Seawatch

Yifeng Huang

August 16, 2018

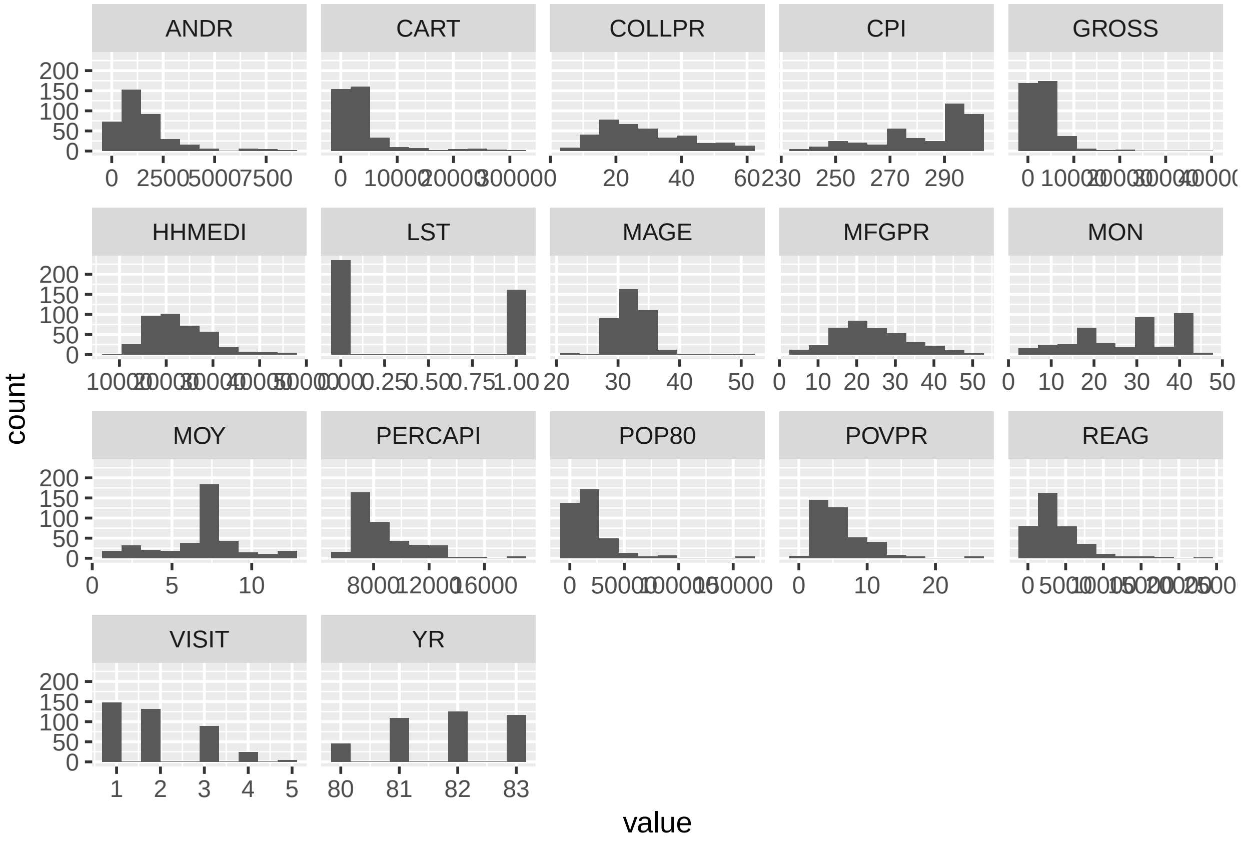
Data Clean

Overview

```
# remove environment
rm(list=ls())

# data import
seawatch.ori<-read_excel("~/MSBA notes/Business Stats/Seawatch C w blanks-1.xlsx")

# we delete CNVHRS, Notes, City and Zip code.
seawatch<-seawatch.ori[,3:20]
seawatch<-seawatch[,-2]
# overview
describe(seawatch)
```

```
##          vars   n     mean       sd  median  trimmed      mad     min
## GROSS       1 394  3441.30  4056.82  2419.5  2755.20  2227.61    43.0
## MOY         2 396     6.62     2.45     7.0     6.66     1.48     1.0
## YR          3 396    81.79     0.99    82.0    81.86     1.48    80.0
## MON         4 396    28.13    11.72    31.0    28.85    17.79     3.0
## VISIT       5 396     2.00     0.97     2.0     1.90     1.48     1.0
## LST         6 396     0.41     0.49     0.0     0.38     0.00     0.0
## CPI         7 396   282.14    16.85   290.6   284.31    13.34   236.4
## POP80       8 387 19179.16 22616.29 13212.0 14863.73 11421.95   688.0
## HHMEDI      9 387 22643.48  6654.16 21304.0 21984.26  6862.96 10108.0
## PERCAPI    10 387  8706.69  2236.39  8060.0  8429.94  1885.87  5188.0
## POVPR      11 387     6.20     3.96     5.1     5.68     2.67     0.4
## MFGPR      12 371    23.89     9.65    22.3    23.41     8.60     3.0
## COLLPR     13 371    28.72    13.20    25.9    27.55    14.97     8.2
## MAGE       14 387    32.23     3.06    32.1    32.07     2.67    21.7
## CART       15 380  3784.08  5064.56  2171.5  2628.76  1965.19    97.0
## REAG       16 380  3881.71  3430.15  3067.0  3359.21  2604.93    84.0
## ANDR       17 380  1528.81  1463.64  1105.0  1262.62   919.21    63.0
##               max    range  skew kurtosis      se
## GROSS     38256.0  38213.0  4.12    25.26  204.38
## MOY          12.0     11.0 -0.24     0.52    0.12
## YR           83.0      3.0 -0.28    -1.03    0.05
## MON          44.0     41.0 -0.27    -1.06    0.59
## VISIT         5.0      4.0  0.67    -0.27    0.05
## LST           1.0      1.0  0.38    -1.86    0.02
## CPI         300.9     64.5 -0.85    -0.38    0.85
## POP80    161799.0 161111.0  3.60    16.78 1149.65
## HHMEDI    47646.0  37538.0  1.03     1.39  338.25
## PERCAPI   17850.0  12662.0  1.30     1.94  113.68
## POVPR        26.1     25.7  2.00     6.23    0.20
## MFGPR       48.4     45.4  0.41    -0.16    0.50
## COLLPR      61.7     53.5  0.65    -0.47    0.69
## MAGE        50.2     28.5  1.33     7.57    0.16
## CART     31225.0  31128.0  3.13    10.62  259.81
## REAG     23339.0  23255.0  2.17     7.07  175.96
## ANDR      8586.0   8523.0  2.28     6.40   75.08
```

```
ggplot(gather(seawatch), aes(value)) +
    geom_histogram(bins = 10) +
    facet_wrap(~key, scales = 'free_x')
```



```
## Note that VISIT and LST are categorical variables

## Convert numberic variables to Categorical
seawatch$VISIT<-as.factor(seawatch$VISIT)
seawatch$LST<-as.factor(seawatch$LST)
```

Missing Values

```
# number of NA's
nrow(seawatch)-nrow(na.omit(seawatch))
```

```
## [1] 34
```

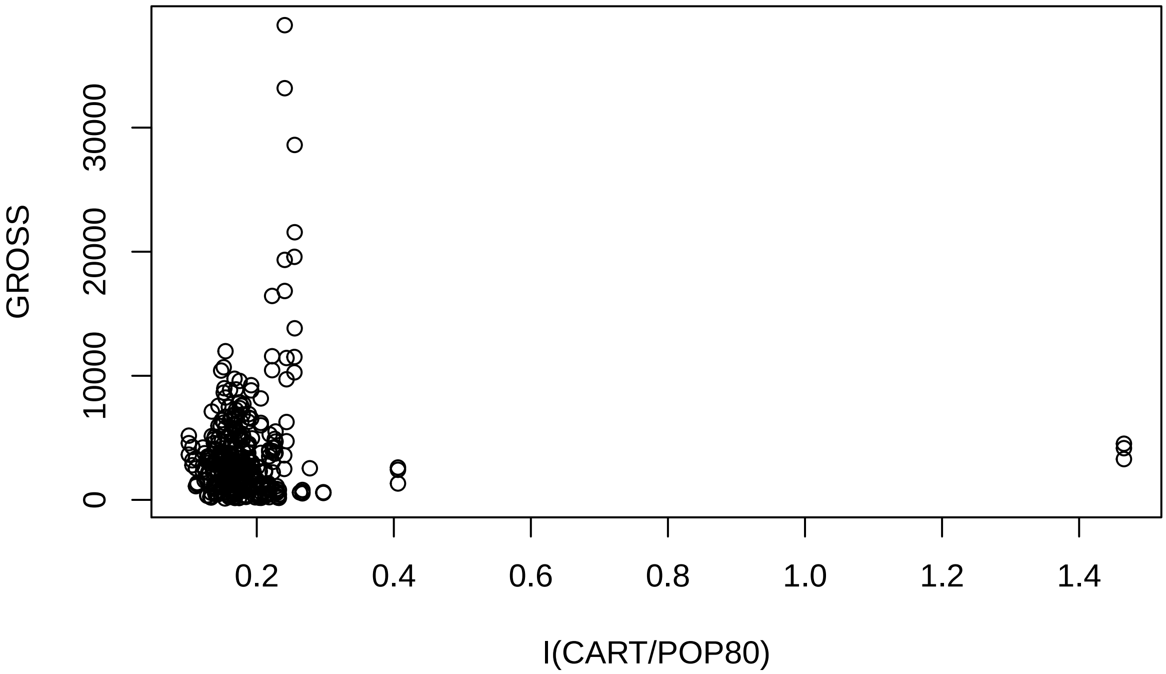
```
## Since there are only 34 rows containing na's we can simply delete it
seawatch<-na.omit(seawatch)
```

Other strange observation

CART, REAG, and ANDR

- For some observations, the number of votes is bigger than total population. Due to the high correlation (0.9533607) between POP80 and the sum of those there vote numbers, we can build a model to predict the right population.

```
# scatter plot
plot(GROSS~I(CART/POP80),data = seawatch)
```



```
summary(seawatch$CART)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     265    1148    2254    3955    4215   31225
```

- Note that some observations have CART/POP80 > 1, indicating CART larger than total population.

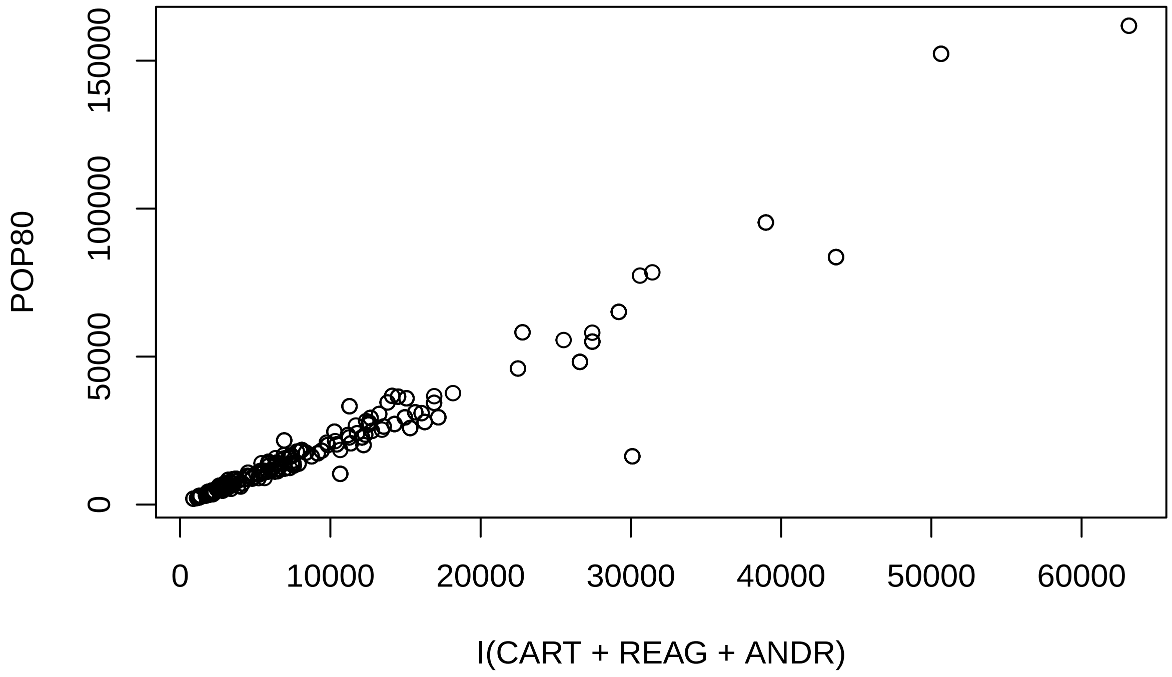
```
# observations that CART or REAG or ANDR is larger than total population
ex.obs<-seawatch$CART>seawatch$POP80 | seawatch$REAG>seawatch$POP80 | seawatch$REAG>seawatch$POP80

seawatch[ex.obs,c("POP80",'REAG','ANDR','CART')]
```

```
## # A tibble: 3 x 4
##   POP80  REAG  ANDR  CART
##   <dbl> <dbl> <dbl> <dbl>
## 1 16301  4638  1572 23888
## 2 16301  4638  1572 23888
## 3 16301  4638  1572 23888
```

```
# correlation
cor(seawatch$POP80,I(seawatch$CART+seawatch$REAG+seawatch$ANDR))
```

```
## [1] 0.9533607
```

```
# plot pop80 and sum of those 3 vote numbers
plot(POP80~I(CART+REAG+ANDR),data = seawatch)
```



```
# original sd
sd(seawatch[-ex.obs,]$POP80)
```

```
## [1] 23057.15
```

```
# predictive model
## full model
pop.lm<-lm(POP80~.+I(CART+REAG+ANDR),data = seawatch[-ex.obs,])
## Predictors selection
step(pop.lm,direction = "backward",trace = 0)
```

```
##
## Call:
## lm(formula = POP80 ~ GROSS + MOY + CPI + HHMEDI + POVPR + MFGPR +
##     MAGE + CART + REAG + ANDR, data = seawatch[-ex.obs, ])
##
## Coefficients:
## (Intercept)        GROSS          MOY          CPI       HHMEDI
##  -1.515e+04   -8.491e-01   -1.790e+02    5.996e+01    1.132e-01
##       POVPR        MFGPR         MAGE         CART         REAG
##   7.209e+02    1.418e+02   -4.204e+02    1.183e+00    2.222e+00
##        ANDR
##   7.499e+00
```

```
## update model
pop.lm<-lm(formula = POP80 ~ GROSS + MOY + CPI + HHMEDI + POVPR + MFGPR +
    MAGE + CART + REAG + ANDR,data = seawatch[-ex.obs,])

## check multicolinearity
vif(pop.lm)
```

```
##     GROSS       MOY       CPI    HHMEDI     POVPR     MFGPR      MAGE
##  4.153757  1.049295  1.275316  2.534493  2.855746  1.470281  1.322949
##      CART      REAG      ANDR
##  5.099883 15.065419 29.754043
```

```
## drop ANDR
pop.lm<-update(pop.lm,.~.-ANDR)
vif(pop.lm)
```

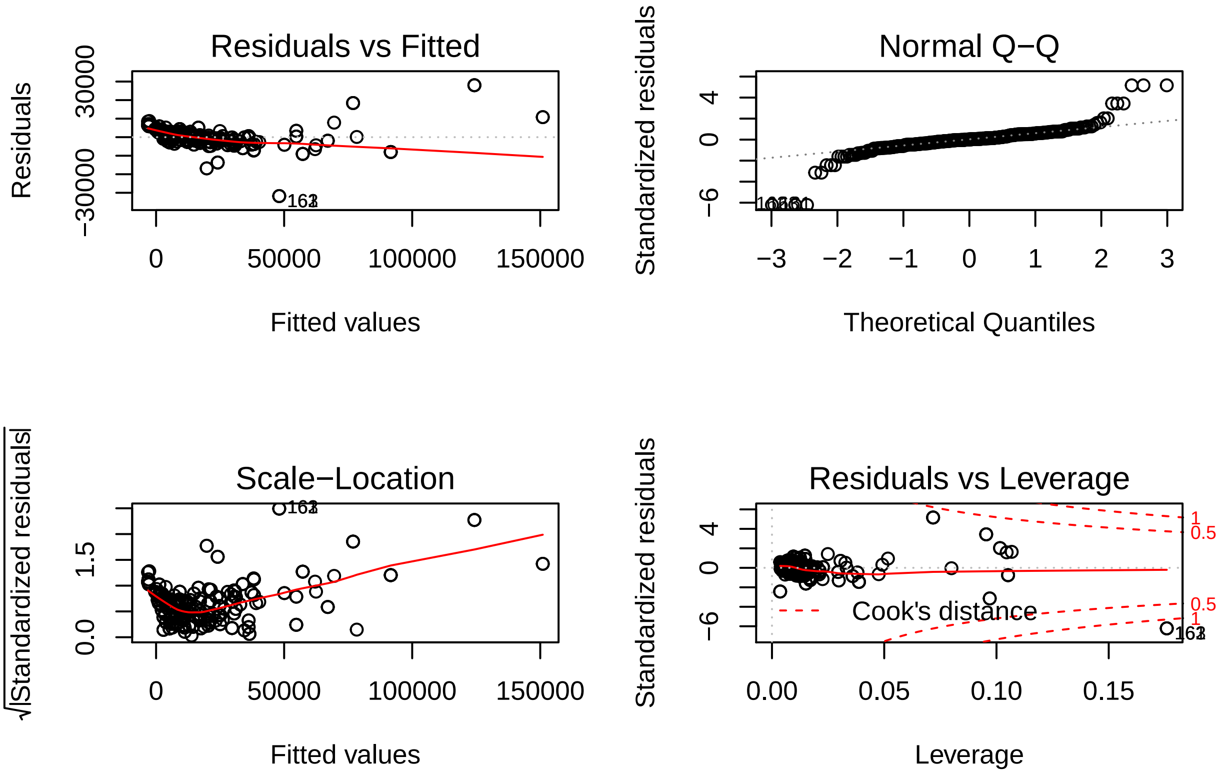
```
##    GROSS      MOY      CPI   HHMEDI    POVPR    MFGPR     MAGE     CART
## 1.729190 1.048175 1.144997 2.474438 2.671959 1.445496 1.307185 4.138947
##     REAG
## 3.836808
```

```
## summary and plot of the model
summary(pop.lm)
```

```
##
## Call:
## lm(formula = POP80 ~ GROSS + MOY + CPI + HHMEDI + POVPR + MFGPR +
##     MAGE + CART + REAG, data = seawatch[-ex.obs, ])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -32491  -1937    114   2064  27325
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  855.88035 7263.72590   0.118  0.90627
## GROSS         -0.09501    0.09354  -1.016  0.31047
## MOY         -206.87222  125.20657  -1.652  0.09938 .
## CPI           16.50456   18.91713   0.872  0.38355
## HHMEDI         0.03962    0.06937   0.571  0.56829
## POVPR        938.76317  122.01284   7.694 1.46e-13 ***
## MFGPR        108.83178   36.95402   2.945  0.00344 **
## MAGE        -502.76916  110.16717  -4.564 6.96e-06 ***
## CART           1.56789    0.11747  13.347  < 2e-16 ***
## REAG           4.19580    0.16951  24.753  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5626 on 351 degrees of freedom
## Multiple R-squared:  0.942,  Adjusted R-squared:  0.9405
## F-statistic: 632.9 on 9 and 351 DF,  p-value: < 2.2e-16
## drop GROSS, MOY, CPI and HHMEDI
pop.lm<-update(pop.lm,.~.-GROSS-MOY-CPI-HHMEDI)
summary(pop.lm)
```

```
##
## Call:
## lm(formula = POP80 ~ POVPR + MFGPR + MAGE + CART + REAG, data = seawatch[-ex.obs,
##     ])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -31778  -2049    114   2375  28029
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5141.8484  4034.3412   1.275   0.2033
## POVPR        914.1322    84.1918  10.858  < 2e-16 ***
## MFGPR        115.1605    33.6754   3.420   0.0007 ***
## MAGE        -508.0712   107.4020  -4.731 3.24e-06 ***
## CART           1.5629     0.1140  13.711  < 2e-16 ***
## REAG           4.1345     0.1648  25.096  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5632 on 355 degrees of freedom
## Multiple R-squared:  0.9412, Adjusted R-squared:  0.9403
## F-statistic:  1136 on 5 and 355 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(pop.lm)
```

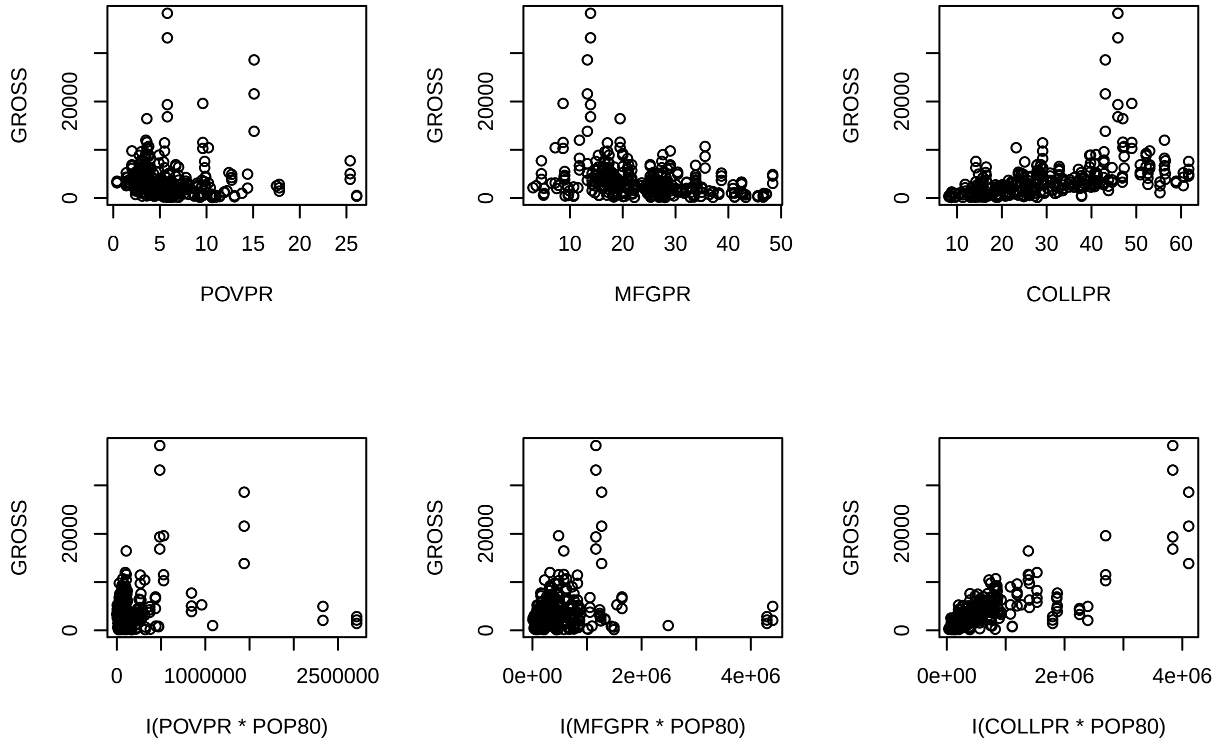


```
## predict the total population
pred<-predict(pop.lm,seawatch[ex.obs,])
seawatch[ex.obs,"POP80"]<-pred
```

POVPR,COLLPR and MFGPR

- Due to the increasing variance, instead of percentage, we transform those varibles to exact number by mutiplying total population. As a result, the linear correlations is more obvious.

```
#Compare variance before and after transformation
par(mfrow=c(2,3))
plot(GROSS~POVPR,data = seawatch)
plot(GROSS~MFGPR,data = seawatch)
plot(GROSS~COLLPR,data = seawatch)
plot(GROSS~I(POVPR*POP80),data = seawatch)
plot(GROSS~I(MFGPR*POP80),data = seawatch)
plot(GROSS~I(COLLPR*POP80),data = seawatch)
```



```r
# Add POVPP,MFGPP,COLLPP to data
seawatch$POVPP<-seawatch$POVPR*seawatch$POP80
seawatch$MFGPP<-seawatch$MFGPR*seawatch$POP80
seawatch$COLLPP<-seawatch$COLLPR*seawatch$POP80
```

Modeling

Training and Testing subsets split

```r
set.seed(1024)
train.num<-sample(1:dim(seawatch)[1],round(nrow(seawatch)*0.75))
seawatch.train<-seawatch[train.num,]
seawatch.test<-seawatch[-train.num,]
```

Predictors Selections

```r
# full model
full.lm<-lm(data = seawatch.train,GROSS~.)

# predictors selection
step(full.lm,direction = "backward",trace = 0)
```

```
##
```

```
## Call:
## lm(formula = GROSS ~ MOY + YR + VISIT + LST + HHMEDI + CART +
##     REAG + ANDR + POVPP + COLLPP, data = seawatch.train)
##
## Coefficients:
## (Intercept)          MOY           YR       VISIT2       VISIT3
##  -5.801e+04    9.315e+01    6.935e+02   -1.625e+02    4.430e+02
##      VISIT4       VISIT5         LST1       HHMEDI         CART
##   2.590e+03    1.700e+04   -7.933e+02    6.632e-02   -3.096e-01
##        REAG         ANDR        POVPP       COLLPP
##  -4.408e-01    2.561e+00   -1.915e-03    3.714e-03
```

```
# update model
fit.lm<-lm(formula = GROSS ~ MOY + YR + VISIT + LST + HHMEDI + CART +
    REAG + ANDR + POVPP + COLLPP, data = seawatch.train)

# check multicolinearity
vif(fit.lm)
```

```
##             GVIF Df GVIF^(1/(2*Df))
## MOY     1.182588  1        1.087469
## YR      3.647111  1        1.909741
## VISIT   3.691875  4        1.177351
## LST     2.093257  1        1.446809
## HHMEDI  2.119586  1        1.455880
## CART   11.627003  1        3.409839
## REAG   19.918715  1        4.463039
## ANDR   41.908423  1        6.473672
## POVPP   4.223934  1        2.055221
## COLLPP 18.224195  1        4.268981
```

```
# drop ANDR
fit.lm<-update(fit.lm,.~.-ANDR)
vif(fit.lm)
```

```
##            GVIF Df GVIF^(1/(2*Df))
## MOY    1.169804  1        1.081575
## YR     3.386926  1        1.840360
## VISIT  3.483543  4        1.168833
## LST    2.057111  1        1.434263
## HHMEDI 1.656580  1        1.287082
## CART   9.243068  1        3.040241
## REAG   3.974517  1        1.993619
## POVPP  3.560529  1        1.886937
## COLLPP 5.917441  1        2.432579
```

```
# drop CART
fit.lm<-update(fit.lm,.~.-CART)
vif(fit.lm)
```

```
##            GVIF Df GVIF^(1/(2*Df))
## MOY    1.169804  1        1.081575
## YR     3.268160  1        1.807805
## VISIT  3.211092  4        1.156995
## LST    2.048469  1        1.431247
## HHMEDI 1.651053  1        1.284933
## REAG   3.521195  1        1.876485
## POVPP  2.650217  1        1.627949
## COLLPP 2.842762  1        1.686049
```

```
# summary
summary(fit.lm)
```

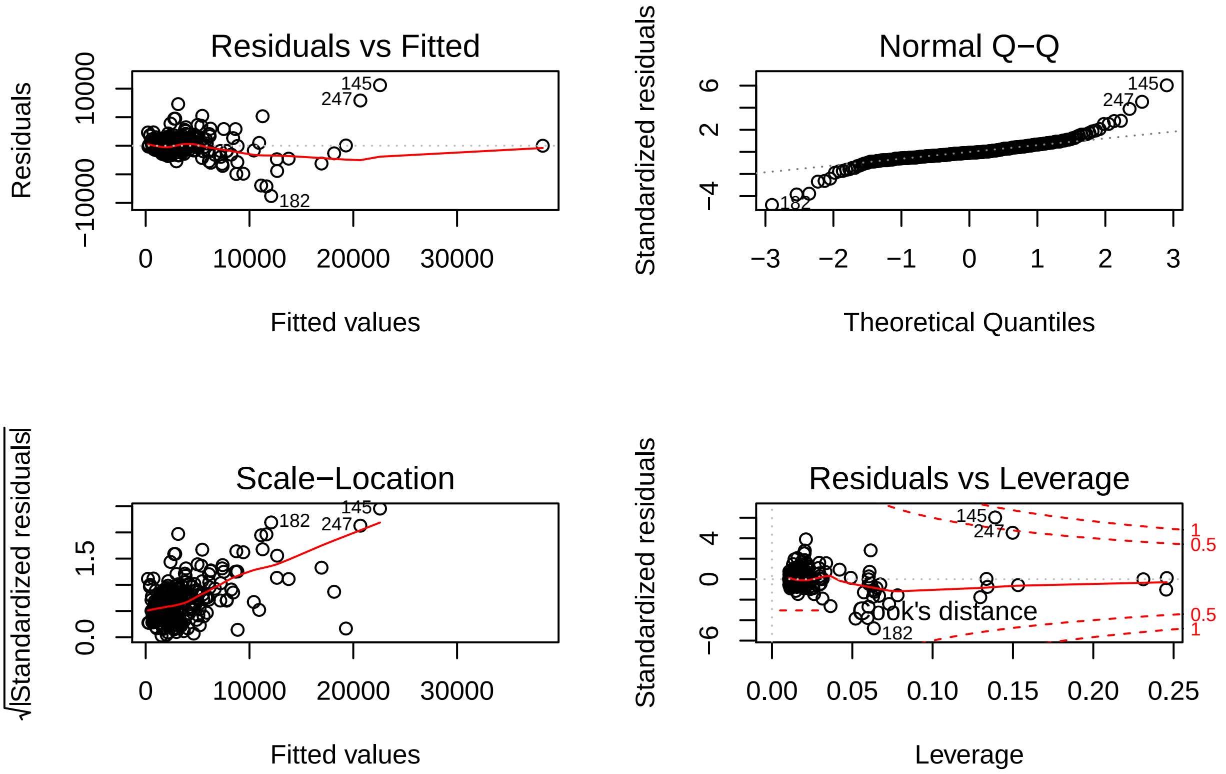
```
##
## Call:
## lm(formula = GROSS ~ MOY + YR + VISIT + LST + HHMEDI + REAG +
##     POVPP + COLLPP, data = seawatch.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8740.9  -747.8  -177.7   700.2 10321.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.377e+03  1.717e+04  -0.546  0.58539
## MOY          6.628e+01  5.079e+01   1.305  0.19304
## YR           1.128e+02  2.081e+02   0.542  0.58823
## VISIT2       3.916e+02  3.193e+02   1.227  0.22106
## VISIT3       1.459e+03  4.404e+02   3.313  0.00105 **
## VISIT4       4.772e+03  6.767e+02   7.052 1.59e-11 ***
## VISIT5       2.079e+04  2.163e+03   9.611  < 2e-16 ***
## LST1        -7.261e+02  3.343e+02  -2.172  0.03074 *
## HHMEDI       1.641e-02  2.220e-02   0.739  0.46042
## REAG         7.518e-02  6.352e-02   1.184  0.23763
## POVPP       -2.824e-03  5.347e-04  -5.282 2.70e-07 ***
## COLLPP       4.558e-03  2.829e-04  16.114  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1898 on 260 degrees of freedom
## Multiple R-squared:  0.8164, Adjusted R-squared:  0.8086
## F-statistic: 105.1 on 11 and 260 DF,  p-value: < 2.2e-16
```

```
# drop YR,POVPR,PERCAPI
fit.lm<-update(fit.lm,.~.-YR-MOY-HHMEDI-REAG)
summary(fit.lm)
```

```
##
## Call:
## lm(formula = GROSS ~ VISIT + LST + POVPP + COLLPP, data = seawatch.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8804.8  -785.8  -157.8   762.9 10598.9
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.253e+02  2.156e+02   3.364 0.000882 ***
## VISIT2       5.104e+02  2.767e+02   1.845 0.066217 .
## VISIT3       1.644e+03  3.577e+02   4.597 6.64e-06 ***
## VISIT4       4.929e+03  5.648e+02   8.726 3.03e-16 ***
## VISIT5       2.128e+04  2.068e+03  10.292  < 2e-16 ***
## LST1        -6.765e+02  2.774e+02  -2.439 0.015398 *
## POVPP        -2.618e-03  4.053e-04  -6.460 5.01e-10 ***
## COLLPP        4.740e-03  2.299e-04  20.617  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1896 on 264 degrees of freedom
## Multiple R-squared:  0.8141, Adjusted R-squared:  0.8091
## F-statistic: 165.1 on 7 and 264 DF,  p-value: < 2.2e-16
```

```
# residual analysis
par(mfrow=c(2,2))
plot(fit.lm)
```

```
## Warning: not plotting observations with leverage one:
##   152

## Warning: not plotting observations with leverage one:
##   152
```



- There exists clear non-constant variance. Also predictors are not normally distributed. As a result, we use the power transformation to modify the model.

Power Transformation model

```
powerTransform(cbind(seawatch.train$GROSS,seawatch$POVPR,seawatch$COLLPP)~1)
```

```
## Warning in cbind(seawatch.train$GROSS, seawatch$POVPR, seawatch$COLLPP):
## number of rows of result is not a multiple of vector length (arg 1)
```

```
## Estimated transformation parameters
##          Y1          Y2          Y3
##  0.12458545  0.10717096 -0.05498497
```

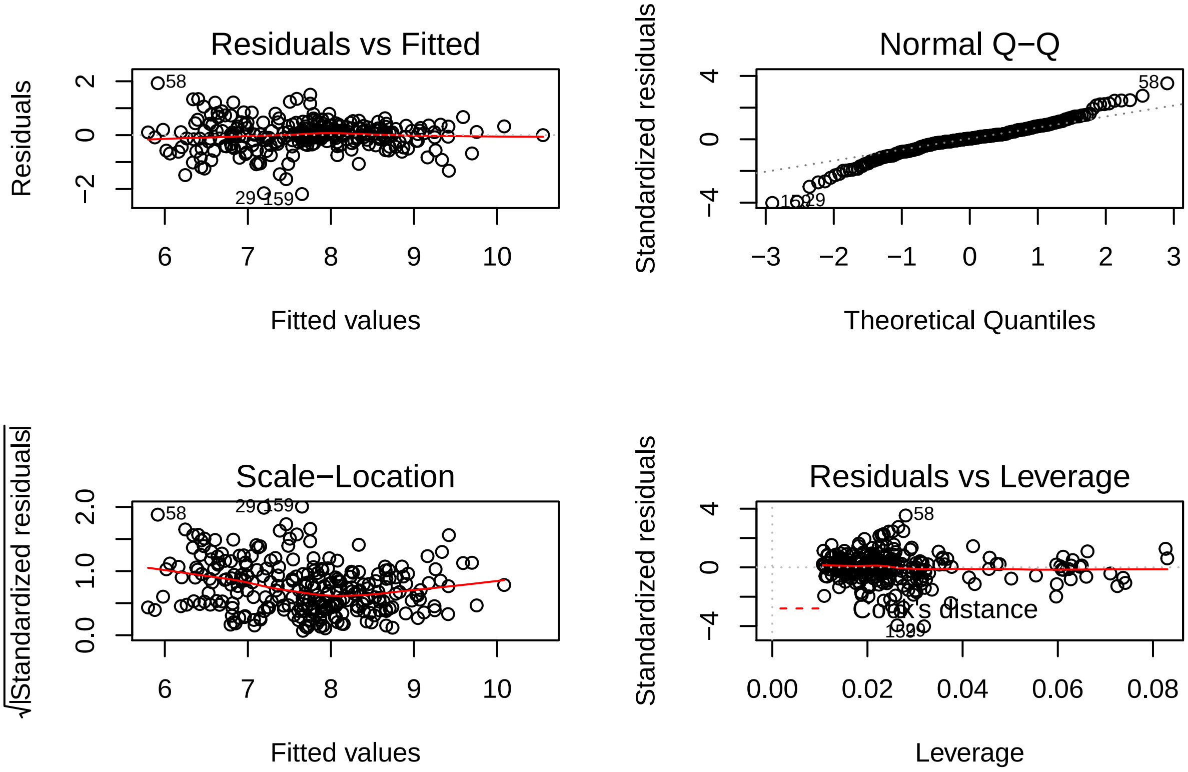
```
# Thus, we build another model by taking natural log on both sides
new.fit.lm<-lm(formula = log(GROSS) ~ VISIT + LST + log(POVPP) + log(COLLPP), data = seawatch.train)
summary(new.fit.lm)
```

```
##
## Call:
## lm(formula = log(GROSS) ~ VISIT + LST + log(POVPP) + log(COLLPP),
##     data = seawatch.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.18775 -0.22711  0.02269  0.27381  1.92732
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.93422    0.47035  -1.986  0.04804 *
## VISIT2       0.17483    0.08055   2.170  0.03087 *
## VISIT3       0.51467    0.10682   4.818 2.45e-06 ***
## VISIT4       0.84580    0.16582   5.101 6.47e-07 ***
## VISIT5       1.57043    0.57016   2.754  0.00629 **
## LST1        -0.25719    0.08260  -3.114  0.00205 **
## log(POVPP)  -0.27453    0.03869  -7.097 1.18e-11 ***
## log(COLLPP)  0.90809    0.04800  18.919  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5531 on 264 degrees of freedom
## Multiple R-squared:  0.7195, Adjusted R-squared:  0.7121
## F-statistic: 96.76 on 7 and 264 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(new.fit.lm)
```

```
## Warning: not plotting observations with leverage one:
##   152
```

```
## Warning: not plotting observations with leverage one:
##   152
```



- Note that the variance of error is more constant and the predictors are distributed better than the original linear model

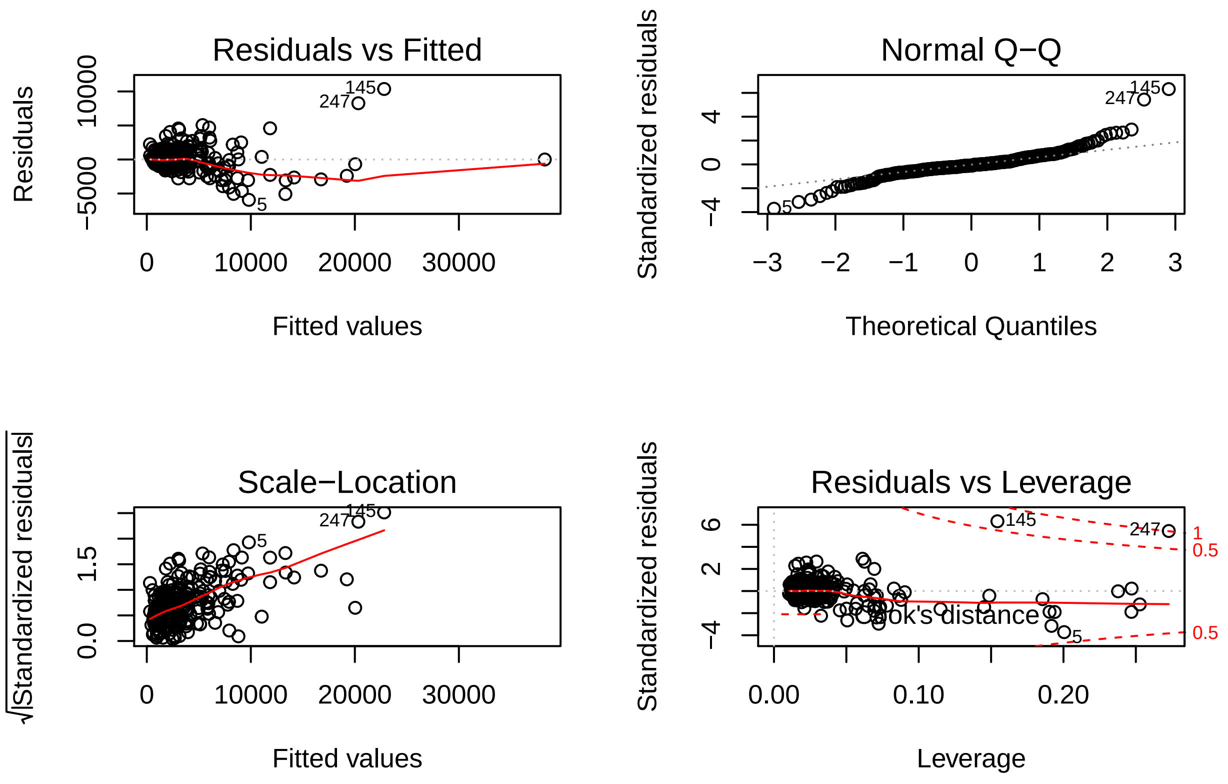
Model based on correlation

```r
model2<- lm(GROSS ~ VISIT + POP80 + PERCAPI + MFGPR + REAG + POVPP + COLLPP,data = seawatch.train)
summary(model2)
```

```
##
## Call:
## lm(formula = GROSS ~ VISIT + POP80 + PERCAPI + MFGPR + REAG +
##     POVPP + COLLPP, data = seawatch.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5930.0  -750.4  -166.7   698.2 10360.4
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.502e+02  6.698e+02  -0.224 0.822673
## VISIT2       3.314e+02  2.591e+02   1.279 0.202013
## VISIT3       1.137e+03  2.996e+02   3.795 0.000184 ***
## VISIT4       3.919e+03  5.029e+02   7.792 1.57e-13 ***
## VISIT5       1.936e+04  1.918e+03  10.092  < 2e-16 ***
## POP80       -2.365e-01  3.756e-02  -6.296 1.29e-09 ***
## PERCAPI      8.191e-03  6.120e-02   0.134 0.893638
## MFGPR        2.078e+01  1.296e+01   1.603 0.110085
## REAG         7.606e-01  1.248e-01   6.094 3.94e-09 ***
## POVPP        4.770e-03  1.284e-03   3.714 0.000249 ***
## COLLPP       6.293e-03  3.888e-04  16.186  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1784 on 261 degrees of freedom
## Multiple R-squared:  0.8372, Adjusted R-squared:  0.831
## F-statistic: 134.3 on 10 and 261 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(model2)
```

```
## Warning: not plotting observations with leverage one:
##   152

## Warning: not plotting observations with leverage one:
##   152
```



```
#multicolinearity check
vif(model2)
```

```
##              GVIF Df GVIF^(1/(2*Df))
## VISIT    1.343200  4        1.037570
## POP80   65.917474  1        8.118958
## PERCAPI  1.590806  1        1.261272
## MFGPR    1.324058  1        1.150677
## REAG    15.398087  1        3.924040
## POVPP   17.316278  1        4.161283
## COLLPP   6.080990  1        2.465966
```

```
model2<-update(model2,.~.-POP80)
vif(model2)
```

```
##             GVIF Df GVIF^(1/(2*Df))
## VISIT   1.322948  4        1.035602
## PERCAPI 1.510947  1        1.229206
## MFGPR   1.184886  1        1.088525
## REAG    3.239980  1        1.799994
## POVPP   2.450437  1        1.565387
## COLLPP  3.022401  1        1.738505
```

```
#powertransformation
powerTransform(cbind(seawatch.train$GROSS,seawatch.train$PERCAPI,seawatch.train$MFGPR,seawatch.train$REA
```

```
## Estimated transformation parameters
##           Y1           Y2           Y3           Y4           Y5
##  0.143119027 -1.417761071  0.645988351  0.205752122  0.058852475
##           Y6
##  0.002208676
```

```
#model2
model2<-lm(log(GROSS) ~ VISIT + 1/PERCAPI + sqrt(MFGPR) + log(REAG) + log(POVPP) + log(COLLPP),data = se
summary(model2)
```

```
##
## Call:
## lm(formula = log(GROSS) ~ VISIT + 1/PERCAPI + sqrt(MFGPR) + log(REAG) +
##     log(POVPP) + log(COLLPP), data = seawatch.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.41640 -0.29439  0.05938  0.29932  1.46012
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.66788    0.60186  -1.110 0.268140
## VISIT2       0.15328    0.08056   1.903 0.058152 .
## VISIT3       0.36345    0.09370   3.879 0.000133 ***
## VISIT4       0.62299    0.15366   4.054 6.63e-05 ***
## VISIT5       1.24336    0.56763   2.190 0.029372 *
## sqrt(MFGPR) -0.09132    0.03720  -2.455 0.014732 *
## log(REAG)   -0.03229    0.10592  -0.305 0.760711
## log(POVPP)  -0.28518    0.04374  -6.520 3.58e-10 ***
## log(COLLPP)  0.94737    0.08122  11.664  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5564 on 263 degrees of freedom
## Multiple R-squared:  0.7172, Adjusted R-squared:  0.7086
## F-statistic: 83.36 on 8 and 263 DF,  p-value: < 2.2e-16
```

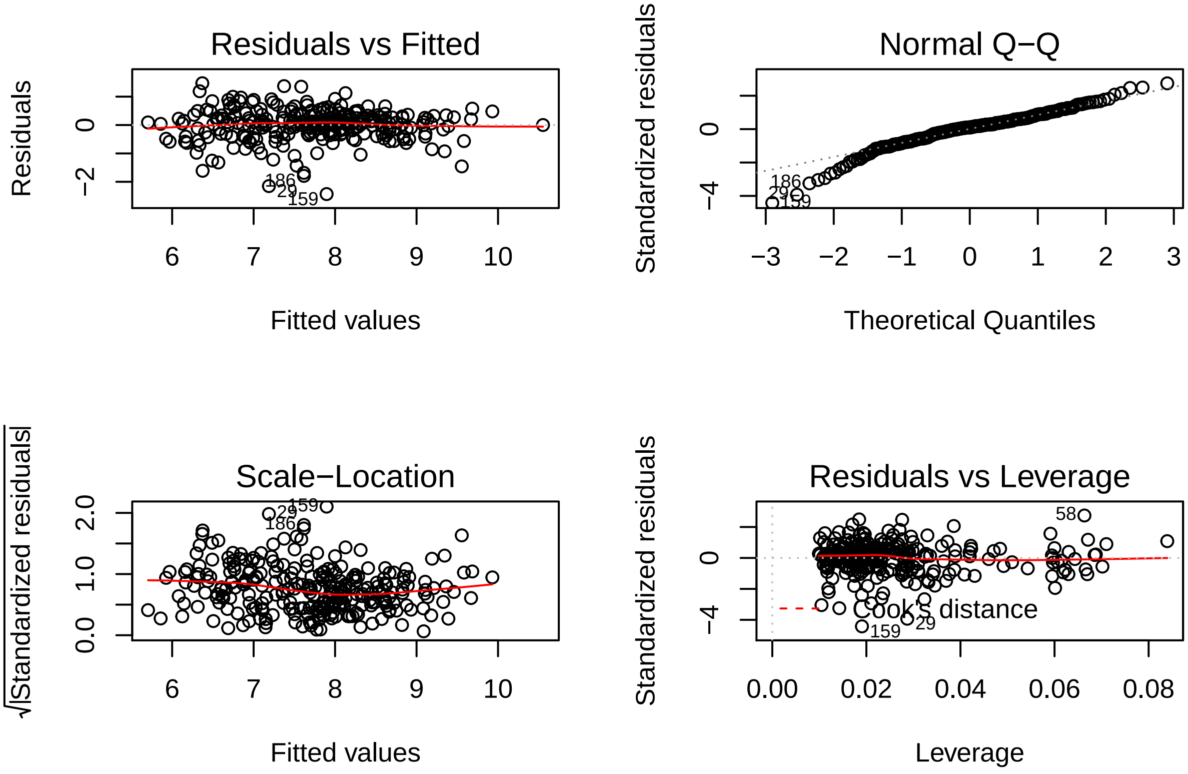
```
model2<-update(model2,.~.-log(REAG))
summary(model2)
```

```
##
## Call:
## lm(formula = log(GROSS) ~ VISIT + sqrt(MFGPR) + log(POVPP) +
##     log(COLLPP), data = seawatch.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.43230 -0.29927  0.05704  0.31874  1.47228
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.57832    0.52439  -1.103 0.271094
## VISIT2       0.15500    0.08022   1.932 0.054418 .
## VISIT3       0.36555    0.09329   3.918 0.000114 ***
## VISIT4       0.62705    0.15282   4.103 5.43e-05 ***
## VISIT5       1.24861    0.56640   2.204 0.028353 *
## sqrt(MFGPR) -0.09498    0.03515  -2.702 0.007342 **
## log(POVPP)  -0.29182    0.03787  -7.707 2.62e-13 ***
## log(COLLPP)  0.92717    0.04687  19.781  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5555 on 264 degrees of freedom
## Multiple R-squared:  0.7171, Adjusted R-squared:  0.7096
## F-statistic: 95.58 on 7 and 264 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(model2)
```

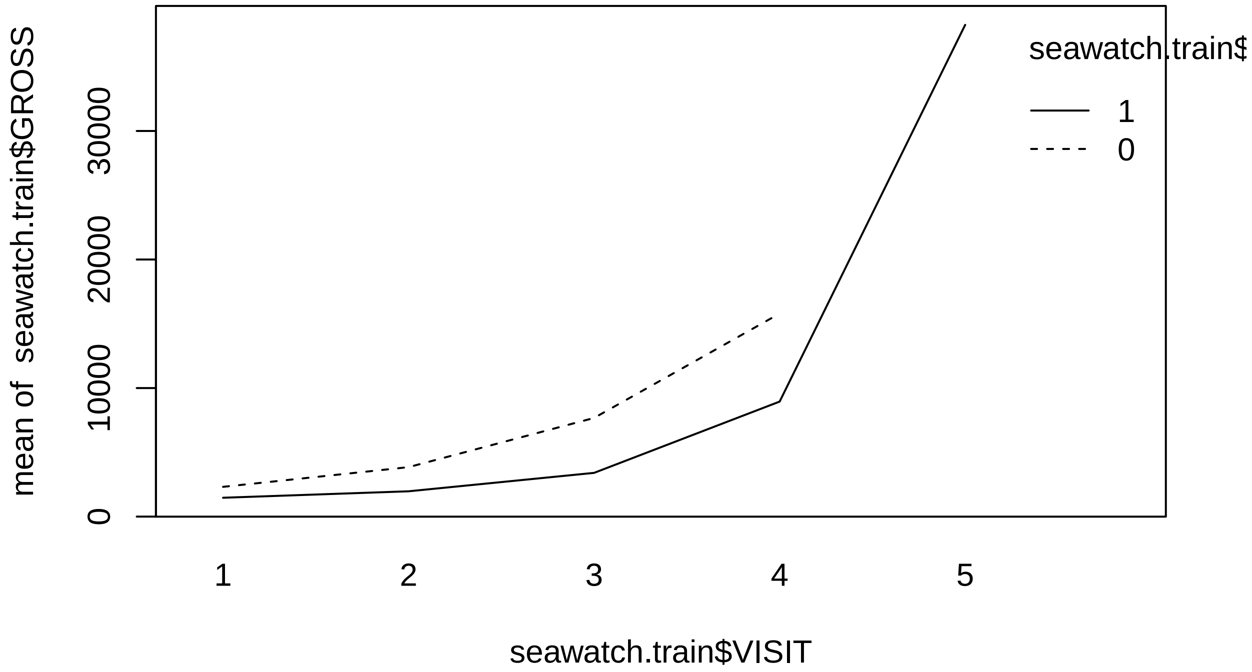
```
## Warning: not plotting observations with leverage one:
##   152
```

```
## Warning: not plotting observations with leverage one:
##   152
```



model with interaction

```
# Interaction plot
interaction.plot(seawatch.train$VISIT,seawatch.train$LST,seawatch.train$GROSS)
```



```
interaction.lm<-lm(data = seawatch.train,GROSS~MOY + YR + VISIT + LST + HHMEDI + CART +  REAG + ANDR + I

# Power transformation
powerTransform(cbind(seawatch.train$GROSS,seawatch.train$MOY,seawatch.train$HHMEDI,seawatch.train$CART,s
```

```
## Estimated transformation parameters
##           Y1           Y2           Y3           Y4           Y5
##  0.142124500  0.979063506 -0.101273718  0.001854217  0.168322097
##           Y6           Y7           Y8
##  0.113241384  0.053069424 -0.016952640
```

```
interaction.lm<-lm(data = seawatch.train,log(GROSS)~MOY + YR + VISIT + LST + log(HHMEDI) + log(CART) +

#predictors selection
step(interaction.lm,trace = 0)
```

```
##
## Call:
## lm(formula = log(GROSS) ~ VISIT + LST + log(CART) + log(REAG) +
##     log(ANDR) + log(POVPP) + log(COLLPP) + VISIT:log(REAG) +
##     VISIT:log(COLLPP) + LST:log(YR) + LST:log(CART) + LST:log(REAG) +
##     LST:log(COLLPP), data = seawatch.train)
##
## Coefficients:
##       (Intercept)             VISIT2             VISIT3
##           10.8365            -0.3555             0.3570
##            VISIT4             VISIT5               LST1
##            -1.7746              1.3286            -61.6314
##          log(CART)           log(REAG)           log(ANDR)
##            -0.3339             -0.3450              0.8543
##         log(POVPP)         log(COLLPP)    VISIT2:log(REAG)
##            -0.2195              0.7354              0.4256
##   VISIT3:log(REAG)    VISIT4:log(REAG)    VISIT5:log(REAG)
##             0.8192              0.9483                  NA
## VISIT2:log(COLLPP)  VISIT3:log(COLLPP)  VISIT4:log(COLLPP)
##            -0.2261             -0.5094             -0.4251
## VISIT5:log(COLLPP)        LST0:log(YR)        LST1:log(YR)
##                 NA             -2.4555             11.1822
##     LST1:log(CART)      LST1:log(REAG)    LST1:log(COLLPP)
##            -0.2702             -0.6843              0.6999
```

```
interaction.lm<-lm(formula = log(GROSS) ~ VISIT + LST + log(CART) + log(REAG) +
    log(ANDR) + log(POVPP) + log(COLLPP) + VISIT:log(REAG) +
    VISIT:log(COLLPP) + LST:log(YR) + LST:log(CART) + LST:log(REAG) +
    LST:log(COLLPP), data = seawatch.train)
# Delete ANDR and CART due to high colinear
vif(lm(formula = log(GROSS) ~ VISIT + LST + log(CART) + log(REAG) +
    log(ANDR) + log(POVPP) + log(COLLPP),data=seawatch.train))
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## VISIT        1.753425  4        1.072719
## LST          1.516618  1        1.231511
## log(CART)   10.643074  1        3.262372
## log(REAG)   14.918347  1        3.862428
## log(ANDR)   28.902853  1        5.376137
## log(POVPP)   4.827899  1        2.197248
## log(COLLPP) 10.529501  1        3.244919
```

```
interaction.lm<-update(interaction.lm,.~.-log(ANDR)-log(CART)-log(REAG))
vif(lm(formula = log(GROSS) ~ VISIT + LST + log(POVPP) + log(COLLPP),data=seawatch.train))
```

```
##                  GVIF Df GVIF^(1/(2*Df))
## VISIT       1.662805  4        1.065627
## LST         1.473458  1        1.213861
## log(POVPP)  1.739047  1        1.318729
## log(COLLPP) 1.977095  1        1.406092
```

```
# Summary
summary(interaction.lm)
```

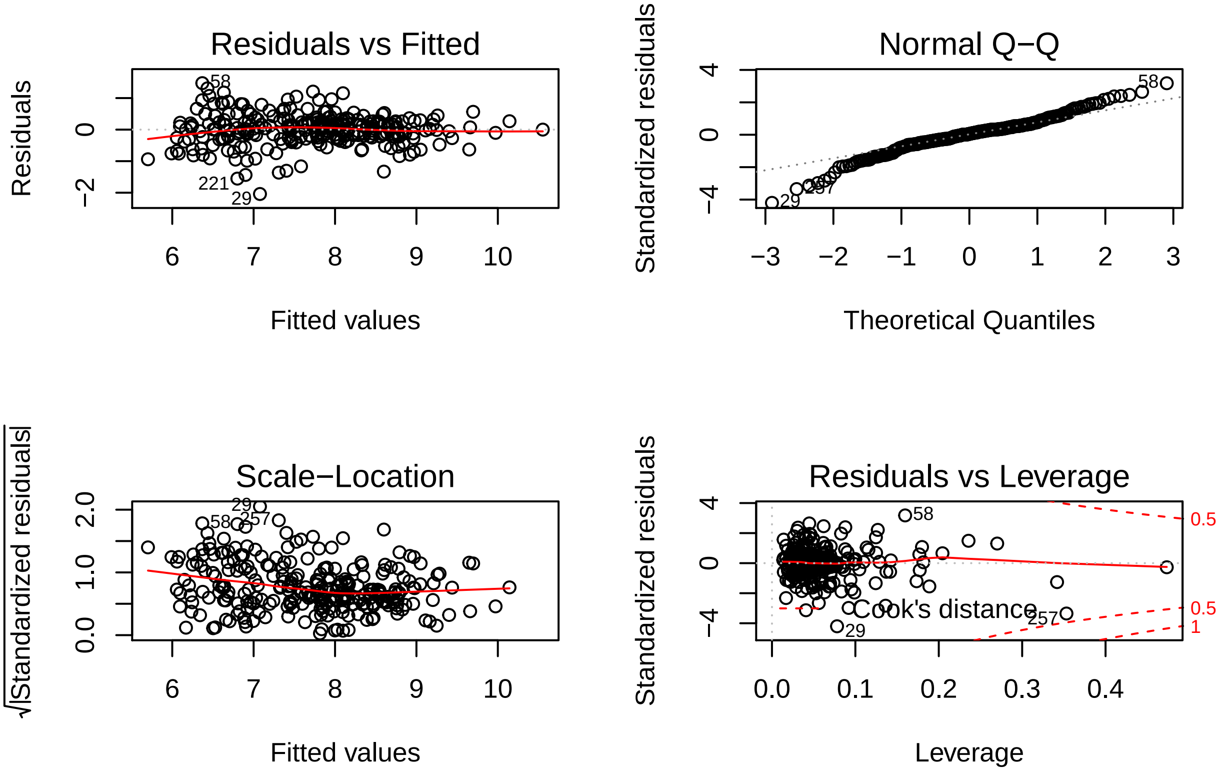
```
##
## Call:
## lm(formula = log(GROSS) ~ VISIT + LST + log(POVPP) + log(COLLPP) +
##     VISIT:log(REAG) + VISIT:log(COLLPP) + LST:log(YR) + LST:log(CART) +
##     LST:log(REAG) + LST:log(COLLPP), data = seawatch.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.99511 -0.22558  0.02319  0.22924  1.48289
##
## Coefficients: (2 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          24.272905  24.787894   0.979 0.328412
## VISIT2              -0.486731   1.038692  -0.469 0.639763
## VISIT3               0.001534   1.329173   0.001 0.999080
## VISIT4              -1.975110   2.484788  -0.795 0.427434
## VISIT5               2.259105   1.608448   1.405 0.161399
## LST1               -62.300339  34.844070  -1.788 0.074986 .
## log(POVPP)          -0.117084   0.055334  -2.116 0.035335 *
## log(COLLPP)          0.998388   0.127257   7.845 1.24e-13 ***
## VISIT1:log(REAG)     0.070206   0.152628   0.460 0.645927
## VISIT2:log(REAG)     0.472828   0.170658   2.771 0.006013 **
## VISIT3:log(REAG)     0.900123   0.255660   3.521 0.000511 ***
## VISIT4:log(REAG)     0.833409   0.414798   2.009 0.045587 *
## VISIT5:log(REAG)           NA         NA      NA       NA
## VISIT2:log(COLLPP)  -0.199478   0.163610  -1.219 0.223902
## VISIT3:log(COLLPP)  -0.484505   0.214585  -2.258 0.024813 *
## VISIT4:log(COLLPP)  -0.281602   0.345664  -0.815 0.416033
## VISIT5:log(COLLPP)         NA         NA      NA       NA
## LST0:log(YR)        -5.944179   5.564870  -1.068 0.286474
## LST1:log(YR)         7.959812   6.399207   1.244 0.214706
## LST0:log(CART)      -0.323994   0.120376  -2.692 0.007591 **
## LST1:log(CART)      -0.621472   0.144371  -4.305 2.40e-05 ***
## LST1:log(REAG)      -0.593899   0.245864  -2.416 0.016426 *
## LST1:log(COLLPP)     0.622829   0.165288   3.768 0.000205 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5047 on 251 degrees of freedom
## Multiple R-squared:  0.7779, Adjusted R-squared:  0.7602
## F-statistic: 43.97 on 20 and 251 DF,  p-value: < 2.2e-16
```

```
# drop VISIT and log(POVPP)
interaction.lm<-update(interaction.lm,.~.-VISIT-log(POVPP))
summary(interaction.lm)
```

```
##
## Call:
## lm(formula = log(GROSS) ~ LST + log(COLLPP) + VISIT:log(REAG) +
##     VISIT:log(COLLPP) + LST:log(YR) + LST:log(CART) + LST:log(REAG) +
##     LST:log(COLLPP), data = seawatch.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.04089 -0.23343  0.02387  0.25991  1.47342
##
## Coefficients: (1 not defined because of singularities)
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          30.00053   24.24917   1.237 0.217159
## LST1                -63.22464   34.57876  -1.828 0.068654 .
## log(COLLPP)           1.05626    0.10903   9.688  < 2e-16 ***
## VISIT1:log(REAG)      0.04575    0.14842   0.308 0.758149
## VISIT2:log(REAG)      0.47390    0.16882   2.807 0.005386 **
## VISIT3:log(REAG)      0.84991    0.25125   3.383 0.000831 ***
## VISIT4:log(REAG)      0.90017    0.40937   2.199 0.028781 *
## VISIT5:log(REAG)      0.20828    0.15853   1.314 0.190075
## log(COLLPP):VISIT2   -0.25185    0.11199  -2.249 0.025372 *
## log(COLLPP):VISIT3   -0.46487    0.15620  -2.976 0.003200 **
## log(COLLPP):VISIT4  -0.47961    0.25227  -1.901 0.058406 .
## log(COLLPP):VISIT5        NA         NA      NA       NA
## LST0:log(YR)        -7.42012    5.43890  -1.364 0.173686
## LST1:log(YR)         6.67108    6.38468   1.045 0.297078
## LST0:log(CART)      -0.46420    0.09990  -4.647 5.41e-06 ***
## LST1:log(CART)      -0.75559    0.12527  -6.032 5.67e-09 ***
## LST1:log(REAG)      -0.59373    0.24244  -2.449 0.015002 *
## LST1:log(COLLPP)     0.62645    0.15968   3.923 0.000112 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5058 on 255 degrees of freedom
## Multiple R-squared:  0.7734, Adjusted R-squared:  0.7592
## F-statistic: 54.39 on 16 and 255 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(interaction.lm)
```

```
## Warning: not plotting observations with leverage one:
##   152

## Warning: not plotting observations with leverage one:
##   152
```



Cross Validation

```
#MSE function
MSE<-function(pred,actual){
  return(mean((pred-actual)^2))
}

#predictions based on each model

pred.test<-predict(update(new.fit.lm,.~+VISIT:LST),newdata = seawatch.test)
```

```
## Warning in predict.lm(update(new.fit.lm, . ~ +VISIT:LST), newdata =
## seawatch.test): prediction from a rank-deficient fit may be misleading
```

```
pred.fit<-predict(fit.lm,newdata = seawatch.test)
pred.fit.new<-predict(new.fit.lm,newdata = seawatch.test)
pred.model2<-predict(model2,newdata = seawatch.test)
pred.interaction<-predict(interaction.lm,newdata = seawatch.test)
```

```
## Warning in predict.lm(interaction.lm, newdata = seawatch.test): prediction
## from a rank-deficient fit may be misleading
```

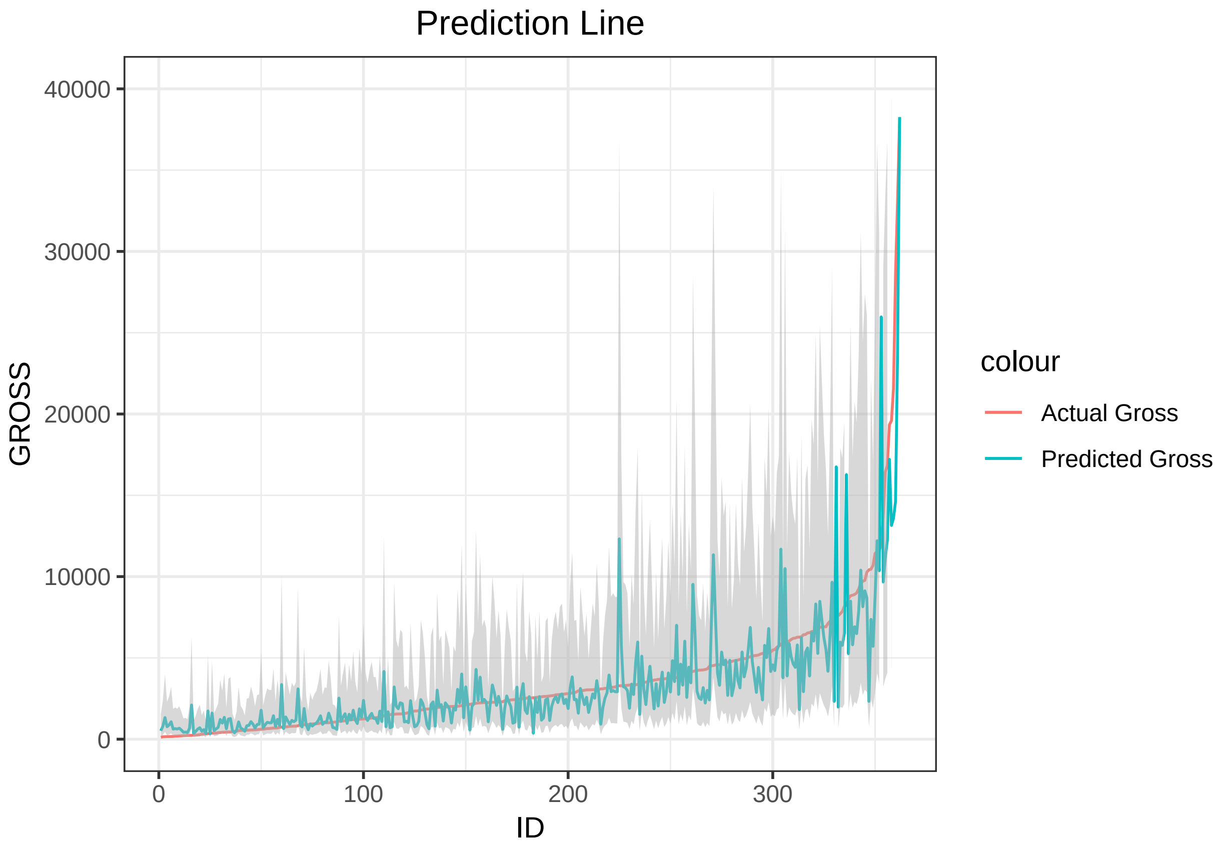
```
#MSE table
data.frame(
  Model=c("fit.lm","fit.powertrans","model2","interaction model","predict test"),
  MSE=c(MSE(pred.fit,seawatch.test$GROSS),MSE(exp(pred.fit.new),seawatch.test$GROSS),MSE(exp(pred.model
```

```
##               Model      MSE
## 1            fit.lm  8248999
## 2    fit.powertrans  5771242
## 3            model2  5959332
## 4 interaction model 13573104
## 5      predict test 25648547
```

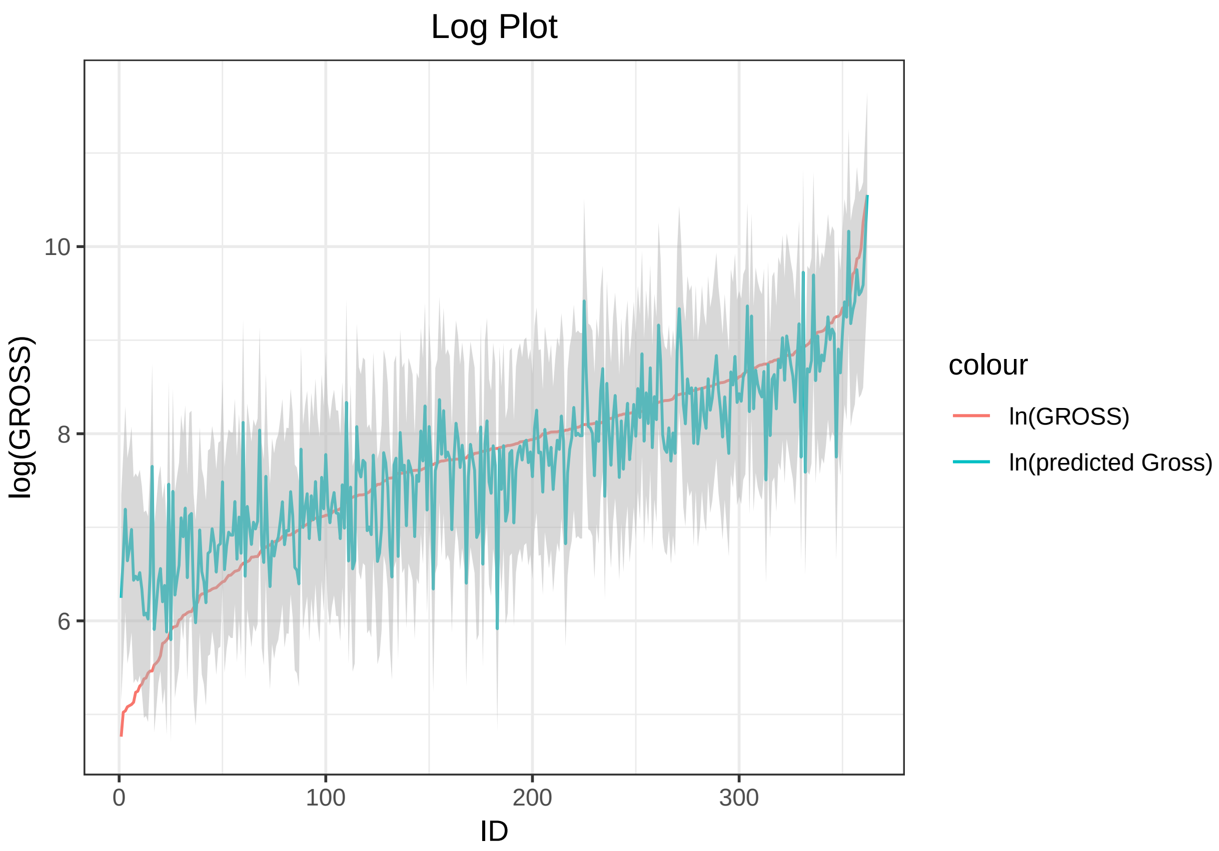
- The model with lowest MSE so far is the log-transformation model.

Visualization

```
seawatch<-seawatch[order(seawatch$GROSS),]
seawatch$pred.value<-exp(predict(new.fit.lm,newdata = seawatch))
seawatch$upper.int<-seawatch$pred.value*3
seawatch$lower.int<-seawatch$pred.value/3
seawatch$ID<-c(1:nrow(seawatch))


#Actual data vs. predicted
ggplot(data=seawatch,aes(x=ID,y=GROSS))+
  geom_line(aes(y=GROSS,color="Actual Gross"))+
  geom_line(aes(y=pred.value, color="Predicted Gross"))+
  theme_bw()+
  geom_ribbon(aes(ymin = pred.value/3, ymax = pred.value*3), fill="grey70",alpha=0.5)+
  ylim(c(0,40000))+
  ggtitle("Prediction Line") +
  theme(plot.title = element_text(hjust = 0.5))
```



```
#take log
ggplot(data=seawatch,aes(x=ID,y=log(GROSS)))+
  geom_line(aes(y=log(GROSS),color="ln(GROSS)"))+
  geom_line(aes(y=log(pred.value), color="ln(predicted Gross)"))+
  theme_bw()+
  geom_ribbon(aes(ymin = log(lower.int), ymax = log(upper.int)), fill="grey70",alpha=0.5)+
  ggtitle("Log Plot") +
  theme(plot.title = element_text(hjust = 0.5))
```


Log Plot
log(GROSS)
10
8
6
0
100
200
300
ID
colour
ln(GROSS)
ln(predicted Gross)