Q2 - Regression

Weizhuo Wang

2/23/2020

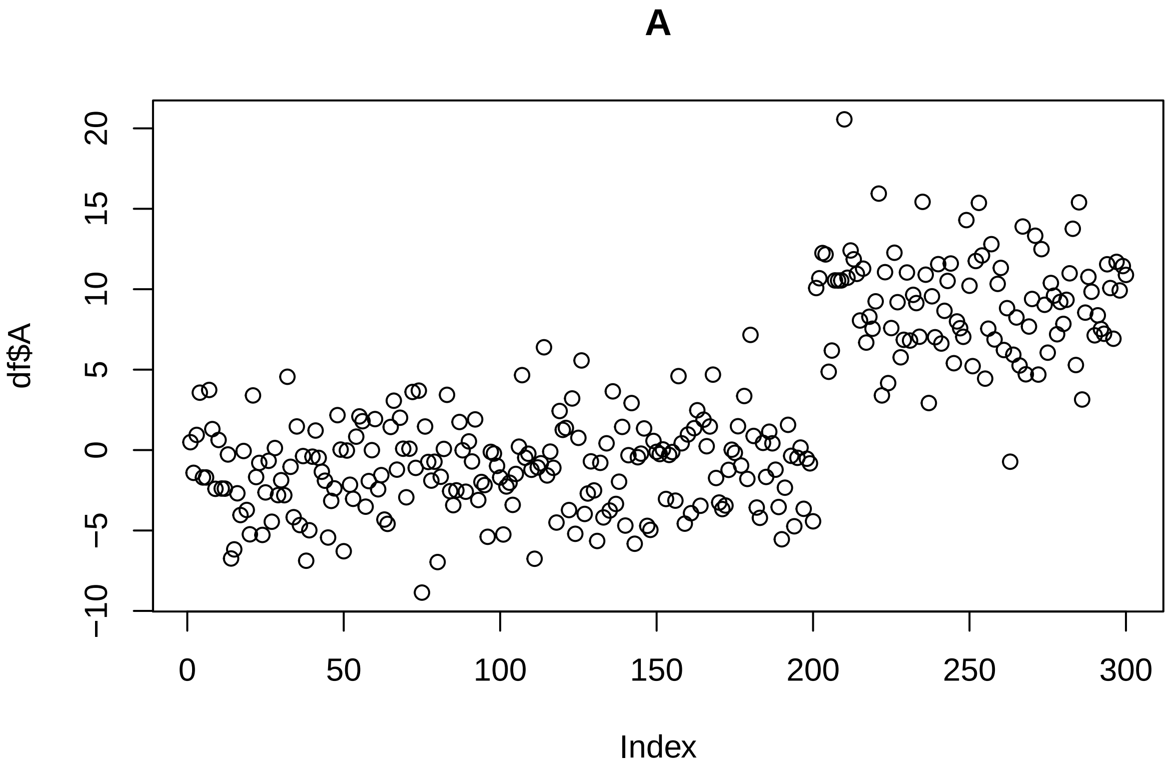
```r
df = readxl::read_xlsx("Adops & Data Scientist Sample Data.xlsx",
                       sheet = 2, col_names = c("A", "B", "C"))
```

Check missing values and variable distribution.

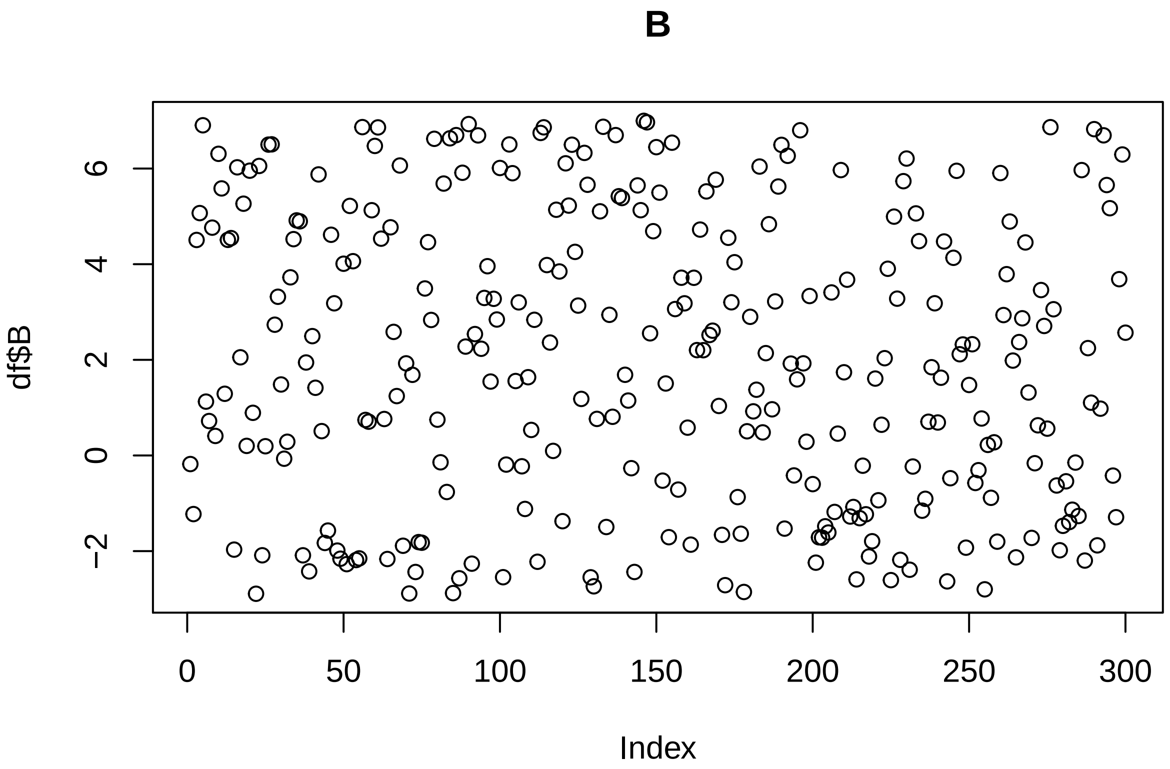
```r
sum(is.na(df))
```

```
## [1] 0
```

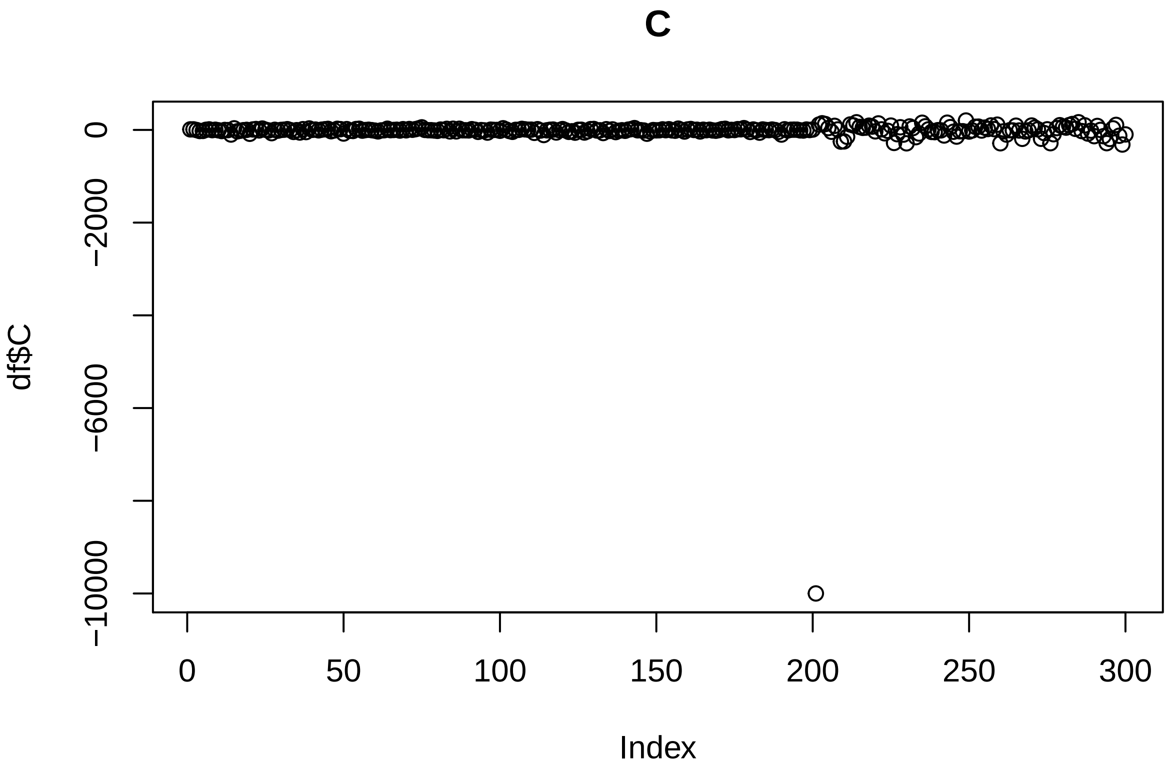
```r
plot(df$A, main = "A")
```



```r
plot(df$B, main = "B")
```



```
plot(df$C, main = "C")
```



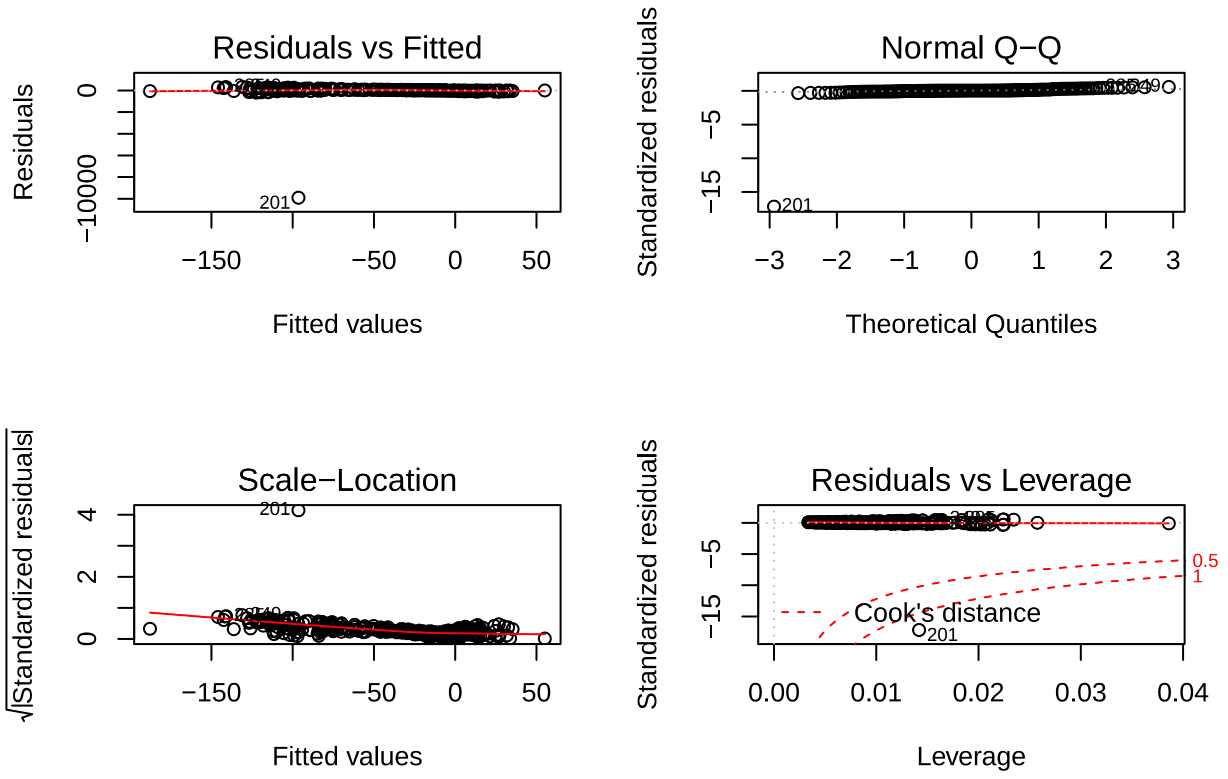
By checking on distributions, there is a potential outlier.

First fit a simple linear regression as initial model.

```r
mod0 = lm(C ~ A + B, data = df)
summary(mod0)
```

```
##
## Call:
## lm(formula = C ~ A + B, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9902.6    -4.9    23.0    53.8   333.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -19.376     44.422  -0.436    0.663
## A             -8.044      6.017  -1.337    0.182
## B             -1.767     11.426  -0.155    0.877
##
## Residual standard error: 581.4 on 297 degrees of freedom
## Multiple R-squared:  0.006018,   Adjusted R-squared:  -0.0006753
## F-statistic: 0.8991 on 2 and 297 DF,  p-value: 0.408
```

```r
par(mfrow = c(2,2))
plot(mod0)
```



Observation #201 is obviously an outlier, so remove it and run regression again.

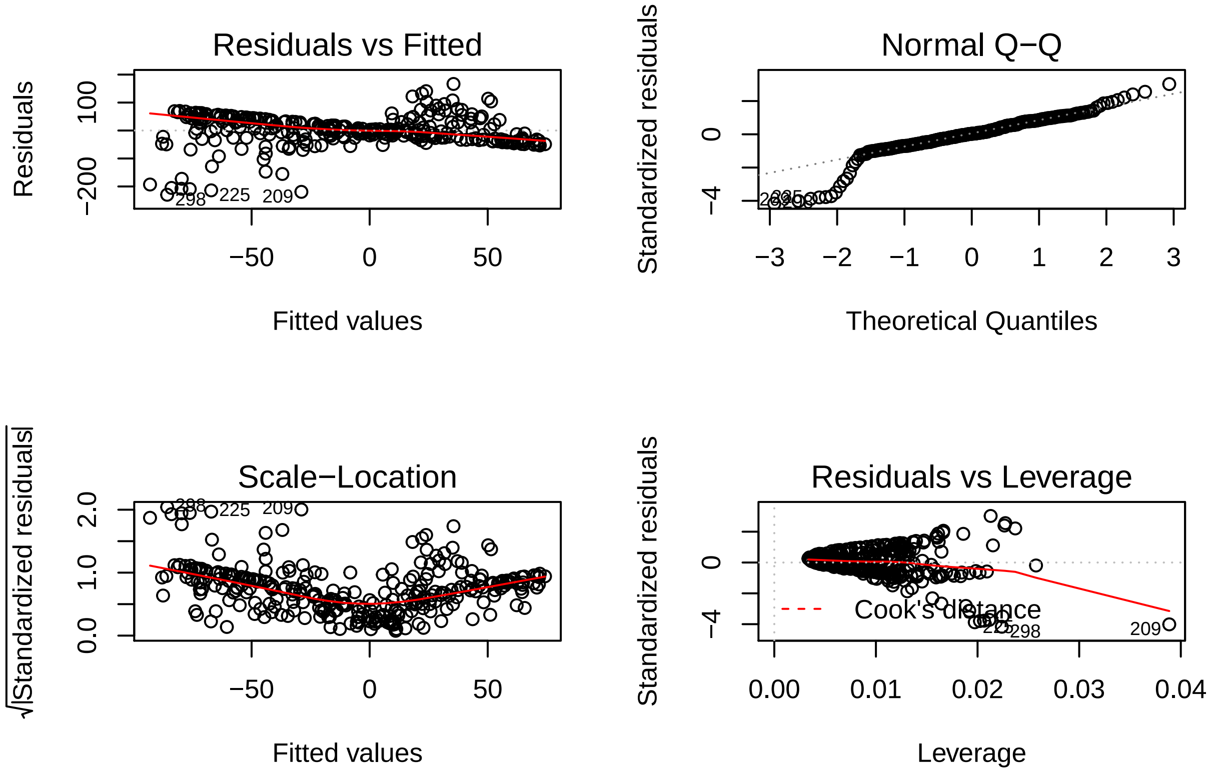
```r
# df2[201,]  # see what 201 looks like
df.modified = df[-201,]
mod1 = lm(C ~ A + B, data = df.modified)
summary(mod1)
```

```
##
## Call:
## lm(formula = C ~ A + B, data = df.modified)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -229.634  -26.400    1.922   33.020  166.514
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25.7393     4.2654   6.034 4.76e-09 ***
## A            -1.3703     0.5779  -2.371   0.0184 *
## B           -15.2259     1.0978 -13.869  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.73 on 296 degrees of freedom
## Multiple R-squared:  0.394,  Adjusted R-squared:  0.3899
## F-statistic: 96.21 on 2 and 296 DF,  p-value: < 2.2e-16
```

```
ppcor::pcor(df.modified[,1:2], method = "pearson")$estimate # no correlation between A and B
```

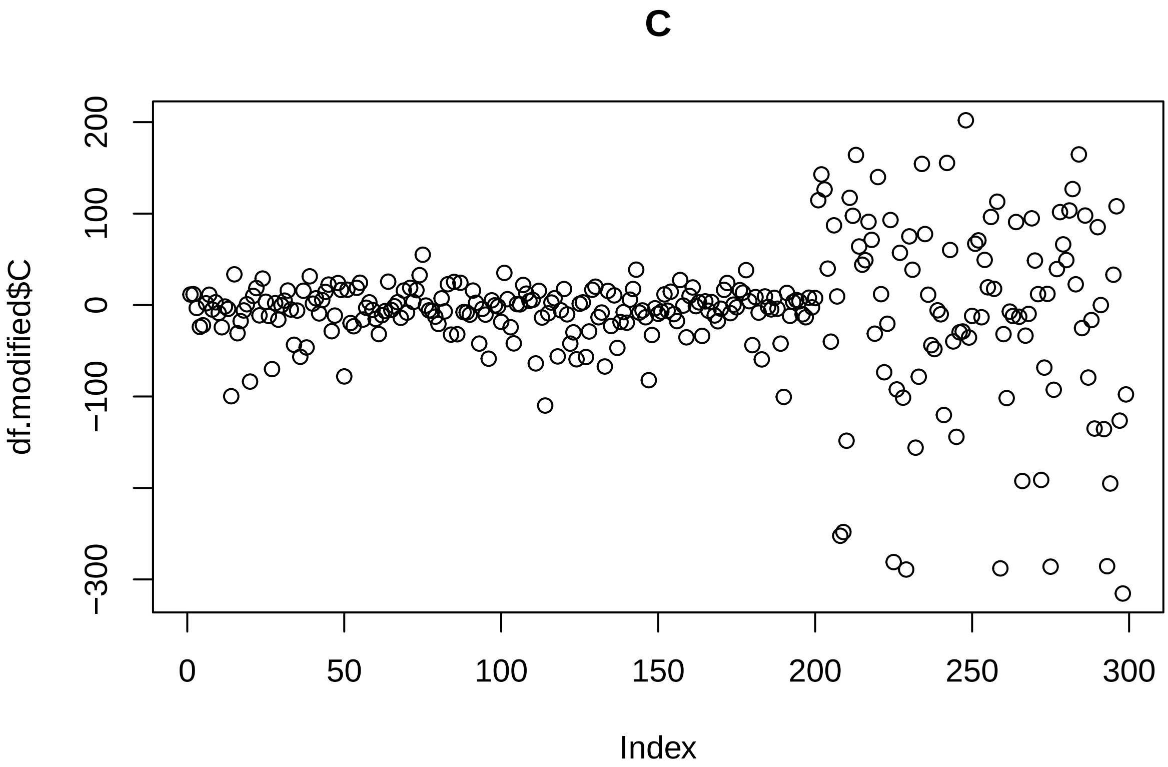
```
##           A         B
## A  1.000000 -0.186992
## B -0.186992  1.000000
```

```
par(mfrow = c(2,2))
plot(mod1)
```

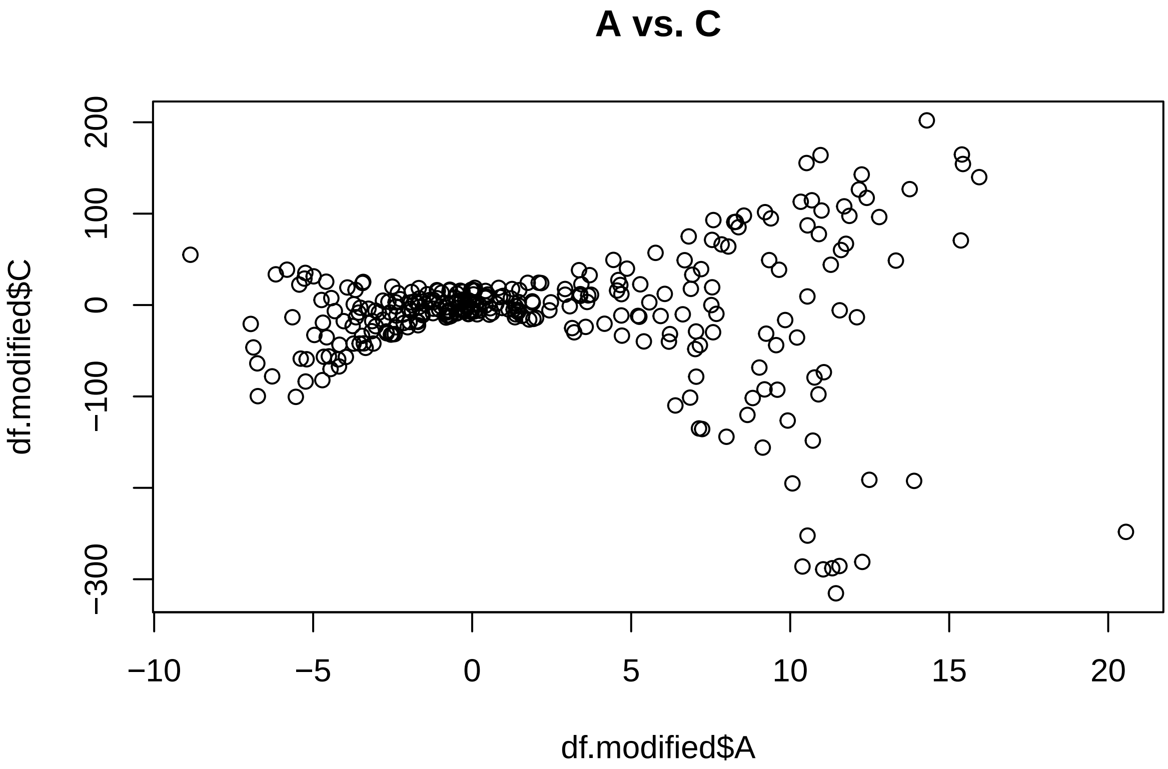


Regression model looks better, and there is no correlation between independent variables, can then start refining model.

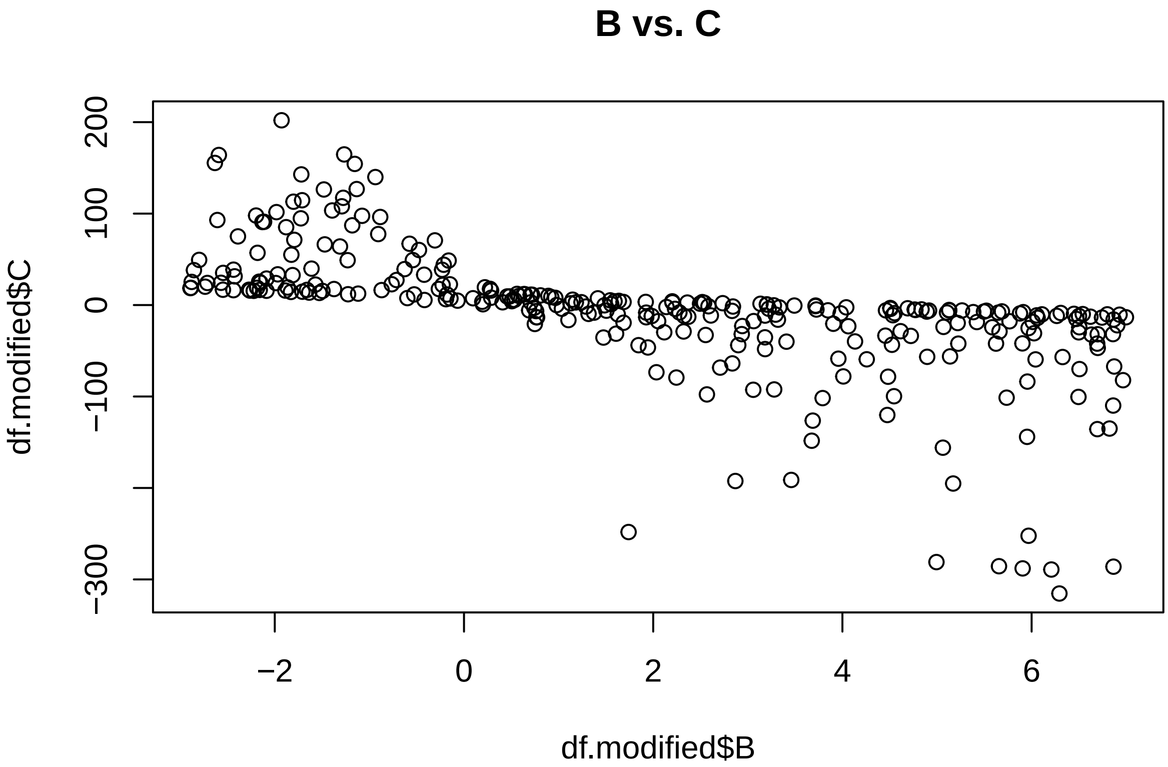
```
plot(df.modified$C, main = "C")
```



```
plot(df.modified$A, df.modified$C, main = "A vs. C")
```



```r
plot(df.modified$B, df.modified$C, main = "B vs. C")
```



Also notice a pattern change on variable C after 200 observations. To solve this, I create time variable (called `t`) to indicate the time of a point being recorded.

For both independent variables, relation to C changes at point 0, therefor a dummy variable representing positive or negative may be useful (`IA` and `IB` respectively, with 1 means negative value of that variable).

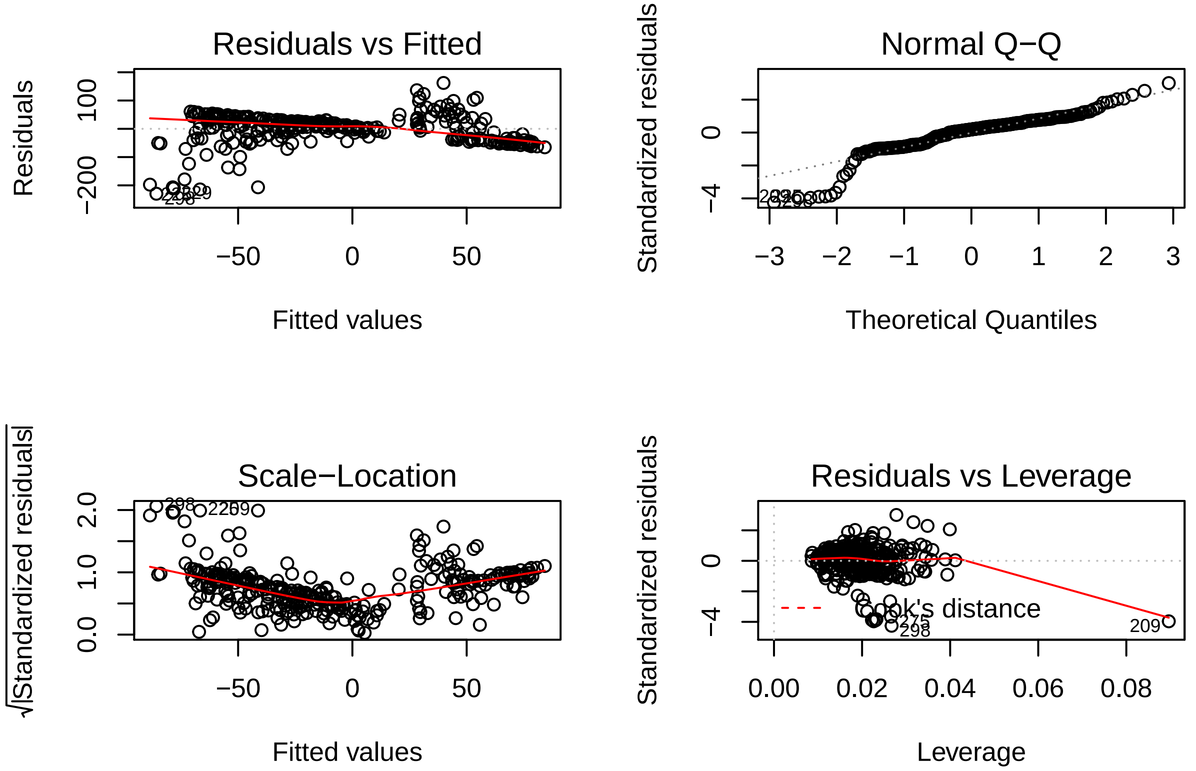
```r
df.modified$t = c(1:nrow(df.modified))
df.modified$IA = as.factor(ifelse(df.modified$A<0, "1", "0"))
df.modified$IB = as.factor(ifelse(df.modified$B<0, "1", "0"))
```

Fit model with all variables.

```r
mod2 = lm(C ~ ., df.modified) # put everything into expaintory variable
summary(mod2)
```

```
##
## Call:
## lm(formula = C ~ ., data = df.modified)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -229.64  -32.76   10.03   30.84  162.15
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.62668   11.65710   1.941  0.05321 .
## A            -1.33484    1.07758  -1.239  0.21643
## B           -10.78126    1.75678  -6.137 2.73e-09 ***
## t            -0.08477    0.05189  -1.634  0.10339
## IA1          -9.92235   10.20539  -0.972  0.33172
## IB1          36.56175   11.44278   3.195  0.00155 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.7 on 293 degrees of freedom
## Multiple R-squared:  0.4221, Adjusted R-squared:  0.4123
## F-statistic: 42.81 on 5 and 293 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(mod2)
```



Model performance is not very satisfactory. Here are some thoughts:

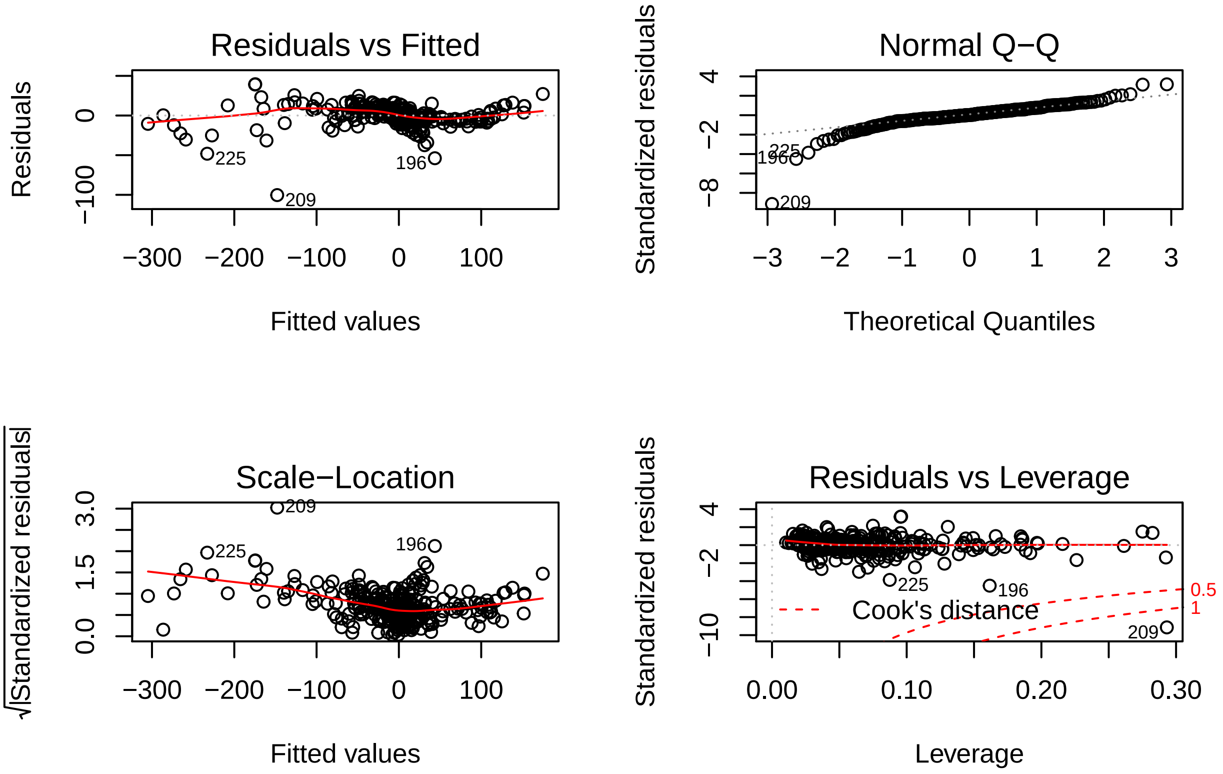
- the negative value of A or B can influence the other, add interaction of `A*B` or `A*IB` may help explain more variance. However, adding both interaction forms may cause overfitting
- time variable `t` influences all other independent variables, so we can add interaction of `t` and all others

With the above ideas, add all variables and their interactions into a full model, then use model selection method (BIC) to choose important features.

```
# this is the full model
# use BIC method to reduce the size of full model
mod3 = lm(C ~ A*B*t*IA*IB, df.modified)
mod3.re = MASS::stepAIC(mod3, direction = "backward", k = log(nrow(df.modified)), trace = 0)
summary(mod3.re)
```

```
##
## Call:
## lm(formula = C ~ A + B + t + IA + IB + A:B + A:t + B:t + A:IA +
##     B:IA + t:IA + A:IB + t:IB + IA:IB + A:B:t + A:B:IA + A:t:IA +
##     B:t:IA + A:IA:IB + A:B:t:IA, data = df.modified)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -100.311   -4.622    0.247    7.058   39.463
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.095763   7.091597   3.116 0.002027 **
## A            -3.054444   1.364114  -2.239 0.025938 *
## B            -5.770460   1.625493  -3.550 0.000452 ***
## t             0.041722   0.038146   1.094 0.275017
## IA1         -13.907756   9.536919  -1.458 0.145885
## IB1         -11.016867   8.583643  -1.283 0.200396
## A:B          -1.708746   0.408169  -4.186 3.81e-05 ***
## A:t           0.005868   0.005617   1.045 0.297157
## B:t           0.042877   0.009770   4.389 1.62e-05 ***
## A:IA1         6.976792   2.520979   2.767 0.006028 **
## B:IA1         5.555398   2.210398   2.513 0.012526 *
## t:IA1        -0.002237   0.057531  -0.039 0.969013
## A:IB1         6.702359   0.989971   6.770 7.62e-11 ***
## t:IB1        -0.143146   0.044021  -3.252 0.001288 **
## IA1:IB1      19.098221  10.146479   1.882 0.060846 .
## A:B:t        -0.011446   0.001657  -6.908 3.34e-11 ***
## A:B:IA1       3.737579   0.662058   5.645 4.06e-08 ***
## A:t:IA1      -0.015244   0.015694  -0.971 0.332245
## B:t:IA1      -0.058412   0.014686  -3.977 8.89e-05 ***
## A:IA1:IB1    -9.583259   2.603564  -3.681 0.000279 ***
## A:B:t:IA1     0.010796   0.003995   2.703 0.007299 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.06 on 278 degrees of freedom
## Multiple R-squared:  0.9688, Adjusted R-squared:  0.9665
## F-statistic: 431.1 on 20 and 278 DF,  p-value: < 2.2e-16
```

```r
par(mfrow = c(2,2))
plot(mod3.re)
```



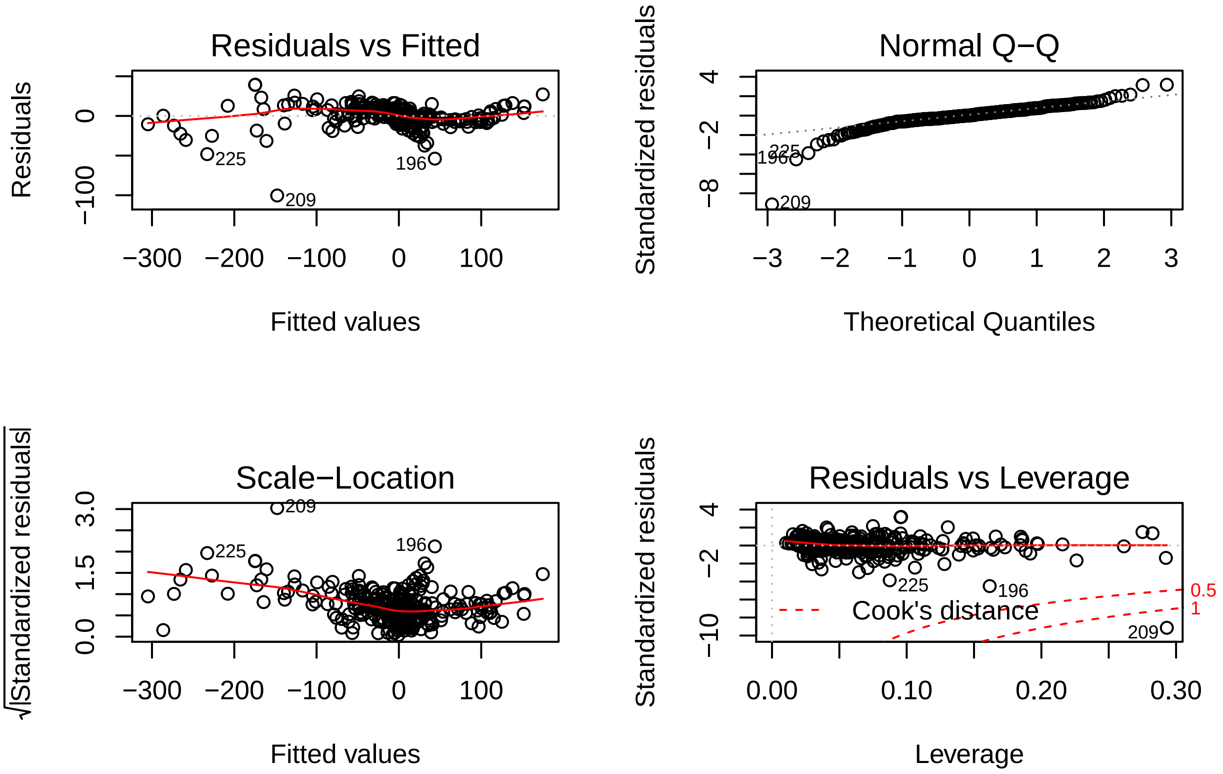
The selected model performance is excellent, it explains about 97% variation in variable C. However, there may exist overfitting.

Even the reduced model still has tons of variables, intuitively, some variables seem not necessary. For example the interaction term between time and `IA`, time variable will use anyway no matter A takes positive or negative value. Therefore, I would delete some interaction terms regarding `IA` and `IB` and run model selection again.

```
# this is the intuition model, again use BIC
mod4 = lm(C ~ A*B*t + IA:A + IA:B + IB:A + IB:B + A:B:IA + A:B:IB, data = df.modified)
mod4.re = MASS::stepAIC(mod4, direction = "backward", k = log(nrow(df.modified)), trace = 0)
summary(mod4.re)
```

```
##
## Call:
## lm(formula = C ~ A + B + A:B + A:IA + B:IA + A:IB + A:B:IA, data = df.modified)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -123.325   -4.893   -0.428    6.248   45.476
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.25691    1.89700   5.407 1.34e-07 ***
## A            1.03729    0.43692   2.374   0.0182 *
## B            4.13993    0.64047   6.464 4.29e-10 ***
## A:B         -4.57601    0.12659 -36.148  < 2e-16 ***
## A:IA1       -0.09123    0.81724  -0.112   0.9112
## B:IA1       -5.29458    0.79602  -6.651 1.44e-10 ***
## A:IB1        1.60503    0.52616   3.050   0.0025 **
## A:B:IA1      7.06171    0.23759  29.723  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.08 on 291 degrees of freedom
## Multiple R-squared:  0.9564, Adjusted R-squared:  0.9553
## F-statistic: 911.4 on 7 and 291 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(mod3.re)
```



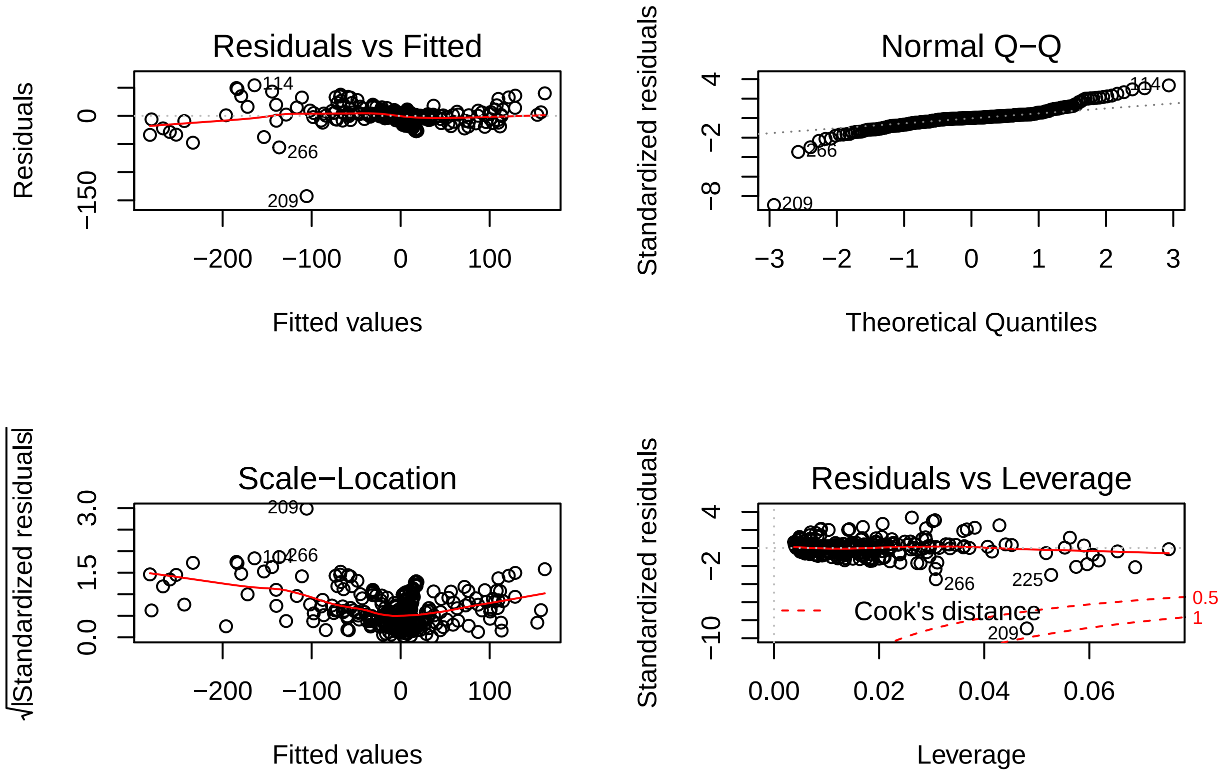
So this time, the model fits the data perfectly without being too complected, diagnosis plots are reasonableness acceptable.

Also notice that time variable is not used in the final model. Again, `IB` seems not useful, so try to delete more interaction terms to see result.

```
mod5 = lm(C ~ A*B + A:B:IA, df.modified)
summary(mod5)
```

```
##
## Call:
## lm(formula = C ~ A * B + A:B:IA, data = df.modified)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -142.531   -5.982    0.245    5.187   54.344
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.36027    1.29257   8.015 2.61e-14 ***
## A              2.09032    0.18821  11.106  < 2e-16 ***
## B              1.47992    0.48489   3.052  0.00248 **
## A:B           -4.51779    0.08095 -55.810  < 2e-16 ***
## A:B:IA1        7.47860    0.21249  35.195  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.38 on 294 degrees of freedom
## Multiple R-squared:  0.948,  Adjusted R-squared:  0.9473
## F-statistic:  1339 on 4 and 294 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(mod5)
```



This model is more ideal in terms of performance and interpretability with all diagnosis plots generally valid. Therefore, I choose it as my final regression mode. The model is written as

$$C = \beta_0 + \beta_1 A + \beta_2 B + \beta_3 A * B + \beta_4 A * B * I_A + \epsilon$$