# linear_regression

*Tom Thorpe*

*June 21, 2018*

## Linear Regression homework

First linear regression homework:

```
#  Introduction
## ==============

#   - Learning objectives:
##     - Learn the R formula interface
##     - Specify factor contrasts to test specific hypotheses
##     - Perform model comparisons
##     - Run and interpret variety of regression models in R

## Set working directory
## -------------------------

##   It is often helpful to start your R session by setting your working
##   directory so you don't have to type the full path names to your data
##   and other files

# set the working directory
# setwd("~/Desktop/Rstatistics")
# setwd("C:/Users/dataclass/Desktop/Rstatistics")
setwd("C:/Users/Tom/git/datasciencefoundation/ForestCoverage/linear_regression")

##   You might also start by listing the files in your working directory

getwd() # where am I?
```

```
## [1] "C:/Users/Tom/git/datasciencefoundation/ForestCoverage/linear_regression"
```

```
list.files("dataSets") # files in the dataSets folder
```

```
## [1] "Exam.rds"   "states.dta" "states.rds"
```

```
## Load the states data
## ------------------------

# read the states data
states.data <- readRDS("dataSets/states.rds")
#get labels
states.info <- data.frame(attributes(states.data)[c("names", "var.labels")])
str(states.data)
```

```
## 'data.frame':    51 obs. of  21 variables:
##  $ state  : chr  "Alabama" "Alaska" "Arizona" "Arkansas" ...
##  $ region : Factor w/ 4 levels "West","N. East",..: 3 1 1 3 1 1 2 3 NA 3 ...
##  $ pop    : num  4041000 550000 3665000 2351000 29760000 ...
##  $ area   : num  52423 570374 113642 52075 155973 ...
```

```
##  $ density: num  77.08 0.96 32.25 45.15 190.8 ...
##  $ metro  : num  67.4 41.1 79 40.1 95.7 ...
##  $ waste  : num  1.11 0.91 0.79 0.85 1.51 ...
##  $ energy : int  393 991 258 330 246 273 234 349 NA 237 ...
##  $ miles  : num  10.5 7.2 9.7 8.9 8.7 ...
##  $ toxic  : num  27.86 37.41 19.65 24.6 3.26 ...
##  $ green  : num  29.2 NA 18.4 26 15.6 ...
##  $ house  : int  30 0 13 25 50 36 64 69 NA 45 ...
##  $ senate : int  10 20 33 37 47 58 87 83 NA 47 ...
##  $ csat   : int  991 920 932 1005 897 959 897 892 840 882 ...
##  $ vsat   : int  476 439 442 482 415 453 429 428 405 416 ...
##  $ msat   : int  515 481 490 523 482 506 468 464 435 466 ...
##  $ percent: int  8 41 26 6 47 29 81 61 71 48 ...
##  $ expense: int  3627 8330 4309 3700 4491 5064 7602 5865 9259 5276 ...
##  $ income : num  27.5 48.3 32.1 24.6 41.7 ...
##  $ high   : num  66.9 86.6 78.7 66.3 76.2 ...
##  $ college: num  15.7 23 20.3 13.3 23.4 ...
##  - attr(*, "datalabel")= chr "U.S. states data 1990-91"
##  - attr(*, "time.stamp")= chr " 6 Apr 2012 08:40"
##  - attr(*, "formats")= chr  "%20s" "%9.0g" "%9.0g" "%9.0g" ...
##  - attr(*, "types")= int  20 251 254 254 254 254 254 252 254 254 ...
##  - attr(*, "val.labels")= chr  "" "region" "" "" ...
##  - attr(*, "var.labels")= chr  "State" "Geographical region" "1990 population" "Land area, square mil
##  - attr(*, "expansion.fields")=List of 4
##   ..$ : chr  "_dta" "_lang_c" "default"
##   ..$ : chr  "_dta" "_lang_list" "default"
##   ..$ : chr  "_dta" "__xi__Vars__To__Drop__" "_Iregion_2 _Iregion_3 _Iregion_4 _IregXperce_2 _IregXpe
##   ..$ : chr  "_dta" "__xi__Vars__Prefix__" "_I _I _I _I _I _I"
##  - attr(*, "version")= int 12
##  - attr(*, "label.table")=List of 1
##   ..$ region: Named int  1 2 3 4
##   .. ..- attr(*, "names")= chr  "West" "N. East" "South" "Midwest"
```

```r
#states.data
#look at last few labels
#tail(states.info, 8)
options(max.print=10000)
states.info
```

```
##      names                         var.labels
## 1    state                              State
## 2   region                Geographical region
## 3      pop                    1990 population
## 4     area            Land area, square miles
## 5  density             People per square mile
## 6    metro Metropolitan area population, %
## 7    waste     Per capita solid waste, tons
## 8   energy Per capita energy consumed, Btu
## 9    miles     Per capita miles/year, 1,000
## 10   toxic Per capita toxics released, lbs
## 11   green Per capita greenhouse gas, tons
## 12   house     House '91 environ. voting, %
## 13  senate    Senate '91 environ. voting, %
## 14    csat        Mean composite SAT score
## 15    vsat           Mean verbal SAT score
```

```
## 16    msat             Mean math SAT score
## 17 percent       % HS graduates taking SAT
## 18 expense Per pupil expenditures prim&sec
## 19  income Median household income, $1,000
## 20    high              % adults HS diploma
## 21 college          % adults college degree
```

```r
## Linear regression
## ===================

## Examine the data before fitting models
## ------------------------------------------

##   Start by examining the data to check for problems.

# summary of expense and csat columns, all rows
sts.ex.sat <- subset(states.data, select = c("expense", "csat"))
summary(sts.ex.sat)
```

```
##     expense          csat
##  Min.   :2960   Min.   : 832.0
##  1st Qu.:4352   1st Qu.: 888.0
##  Median :5000   Median : 926.0
##  Mean   :5236   Mean   : 944.1
##  3rd Qu.:5794   3rd Qu.: 997.0
##  Max.   :9259   Max.   :1093.0
```

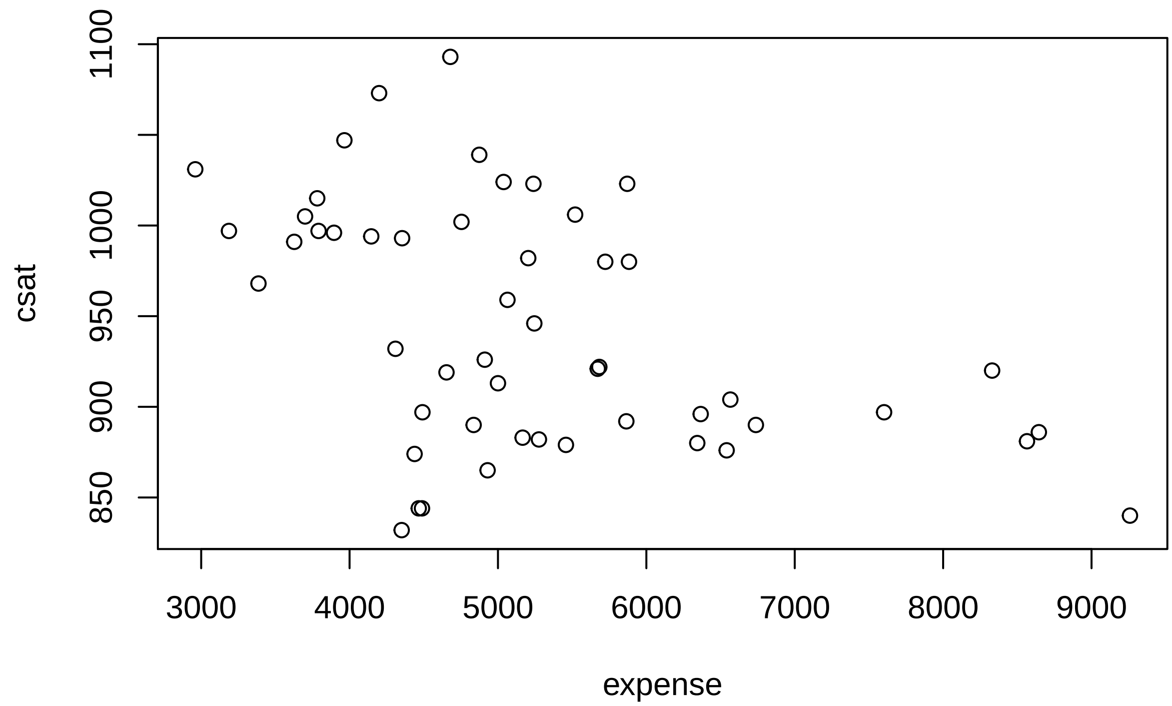
```r
# correlation between expense and csat
cor(sts.ex.sat)
```

```
##            expense       csat
## expense  1.0000000 -0.4662978
## csat    -0.4662978  1.0000000
```

```r
## Plot the data before fitting models
## ---------------------------------------

##   Plot the data to look for multivariate outliers, non-linear
##   relationships etc.

# scatter plot of expense vs csat
plot(sts.ex.sat)
```




```
## Linear regression example
## -----------------------------

##   - Linear regression models can be fit with the `lm()' function
##   - For example, we can use `lm' to predict SAT scores based on
##     per-pupal expenditures:

# Fit our regression model
sat.mod <- lm(csat ~ expense, # regression formula
              data=states.data) # data set
# Summarize and print the results
summary(sat.mod) # show regression coefficients table
```

```
##
## Call:
## lm(formula = csat ~ expense, data = states.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -131.811  -38.085    5.607   37.852  136.495
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.061e+03  3.270e+01   32.44  < 2e-16 ***
## expense     -2.228e-02  6.037e-03   -3.69 0.000563 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 59.81 on 49 degrees of freedom
## Multiple R-squared:  0.2174, Adjusted R-squared:  0.2015
## F-statistic: 13.61 on 1 and 49 DF,  p-value: 0.0005631
```

```r
sat.mod
```

```
##
## Call:
## lm(formula = csat ~ expense, data = states.data)
##
## Coefficients:
## (Intercept)      expense
##  1060.73244     -0.02228
```

```r
## Why is the association between expense and SAT scores /negative/?
## ---------------------------------------------------------------------

##   Many people find it surprising that the per-capita expenditure on
##   students is negatively related to SAT scores. The beauty of multiple
##   regression is that we can try to pull these apart. What would the
##   association between expense and SAT scores be if there were no
##   difference among the states in the percentage of students taking the
##   SAT?

summary(lm(csat ~ expense + percent, data = states.data))
```

```
##
## Call:
## lm(formula = csat ~ expense + percent, data = states.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -62.921 -24.318   1.741  15.502  75.623
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 989.807403  18.395770  53.806  < 2e-16 ***
## expense       0.008604   0.004204   2.046   0.0462 *
## percent      -2.537700   0.224912 -11.283 4.21e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.62 on 48 degrees of freedom
## Multiple R-squared:  0.7857, Adjusted R-squared:  0.7768
## F-statistic: 88.01 on 2 and 48 DF,  p-value: < 2.2e-16
```

```r
## The lm class and methods
## ----------------------------

##   OK, we fit our model. Now what?
##   - Examine the model object:

class(sat.mod)
```

```
## [1] "lm"
```

```r
names(sat.mod)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```r
methods(class = class(sat.mod))[1:9]
```

```
## [1] "add1.lm"                   "alias.lm"
## [3] "anova.lm"                  "case.names.lm"
## [5] "coerce,oldClass,S3-method" "confint.lm"
## [7] "cooks.distance.lm"         "deviance.lm"
## [9] "dfbeta.lm"
```

```r
##   - Use function methods to get more information about the fit

confint(sat.mod)
```

```
##                    2.5 %        97.5 %
## (Intercept) 995.01753164 1126.44735626
## expense      -0.03440768   -0.01014361
```

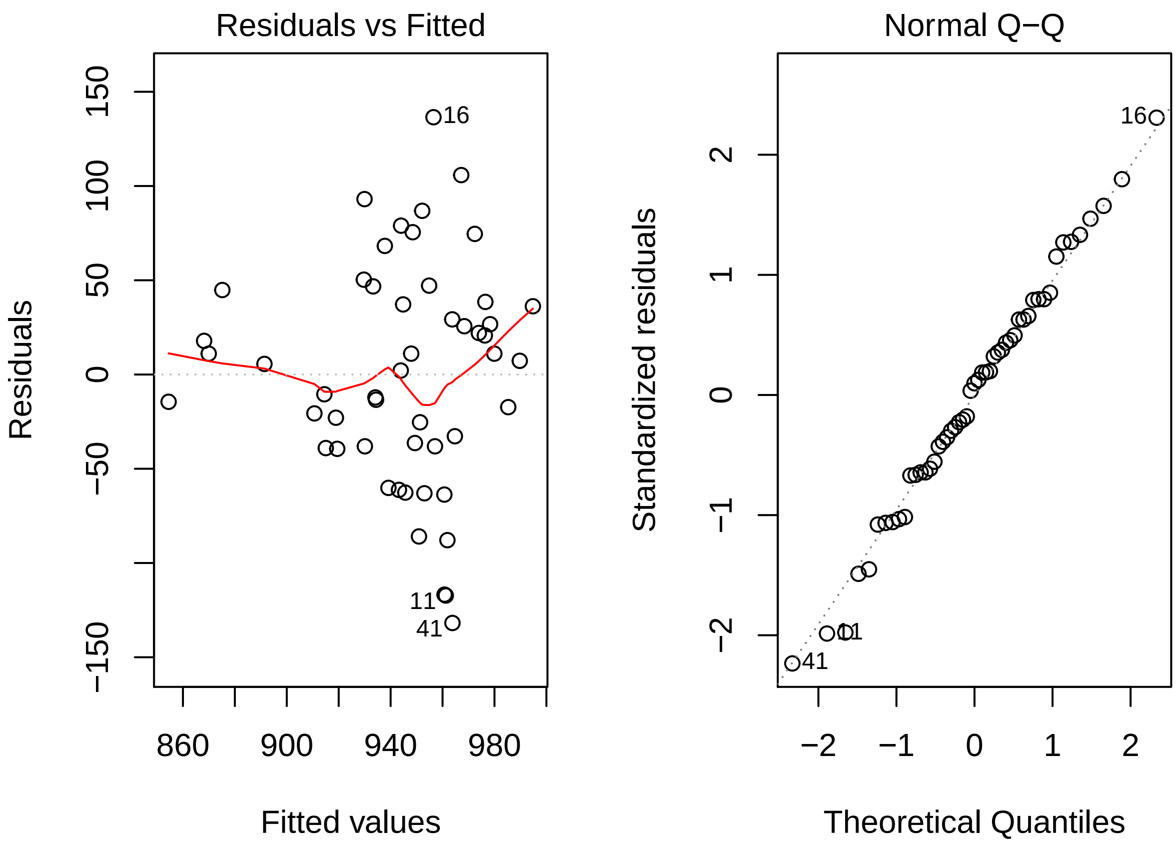
```r
# hist(residuals(sat.mod))

## Linear Regression Assumptions
## ---------------------------------

##   - Ordinary least squares regression relies on several assumptions,
##     including that the residuals are normally distributed and
##     homoscedastic, the errors are independent and the relationships are
##     linear.

##   - Investigate these assumptions visually by plotting your model:

par(mar = c(4, 4, 2, 2), mfrow = c(1, 2)) #optional
plot(sat.mod, which = c(1, 2)) # "which" argument optional
```

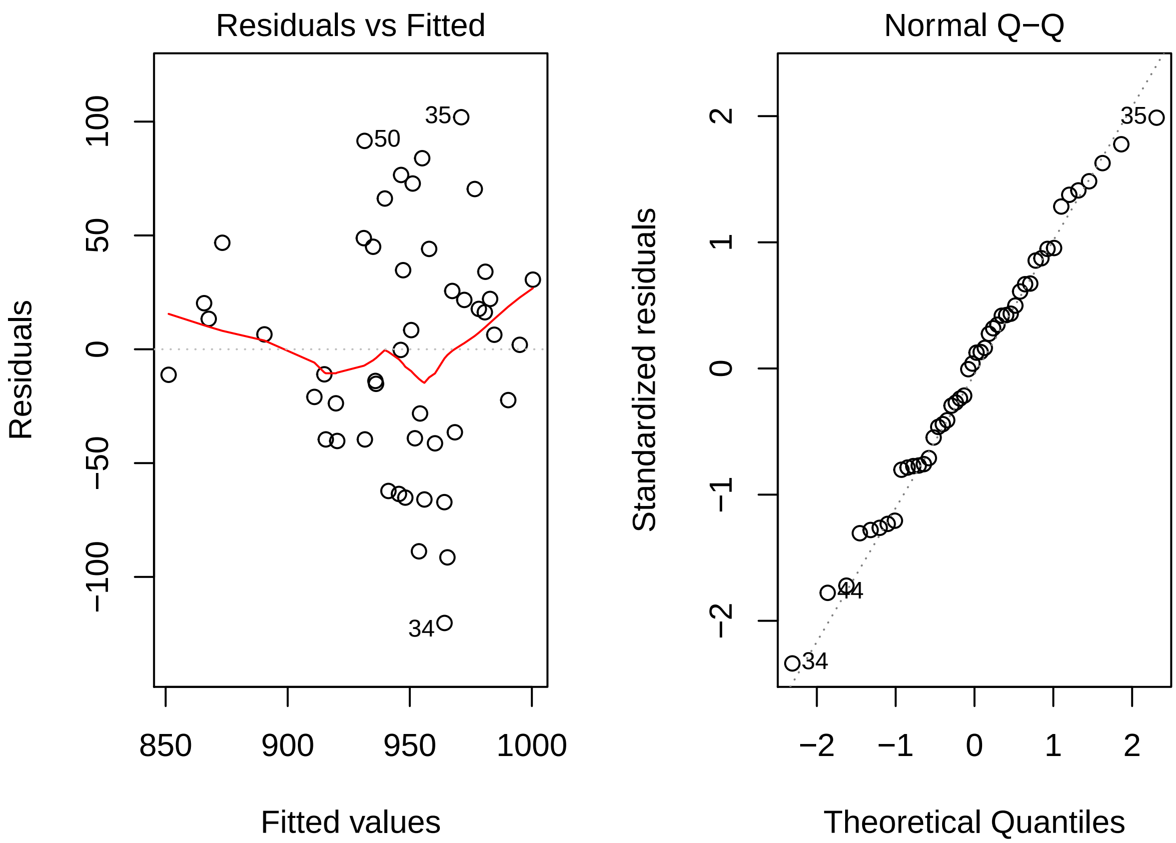



```
sat.mod
```

```
##
## Call:
## lm(formula = csat ~ expense, data = states.data)
##
## Coefficients:
## (Intercept)      expense
##  1060.73244     -0.02228
```

```
states2.data<-states.data[c(-11,-16,-41),]
sat2.mod <- lm(csat ~ expense, # regression formula
              data=states2.data) # data set

plot(sat2.mod, which = c(1, 2))
```




```r
summary(sat2.mod)
```

```
##
## Call:
## lm(formula = csat ~ expense, data = states2.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -120.226  -39.222    4.168   34.222  101.929
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.071e+03  2.905e+01  36.850  < 2e-16 ***
## expense     -2.369e-02  5.312e-03  -4.459 5.26e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.15 on 46 degrees of freedom
## Multiple R-squared:  0.3018, Adjusted R-squared:  0.2866
## F-statistic: 19.88 on 1 and 46 DF,  p-value: 5.258e-05
```

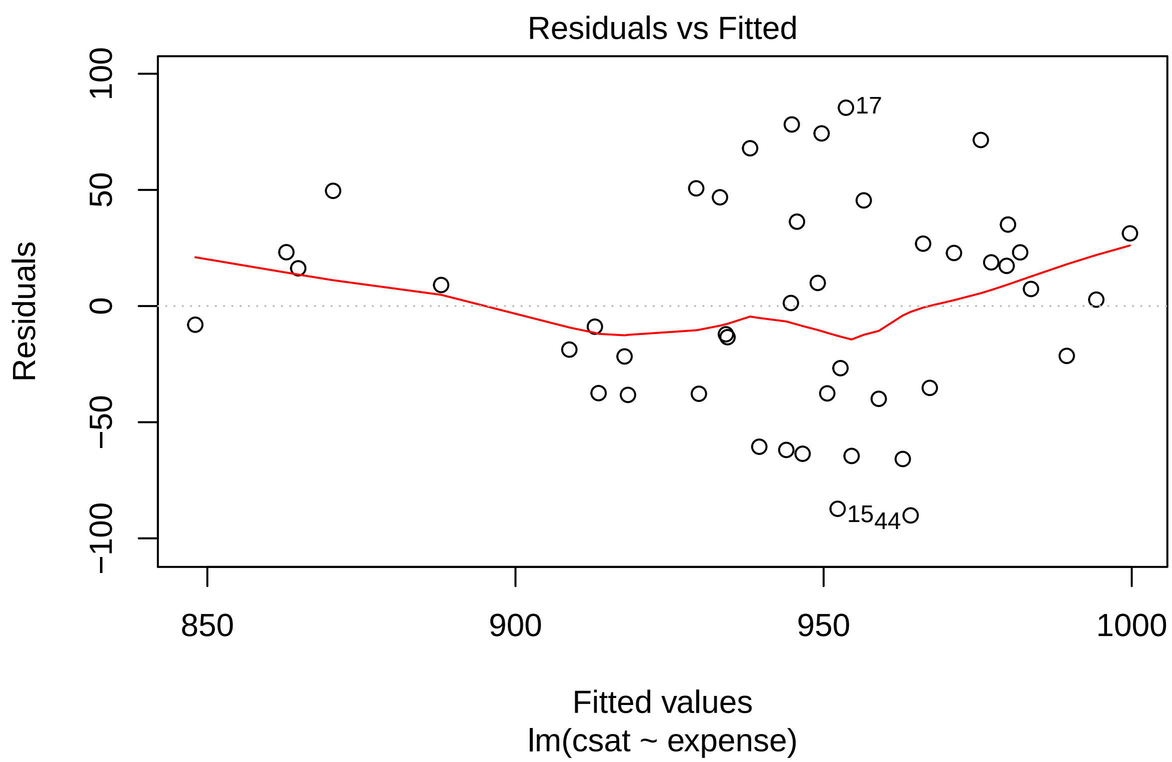
Removing the outliers in the first SAT linear regression model improves the r-square value from 0.21 to 0.30 which is an improvement but creates three more outliers in the new linear regression model, sat2.mod.

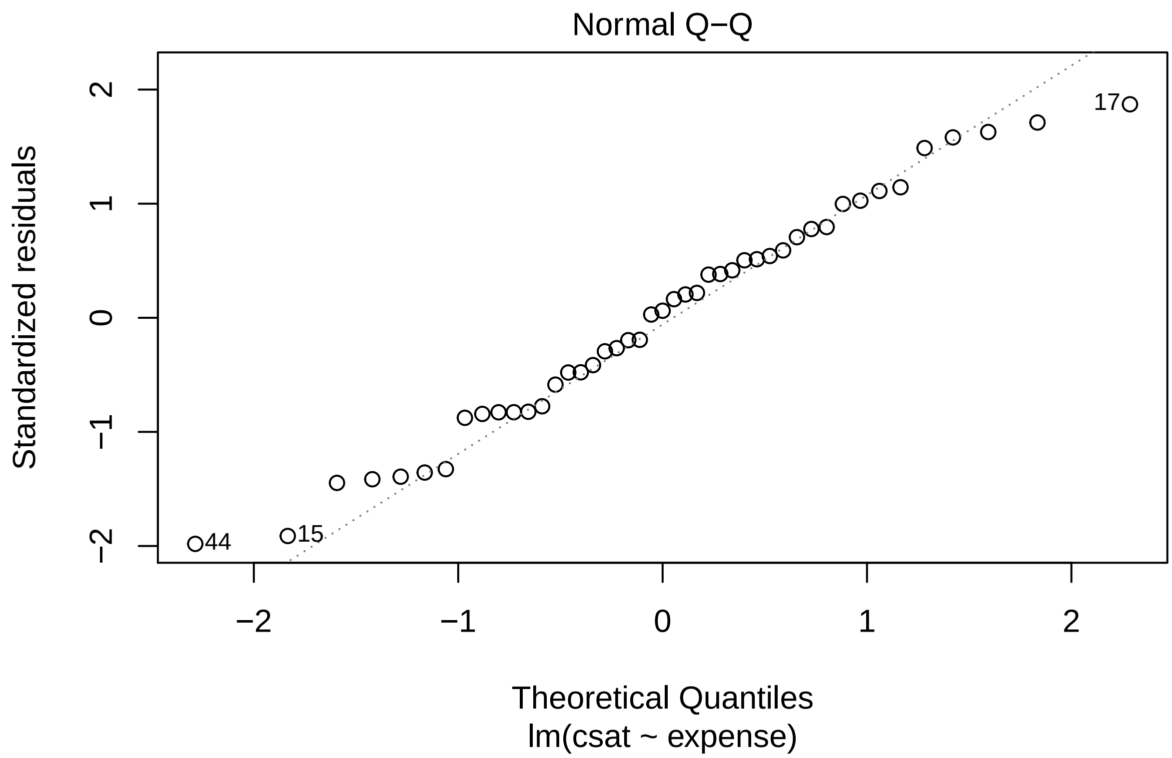
The red lines in the graphs look about the same. So let's try removing the nexzt 3 outliers.

As we see below, the r-square increases to 0.37 but the shape of the red line is still about the same.

```r
states3.data<-states.data[c(-11,-16,-34,-35,-41,-50),]
#states3.data
sat3.mod <- lm(csat ~ expense, # regression formula
               data=states3.data) # data set

plot(sat3.mod, which = c(1, 2))
```






```r
summary(sat3.mod)
```

```
##
## Call:
## lm(formula = csat ~ expense, data = states3.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.124 -37.485   2.752  31.286  85.374
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.071e+03  2.620e+01  40.882  < 2e-16 ***
## expense     -2.408e-02  4.759e-03  -5.059 8.33e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 46.19 on 43 degrees of freedom
## Multiple R-squared:  0.3731, Adjusted R-squared:  0.3586
## F-statistic:  25.6 on 1 and 43 DF,  p-value: 8.334e-06
```

```r
## Comparing models
## --------------------

##   Do congressional voting patterns predict SAT scores over and above
##   expense? Fit two models and compare them:
```

```r
# fit another model, adding house and senate as predictors
sat.voting.mod <-  lm(csat ~ expense + house + senate,
                      data = na.omit(states.data))
sat.mod <- update(sat.mod, data=na.omit(states.data))
# compare using the anova() function
anova(sat.mod, sat.voting.mod)
```

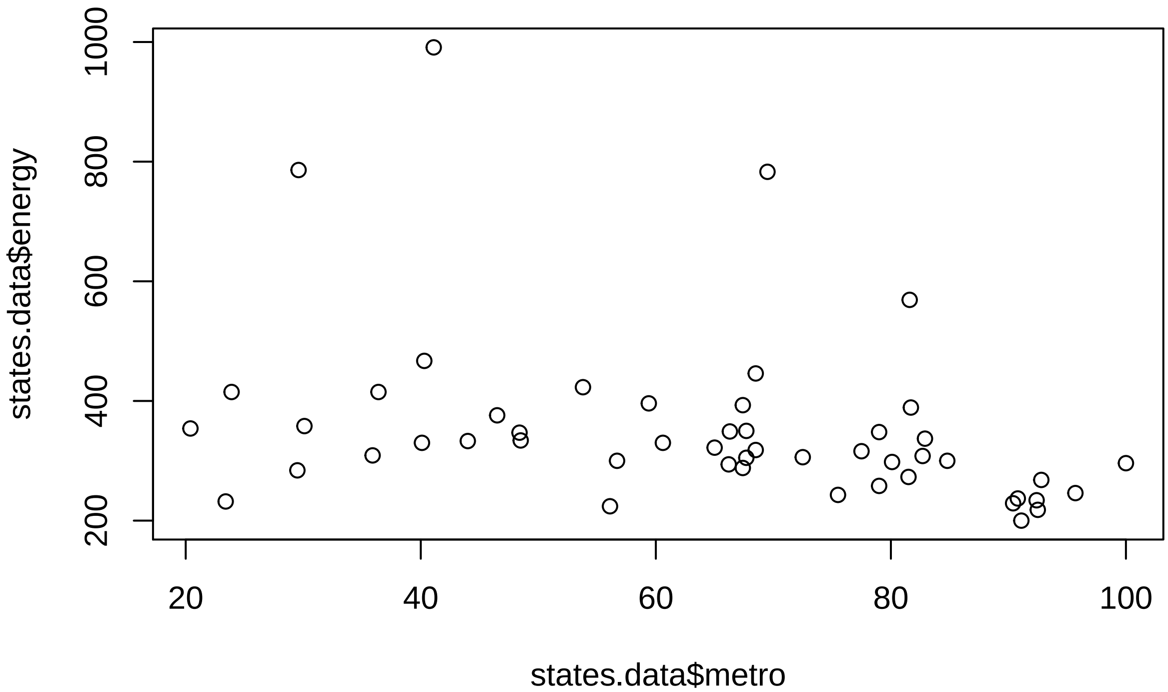
```
## Analysis of Variance Table
##
## Model 1: csat ~ expense
## Model 2: csat ~ expense + house + senate
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     46 169050
## 2     44 149284  2     19766 2.9128 0.06486 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coef(summary(sat.voting.mod))
```

```
##                 Estimate   Std. Error    t value     Pr(>|t|)
## (Intercept) 1082.93438041 38.633812740 28.0307405 1.067795e-29
## expense       -0.01870832  0.009691494 -1.9303852 6.001998e-02
## house         -1.44243754  0.600478382 -2.4021473 2.058666e-02
## senate         0.49817861  0.513561356  0.9700469 3.373256e-01
```

```r
## Exercise: least squares regression
## ────────────────────────────────────────

##   Use the /states.rds/ data set. Fit a model predicting energy consumed
##   per capita (energy) from the percentage of residents living in
##   metropolitan areas (metro). Be sure to
##   1. Examine/plot the data before fitting the model
plot(states.data$metro,states.data$energy)
```



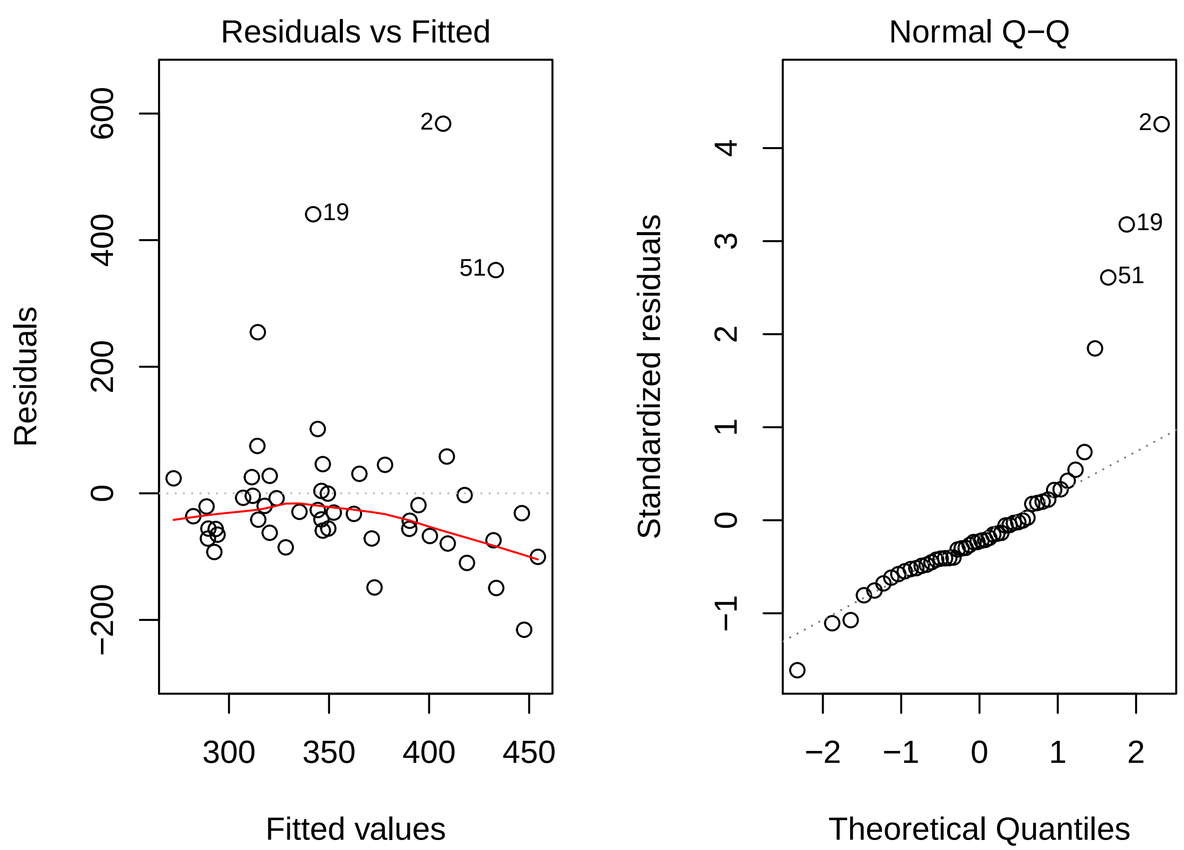

The plot does not look very linear. It looks like it is the top portion of a circle segment.

```
##   2. Print and interpret the model `summary'
energy.mod=lm(energy ~ metro, data=states.data)
summary(energy.mod)
```

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

The linear regression model using percentage of residents living in a metro area to predict energy usage is not a very good model with an r-square of only 0.11. The intercept is very important with a Pr(>|t|) of nearly zero and metro having somewhat significant with a Pr(>|t|) of 0.015. The coefficient for metro is negative making it suspect for playing a large role in predicting energy consumption.

```
##   3. `plot' the model to look for deviations from modeling assumptions
par(mar = c(4, 4, 2, 2), mfrow = c(1, 2)) #optional
plot(energy.mod, which = c(1, 2)) # "which" argument optional
```




Here again the red line resembles the top section of a circle segment, so it is not reflecting a linear relationship.

```
##   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?
energy2.mod=lm(energy ~ metro + waste + miles + green + expense + income + high + college
               + density + pop + area,
               data=states.data)
summary(energy2.mod)
```

```
##
## Call:
## lm(formula = energy ~ metro + waste + miles + green + expense +
##     income + high + college + density + pop + area, data = states.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -106.69  -39.04   -8.22   18.03  324.00
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.338e+02  2.708e+02   1.232    0.226
## metro        1.319e+00  9.685e-01   1.362    0.182
## waste       -4.016e+01  5.958e+01  -0.674    0.505
## miles        1.300e+01  1.570e+01   0.828    0.413
## green        4.750e+00  8.584e-01   5.533 2.93e-06 ***
## expense      1.843e-03  1.598e-02   0.115    0.909
## income      -3.106e+00  5.263e+00  -0.590    0.559
## high        -2.170e+00  3.941e+00  -0.551    0.585
## college     -1.826e+00  5.991e+00  -0.305    0.762
## density      1.408e-02  9.875e-02   0.143    0.887
## pop         -1.435e-06  3.701e-06  -0.388    0.701
## area         4.353e-04  3.934e-04   1.107    0.276
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.24 on 36 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.683,  Adjusted R-squared:  0.5862
## F-statistic: 7.052 on 11 and 36 DF,  p-value: 3.361e-06
```

Adding almost all of the variables raises the r-square to significantly to 0.68. I was expecting waste to play a larger role in energy consumption thinking that more waste implies more energy. The linear regression model shows per-capita green house gas emissions as the most important factor in predicting energy consumption. After thinking about the role of green house emissions, it makes sense that it is a good predictor for energy consumption.

```r
energy2.mod=lm(energy ~ metro + green + area, data=states.data)
summary(energy2.mod)
```

```
##
## Call:
## lm(formula = energy ~ metro + green + area, data = states.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -90.34 -38.23 -11.64  29.82 373.62
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.843e+02  4.744e+01   3.884 0.000341 ***
## metro       4.909e-03  5.373e-01   0.009 0.992752
## green       5.233e+00  7.301e-01   7.168  6.5e-09 ***
## area        4.469e-04  2.348e-04   1.903 0.063589 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 74.04 on 44 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.6248, Adjusted R-squared:  0.5992
## F-statistic: 24.42 on 3 and 44 DF,  p-value: 1.858e-09
```

Using just metro percent, green house emmissions and area variables only reduces the r-square by 0.06 to 0.62 which simplifies the model. Metro percent is not a good contributor and should be removed.

```r
## Interactions and factors
## ==========================

## Modeling interactions
## -------------------------

##   Interactions allow us assess the extent to which the association
##   between one predictor and the outcome depends on a second predictor.
##   For example: Does the association between expense and SAT scores
##   depend on the median income in the state?

  #Add the interaction to the model
sat.expense.by.percent <- lm(csat ~ expense*income,
                             data=states.data)
#Show the results
  coef(summary(sat.expense.by.percent)) # show regression coefficients table
```

```
##                     Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept)     1.380364e+03 1.720863e+02  8.021351 2.367069e-10
## expense        -6.384067e-02 3.270087e-02 -1.952262 5.687837e-02
## income         -1.049785e+01 4.991463e+00 -2.103161 4.083253e-02
## expense:income  1.384647e-03 8.635529e-04  1.603431 1.155395e-01
```

```r
## Regression with categorical predictors
## ------------------------------------------

##   Let's try to predict SAT scores from region, a categorical variable.
##   Note that you must make sure R does not think your categorical
##   variable is numeric.

# make sure R knows region is categorical
str(states.data$region)
```

```
##  Factor w/ 4 levels "West","N. East",..: 3 1 1 3 1 1 2 3 NA 3 ...
```

```r
states.data$region <- factor(states.data$region)
#Add region to the model
sat.region <- lm(csat ~ region,
                 data=states.data)
#Show the results
coef(summary(sat.region)) # show regression coefficients table
```

```
##                 Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)    946.30769   14.79582 63.9577807 1.352577e-46
## regionN. East -56.75214   23.13285 -2.4533141 1.800383e-02
## regionSouth   -16.30769   19.91948 -0.8186806 4.171898e-01
## regionMidwest  63.77564   21.35592  2.9863209 4.514152e-03
```

```r
anova(sat.region) # show ANOVA table
```

```
## Analysis of Variance Table
##
## Response: csat
##           Df Sum Sq Mean Sq F value    Pr(>F)
## region     3  82049 27349.8  9.6102 4.859e-05 ***
## Residuals 46 130912  2845.9
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##   Again, *make sure to tell R which variables are categorical by
##   converting them to factors!*

## Setting factor reference groups and contrasts
## -----------------------------------------------

##   In the previous example we use the default contrasts for region. The
##   default in R is treatment contrasts, with the first level as the
##   reference. We can change the reference group or use another coding
##   scheme using the `C' function.

# print default contrasts
contrasts(states.data$region)
```

```
##         N. East South Midwest
## West          0     0       0
## N. East       1     0       0
## South         0     1       0
## Midwest       0     0       1
```

```
# change the reference group
coef(summary(lm(csat ~ C(region, base=4),
                data=states.data)))
```

```
##                         Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)           1010.08333   15.39998 65.589930 4.296307e-47
## C(region, base = 4)1   -63.77564   21.35592 -2.986321 4.514152e-03
## C(region, base = 4)2  -120.52778   23.52385 -5.123641 5.798399e-06
## C(region, base = 4)3   -80.08333   20.37225 -3.931000 2.826007e-04
```

```
# change the coding scheme
coef(summary(lm(csat ~ C(region, contr.helmert),
                data=states.data)))
```

```
##                              Estimate Std. Error     t value     Pr(>|t|)
## (Intercept)                943.986645   7.706155 122.4977451 1.689670e-59
## C(region, contr.helmert)1 -28.376068  11.566423  -2.4533141 1.800383e-02
## C(region, contr.helmert)2   4.022792   5.884552   0.6836191 4.976450e-01
## C(region, contr.helmert)3  22.032229   4.446777   4.9546509 1.023364e-05
```

```
##   See also `?contrasts', `?contr.treatment', and `?relevel'.

## Exercise: interactions and factors
## ----------------------------------------

##   Use the states data set.

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

energy3.mod=lm(energy ~ pop*green + area, data=states.data)
summary(energy3.mod)
```

```
##
```

```
## Call:
## lm(formula = energy ~ pop * green + area, data = states.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -105.16  -33.34   -7.17   23.20  313.84
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.041e+02  2.239e+01   9.115 1.33e-11 ***
## pop         -1.705e-05  4.711e-06  -3.619 0.000775 ***
## green        4.296e+00  6.556e-01   6.552 5.74e-08 ***
## area         2.790e-04  2.305e-04   1.210 0.232753
## pop:green    9.442e-07  2.547e-07   3.707 0.000595 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65.01 on 43 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.7173, Adjusted R-squared:  0.691
## F-statistic: 27.27 on 4 and 43 DF,  p-value: 2.628e-11
```

Having population interact with green house emissions data raises the r-square back to 0.71, an improvement of 9%.

```
##   2. Try adding region to the model. Are there significant differences
##      across the four regions?

energy4.mod=lm(energy ~ region + area + pop*green, data=states.data)
summary(energy4.mod)
```

```
##
## Call:
## lm(formula = energy ~ region + area + pop * green, data = states.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -98.04 -31.65 -10.76  20.12 288.36
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.083e+02  4.176e+01   4.987 1.24e-05 ***
## regionN. East -1.437e+01  4.059e+01  -0.354  0.72513
## regionSouth    2.337e+01  3.357e+01   0.696  0.49041
## regionMidwest -3.162e+01  3.353e+01  -0.943  0.35130
## area           2.566e-04  3.188e-04   0.805  0.42552
## pop           -1.674e-05  5.123e-06  -3.267  0.00224 **
## green          4.306e+00  6.545e-01   6.578 7.29e-08 ***
## pop:green      9.276e-07  2.952e-07   3.142  0.00316 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63.17 on 40 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.7517, Adjusted R-squared:  0.7083
```

```
## F-statistic:  17.3 on 7 and 40 DF,  p-value: 2.68e-10
```

Adding categorical region data as an interaction term raises the r-square to 0.75, a further improvement of 4%. This is the best model so far. Adding a factor and interaction to the model has really helped.

```r
# print default contrasts
contrasts(states.data$region)
```

```
##         N. East South Midwest
## West          0     0       0
## N. East       1     0       0
## South         0     1       0
## Midwest       0     0       1
```

```r
# change the reference group
coef(summary(lm(energy ~ C(region, base=4),
                data=states.data)))
```

```
##                        Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)           344.00000   40.83392  8.4243691 7.058813e-11
## C(region, base = 4)1   61.61538   56.62646  1.0881024 2.822182e-01
## C(region, base = 4)2  -94.88889   62.37484 -1.5212686 1.350374e-01
## C(region, base = 4)3   36.12500   54.01820  0.6687561 5.069935e-01
```

```r
# change the coding scheme
coef(summary(lm(energy ~ C(region, contr.helmert),
                data=states.data)))
```

```
##                               Estimate Std. Error     t value     Pr(>|t|)
## (Intercept)                 344.7128739   20.43331 16.87014156 2.504367e-21
## C(region, contr.helmert)1  -78.2521368   30.66903 -2.55150316 1.411486e-02
## C(region, contr.helmert)2   17.5872507   15.60323  1.12715468 2.655225e-01
## C(region, contr.helmert)3   -0.2376246   11.79088 -0.02015325 9.840083e-01
```

I need to study the 'contrasts' and 'coef' more to comment on them.