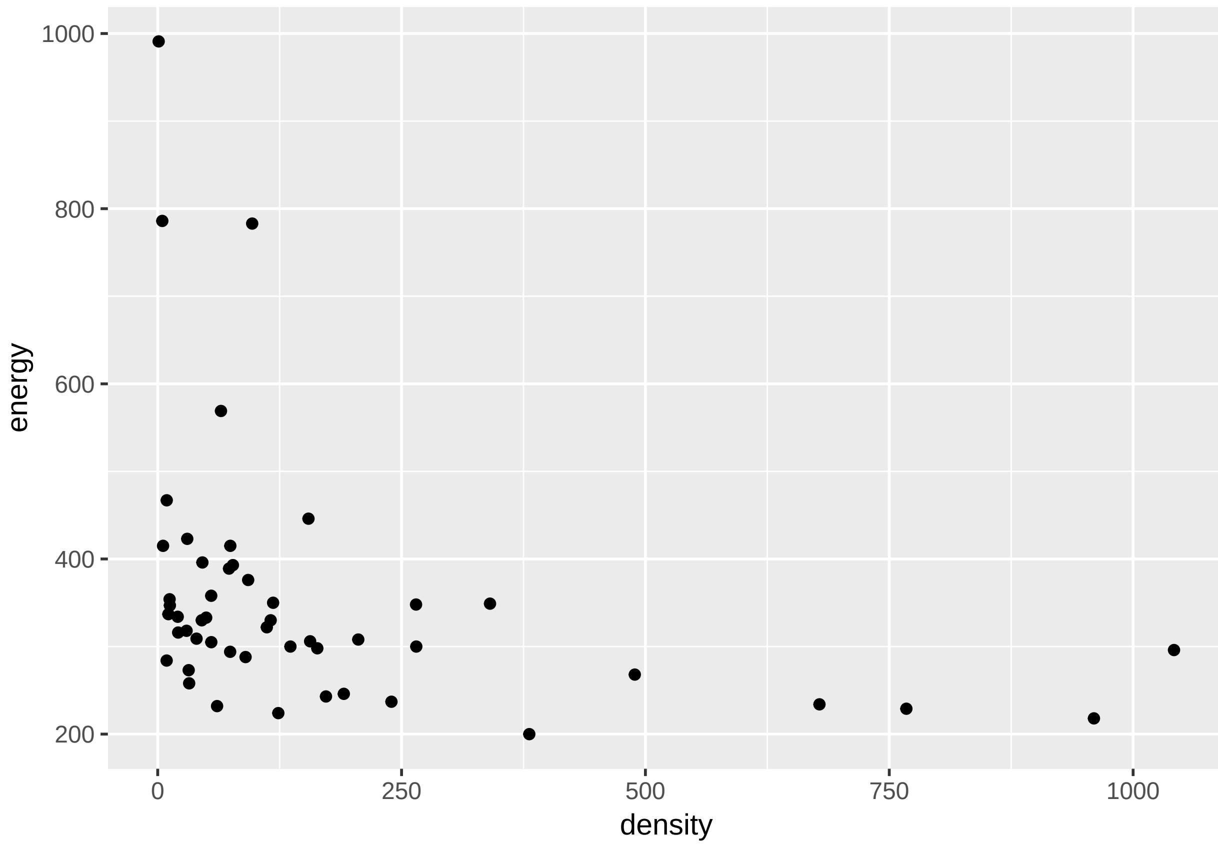
Select one or more additional predictors to add to your model and repeat steps 1-3. Is this model significantly better than the model with metro as the only predictor?

For this exercise, we're using DENSITY and INCOME

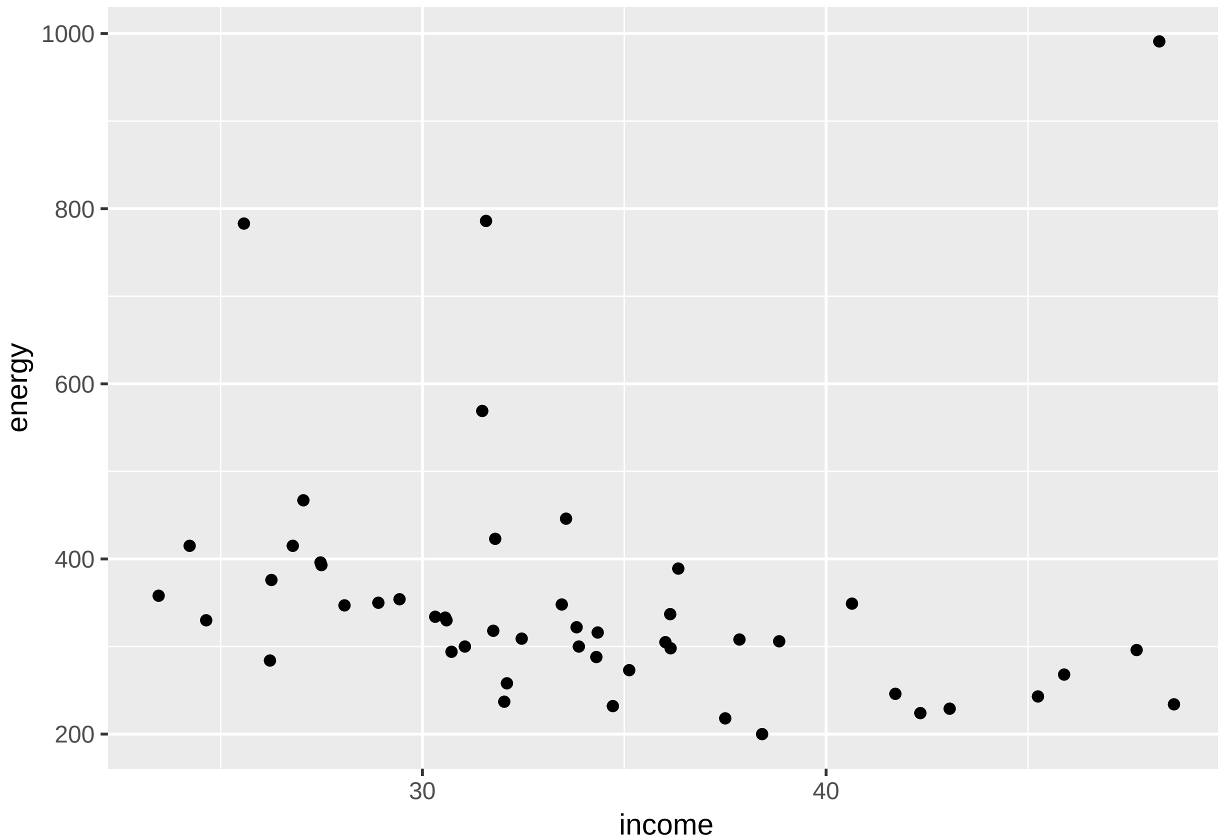
```r
# Select data for METRO, ENERGY, DENSITY, and INCOME
statesMetEnerDenInc <- subset(statesData, select = c("metro", "energy", "density", "income"))

summary(statesMetEnerDenInc)
```

```
##      metro            energy          density            income     
##  Min.   : 20.40   Min.   :200.0   Min.   :   0.96   Min.   :23.46  
##  1st Qu.: 46.98   1st Qu.:285.0   1st Qu.:  31.88   1st Qu.:29.88  
##  Median : 67.55   Median :320.0   Median :  75.76   Median :33.45  
##  Mean   : 64.07   Mean   :354.5   Mean   : 166.04   Mean   :33.96  
##  3rd Qu.: 81.58   3rd Qu.:371.5   3rd Qu.: 170.29   3rd Qu.:36.92  
##  Max.   :100.00   Max.   :991.0   Max.   :1041.92   Max.   :48.62  
##  NA's   :1        NA's   :1       NA's   :1
```

```r
# Plot X=METRO against Y=ENERGY
ggplot(statesMetEnerDenInc, aes(x=density,y=energy)) +
  geom_point()
```




```r
ggplot(statesMetEnerDenInc, aes(x=income,y=energy)) +
  geom_point()
```




```r
cor(statesMetEnerDenInc, use="pairwise")
```

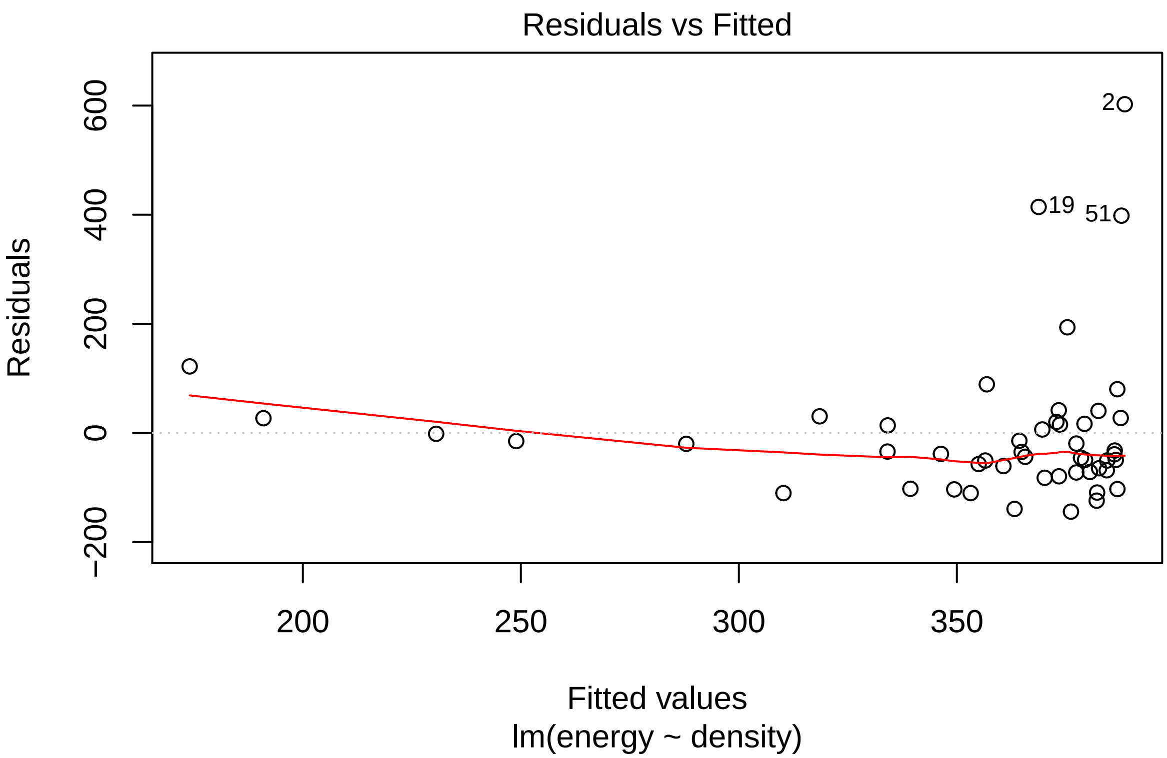
```
##              metro     energy    density     income
## metro    1.0000000 -0.3397445  0.6001587  0.5795649
## energy  -0.3397445  1.0000000 -0.3284203 -0.1436852
## density  0.6001587 -0.3284203  1.0000000  0.5928678
## income   0.5795649 -0.1436852  0.5928678  1.0000000
```

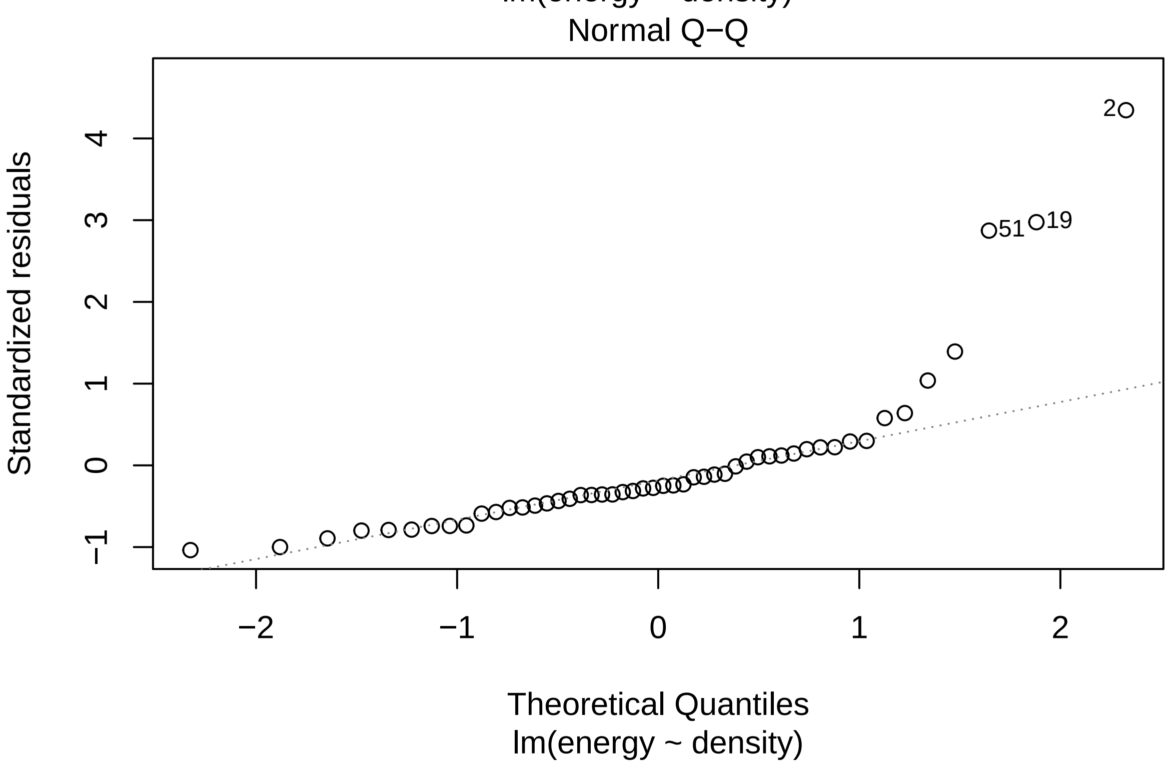
```r
modelDenEner <- lm(energy ~ density, data = statesMetEnerDenInc)

summary(modelDenEner)
```

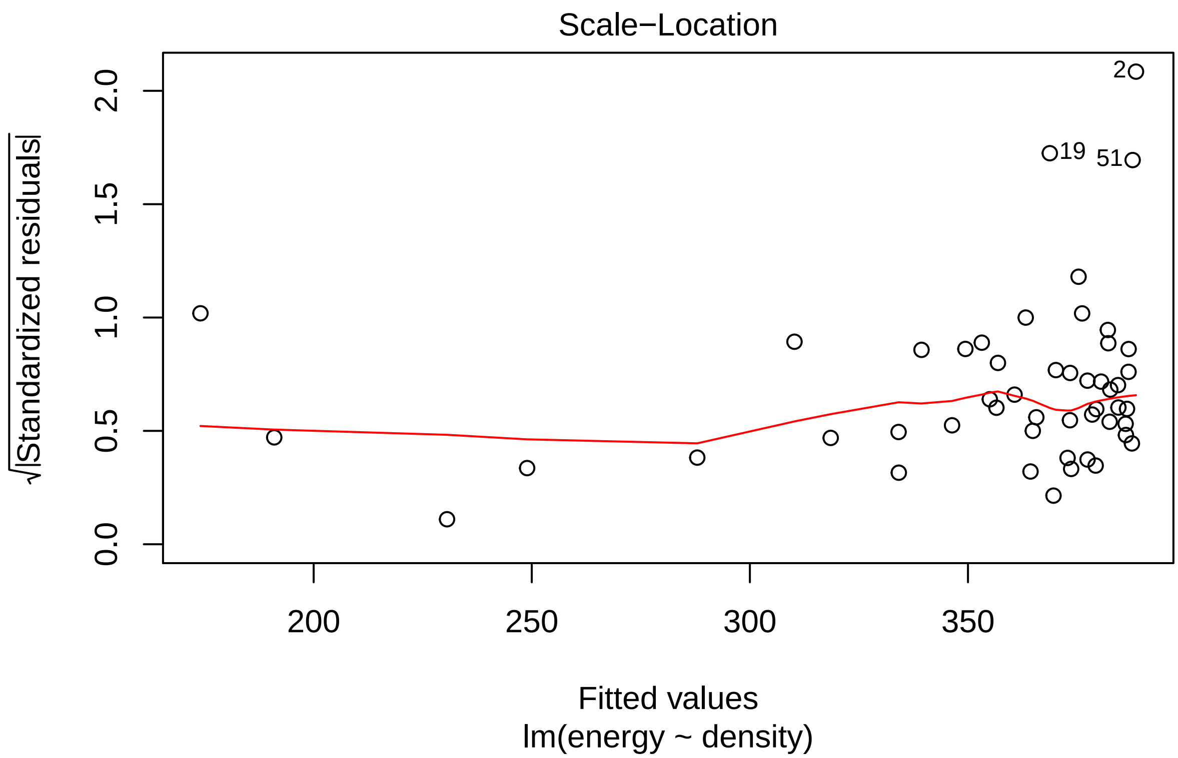
```
##
## Call:
## lm(formula = energy ~ density, data = statesMetEnerDenInc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -144.17  -70.73  -36.60   19.31  602.49
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 388.70969   24.45374  15.896   <2e-16 ***
## density      -0.20603    0.08553  -2.409   0.0199 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140.8 on 48 degrees of freedom
```

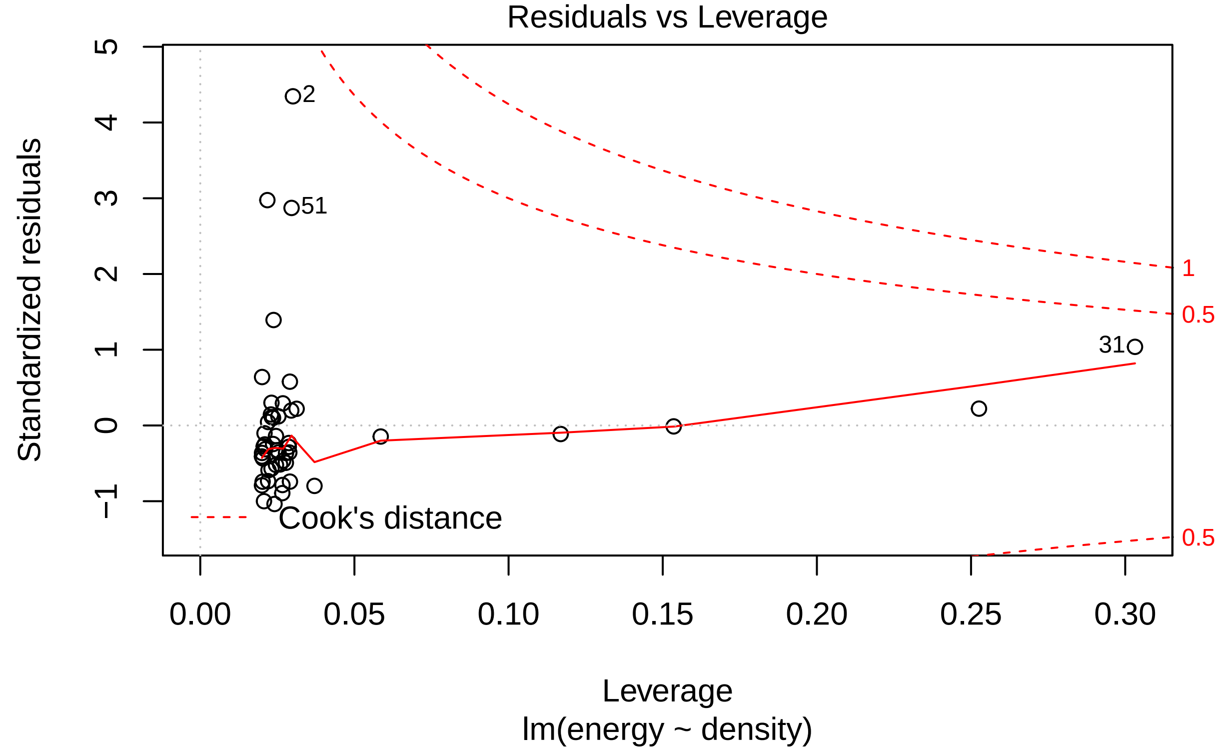
```
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1079, Adjusted R-squared:  0.08927
## F-statistic: 5.803 on 1 and 48 DF,  p-value: 0.01988
```

```r
plot(modelDenEner)
```






Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
200
250
300
350
2
19
51
Fitted values
lm(energy ~ density)


Residuals vs Leverage
Standardized residuals
−1
0
1
2
3
4
5
0.00
0.05
0.10
0.15
0.20
0.25
0.30
2
51
31
1
0.5
0.5
Cook's distance
Leverage
lm(energy ~ density)

Exercise 1

1.1 Add on to the regression

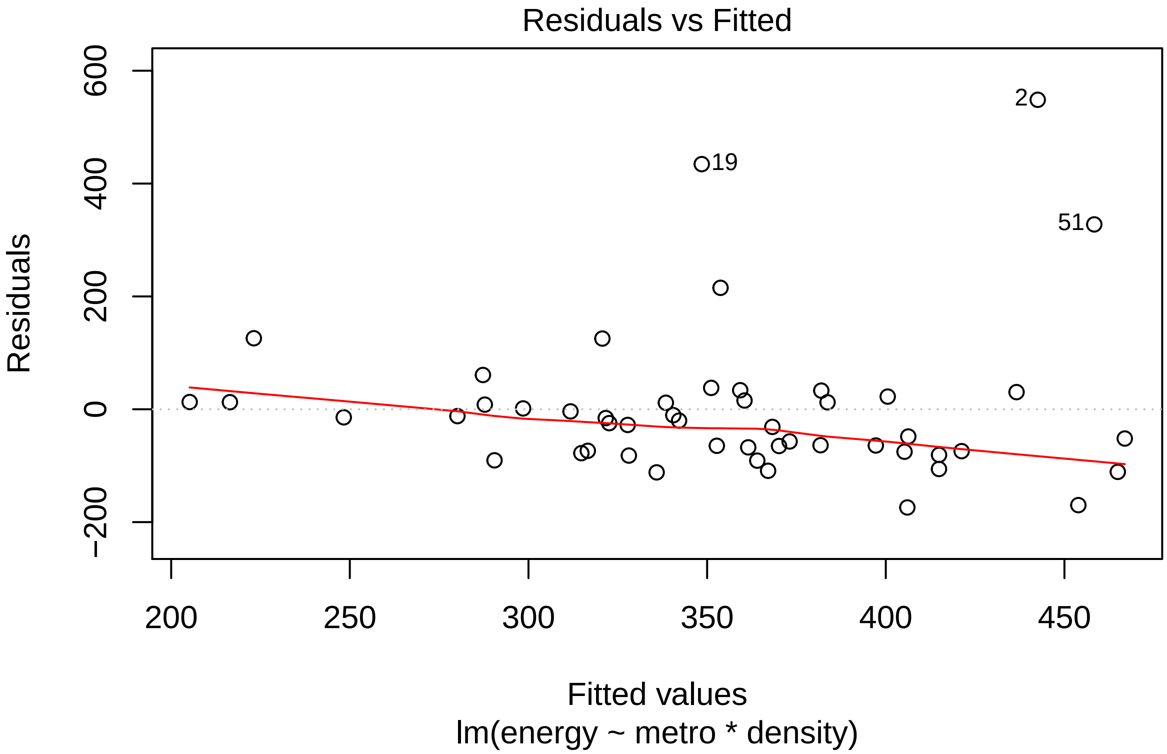
Add on to the regression equation that you created in exercise 1 by generating an interaction term and testing the interaction.

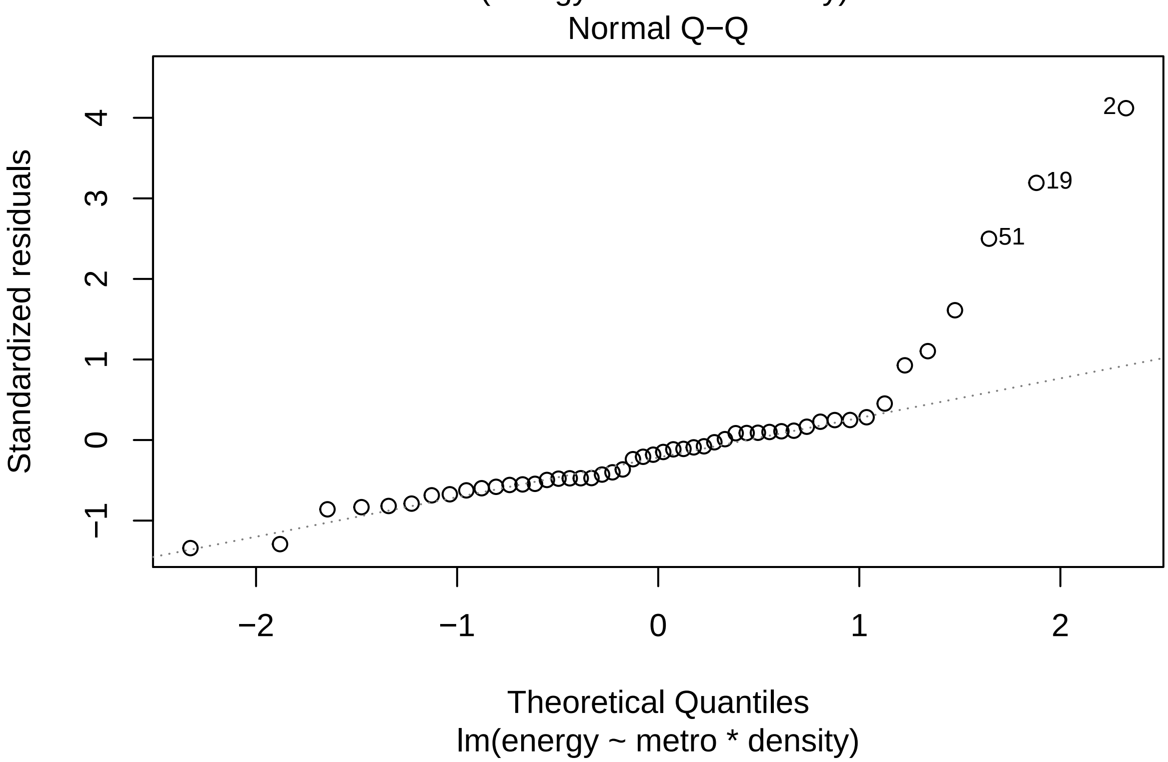
I used the following variables in my previous exercise: * Metro * Energy * Density * Income

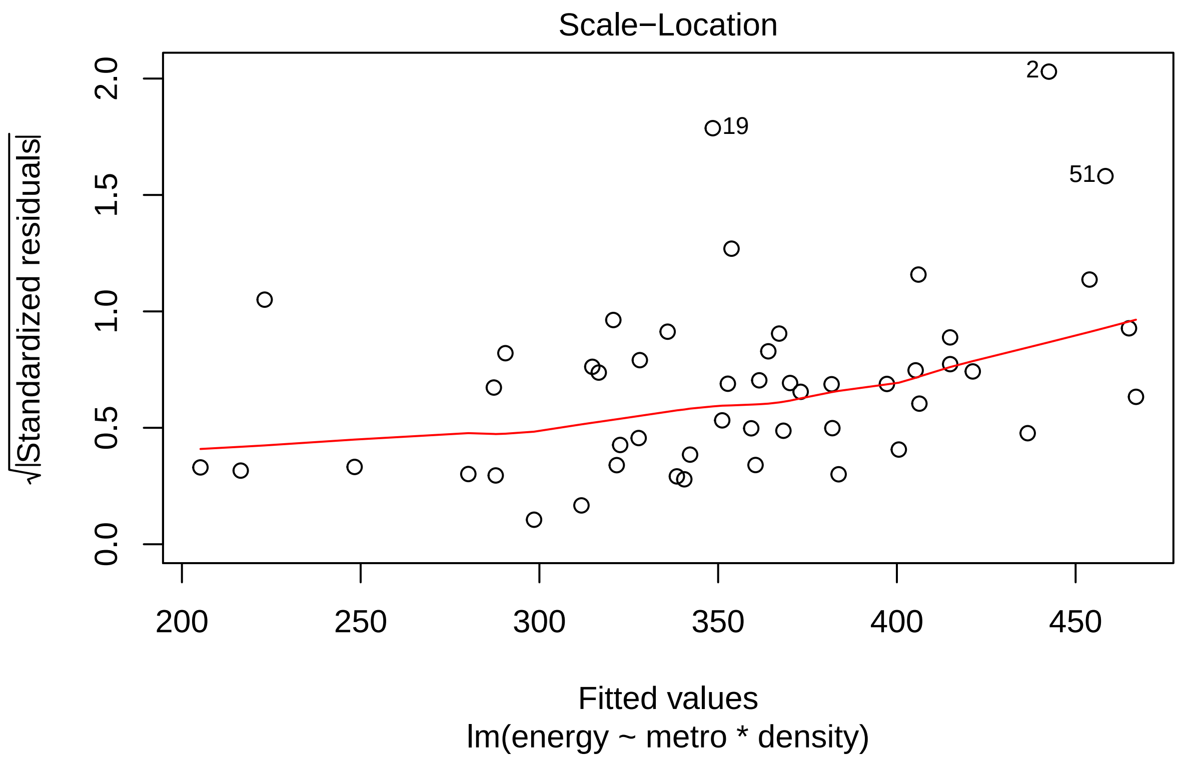
```
# Use statesMetEnerDenInc
#modelDenEner <- lm(energy ~ density, data = statesMetEnerDenInc)

modelEnergyMetroXDensity <- lm(energy ~ metro * density, data = statesMetEnerDenInc)

summary(modelEnergyMetroXDensity)
```

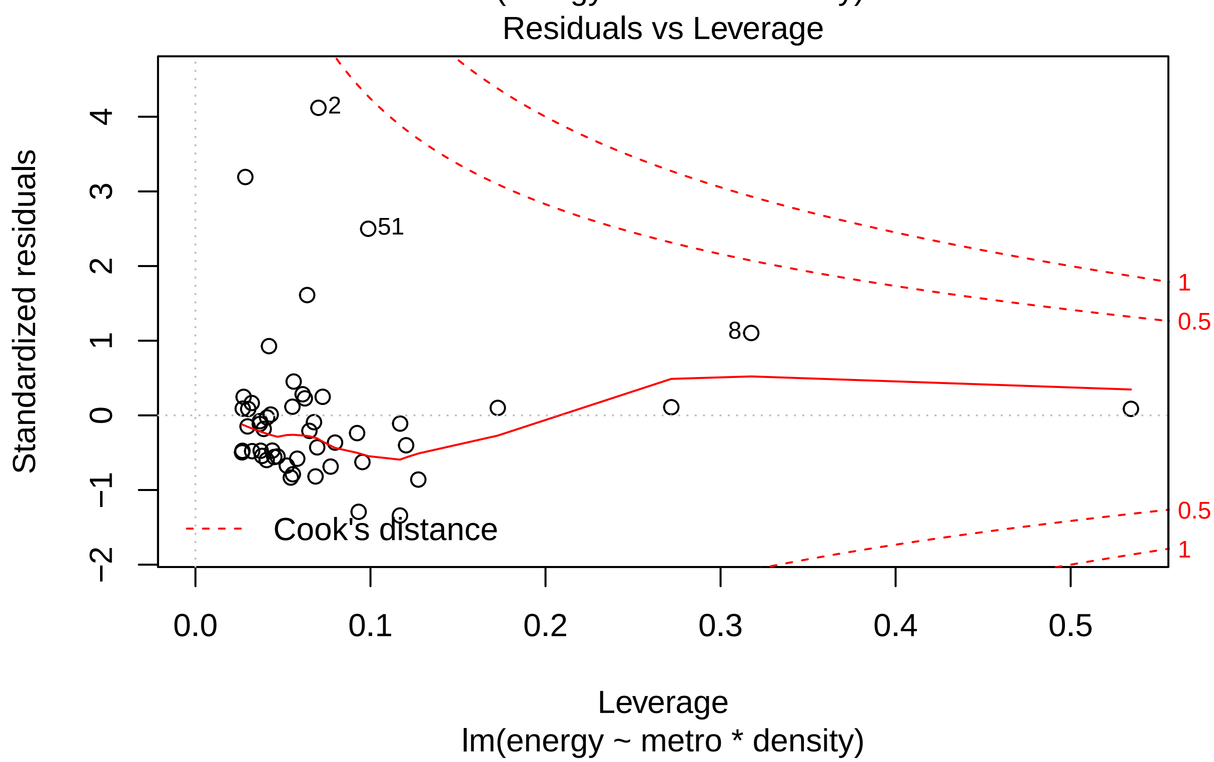
```
##
## Call:
## lm(formula = energy ~ metro * density, data = statesMetEnerDenInc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -174.03  -74.07  -22.37   14.86  548.47
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   514.104243  72.513405   7.090 6.68e-09 ***
## metro          -1.721120   1.135555  -1.516    0.136
## density        -1.438179   0.911292  -1.578    0.121
## metro:density   0.013861   0.009534   1.454    0.153
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 138.1 on 46 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1775, Adjusted R-squared:  0.1238
## F-statistic: 3.308 on 3 and 46 DF,  p-value: 0.02823
```

```
plot(modelEnergyMetroXDensity)
```


Residuals vs Fitted
Residuals
600
400
200
0
−200
200
250
300
350
400
450
Fitted values
lm(energy ~ metro * density)
2
19
51


Normal Q−Q
Standardized residuals
4
3
2
1
0
−1
−2
−1
0
1
2
Theoretical Quantiles
lm(energy ~ metro * density)
2
19
51





1.2 Regions

Try adding a region to the model. Are there significant differences across the four regions?

? Any easier way to do naming conventions?

```r
# Select data for METRO, ENERGY, DENSITY, INCOME, REGIONS
statesMetEnerDenIncReg <- subset(statesData, select = c("metro", "energy", "density", "income","region"

modelEnergyMetroXDensityRegion <- lm(energy ~ metro * density + region, data = statesMetEnerDenIncReg)

# I think we're supposed to use ANOVA for this exercise
anova(modelEnergyMetroXDensityRegion)
```

```
## Analysis of Variance Table
##
## Response: energy
##                Df Sum Sq Mean Sq F value  Pr(>F)
## metro           1 123064  123064  6.6011 0.01374 *
## density         1  25837   25837  1.3859 0.24557
## region          3  80605   26868  1.4412 0.24400
## metro:density   1  35018   35018  1.8783 0.17763
## Residuals      43 801642   18643
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Exercise 2

Load the NatHealth2011.rds data

```r
NatHealth2011 <- readRDS("dataSets/NatHealth2011.rds")
labs <- attributes(NatHealth2011)$labels
```

2.1 Use GLM

Use glm to conduct a logistic regression to predict ever worked (everwrk) using age (age_p) and marital status (r_maritl).

```r
# Select the 3 variables
NH2011_WorkAgeMar <- subset(NatHealth2011, select = c("everwrk", "r_maritl", "age_p"))

# Summary before transformation
summary(NH2011_WorkAgeMar)
```

```
##                everwrk                                  r_maritl
##   1 Yes            :12153   1 Married - spouse in household:13943
##   2 No             : 1887   7 Never married                : 7763
##   7 Refused        :   17   5 Divorced                     : 4511
##   8 Not ascertained:    0   4 Widowed                      : 3069
##   9 Don't know     :    8   8 Living with partner          : 2002
##   NA's             :18949   6 Separated                    : 1121
##                             (Other)                        :  605
##      age_p
##   Min.   :18.00
##   1st Qu.:33.00
##   Median :47.00
##   Mean   :48.11
##   3rd Qu.:62.00
```

```
##  Max.   :85.00
##
```

```r
# Transform using a factor on Work, and drop any unused levels in the Marital Status
NH2011_WorkAgeMar <- transform(NH2011_WorkAgeMar, everwrk = factor(everwrk, levels = c("1 Yes", "2 No"))

Model_NH2011_WorkAgeMar <- glm(everwrk ~ age_p + r_maritl, data = NH2011_WorkAgeMar, family = "binomial

summary(Model_NH2011_WorkAgeMar)
```

```
##
## Call:
## glm(formula = everwrk ~ age_p + r_maritl, family = "binomial",
##     data = NH2011_WorkAgeMar)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0436  -0.5650  -0.4391  -0.3370   2.7308
##
## Coefficients:
##                                              Estimate Std. Error z value
## (Intercept)                                 -0.440248   0.093538  -4.707
## age_p                                       -0.029812   0.001645 -18.118
## r_maritl2 Married - spouse not in household  0.049675   0.217310   0.229
## r_maritl4 Widowed                            0.683618   0.084335   8.106
## r_maritl5 Divorced                          -0.730115   0.111681  -6.538
## r_maritl6 Separated                         -0.128091   0.151366  -0.846
## r_maritl7 Never married                      0.343611   0.069222   4.964
## r_maritl8 Living with partner               -0.443583   0.137770  -3.220
## r_maritl9 Unknown marital status             0.395480   0.492967   0.802
##                                             Pr(>|z|)
## (Intercept)                                 2.52e-06 ***
## age_p                                        < 2e-16 ***
## r_maritl2 Married - spouse not in household  0.81919
## r_maritl4 Widowed                           5.23e-16 ***
## r_maritl5 Divorced                          6.25e-11 ***
## r_maritl6 Separated                          0.39742
## r_maritl7 Never married                     6.91e-07 ***
## r_maritl8 Living with partner                0.00128 **
## r_maritl9 Unknown marital status             0.42241
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11082  on 14039  degrees of freedom
## Residual deviance: 10309  on 14031  degrees of freedom
##   (18974 observations deleted due to missingness)
## AIC: 10327
##
## Number of Fisher Scoring iterations: 5
```

#2.2 Working v Marital Prediction Predict the probability of working for each level of marital status.

```r
# Install and Load the Effects library
```

```r
#install.packages("effects")
library(effects)

# Use Effect to predict the probability on the model from the previous exercise

data.frame(Effect("r_maritl", Model_NH2011_WorkAgeMar))
```

```
##                               r_maritl        fit          se      lower
## 1     1 Married - spouse in household 0.10822000 0.004259644 0.10014980
## 2 2 Married - spouse not in household 0.11310823 0.021393167 0.07746061
## 3                            4 Widowed 0.19381087 0.010634762 0.17381358
## 4                           5 Divorced 0.05524394 0.005361664 0.04562877
## 5                          6 Separated 0.09646417 0.012707502 0.07426824
## 6                      7 Never married 0.14611000 0.007459212 0.13208775
## 7                8 Living with partner 0.07224958 0.008904955 0.05662466
## 8             9 Unknown marital status 0.15270076 0.063528455 0.06440837
##        upper
## 1 0.11685606
## 2 0.16227532
## 3 0.21550873
## 4 0.06674358
## 5 0.12440219
## 6 0.16134411
## 7 0.09176661
## 8 0.32055728
```

Exercise 3

Where's the bh1996 dataset?

```r
# Need to load a package called MULTILEVEL
#install.packages("multilevel")
data(bh1996, package="multilevel")
```