**My computer not working so I have to work on desktop from university so that why The generating of the random data doesnt exist here but however the data I took it from the software it is the the output of my generating before my laptop crashes.**

**I used random data see more precisely the varaibles that impact the value of y ,Note:the generating of the data was in my laptop.**

```
data <- read.csv("NewData.csv", header = TRUE, stringsAsFactors = FALSE, sep = ",")


data$Date <- as.POSIXct(data$Date, format = "%Y-%m-%d-%H:%M:%S")

# View the first few rows
head(data)
```

```
##                  Date        x1        x2         x3        x4         x5
## 1 2024-02-16 09:43:14 0.2875775 0.9860543 0.47068183 0.8585914 0.63918581
## 2 2024-02-16 09:43:15 0.7883051 0.1370675 0.36584547 0.8873848 0.12482240
## 3 2024-02-16 09:43:16 0.4089769 0.9053096 0.12127205 0.4890915 0.25526578
## 4 2024-02-16 09:43:18 0.8830174 0.5763018 0.04699368 0.7180918 0.82057469
## 5 2024-02-16 09:43:19 0.9404673 0.3954489 0.26279630 0.4867056 0.80378039
## 6 2024-02-16 09:43:20 0.0455565 0.4498025 0.96864117 0.9887089 0.04583463
##          x6        x7         x8        x9        x10       x11         y
## 1 0.1596740 0.7323217 0.13871334 0.3862818 0.09863575 0.3044642 1.0670591
## 2 0.1445159 0.6097107 0.55719414 0.5423828 0.62344997 0.8328188 2.1846785
## 3 0.1491804 0.2243721 0.25292843 0.7442531 0.25637276 0.5936475 0.3450325
## 4 0.5144343 0.9161710 0.07360617 0.3333324 0.47926656 0.8071966 1.2822883
## 5 0.4928273 0.8027610 0.76264863 0.3249152 0.47714351 0.2940508 0.4845092
## 6 0.6163428 0.3124086 0.60636878 0.4237430 0.83049142 0.1410852 1.0730924
```

```
summary(data)
```

```
##       Date                           x1                  x2
##  Min.   :2024-02-16 09:43:14   Min.   :0.0006248   Min.   :0.0004653
##  1st Qu.:2024-02-16 09:45:03   1st Qu.:0.2639776   1st Qu.:0.2536403
##  Median :2024-02-16 09:46:51   Median :0.4793781   Median :0.5004286
##  Mean   :2024-02-16 09:46:50   Mean   :0.5007171   Mean   :0.4989421
##  3rd Qu.:2024-02-16 09:48:32   3rd Qu.:0.7418928   3rd Qu.:0.7559663
##  Max.   :2024-02-16 09:50:28   Max.   :0.9994045   Max.   :0.9966172
##        x3                 x4                 x5                 x6
##  Min.   :0.002479   Min.   :0.001156   Min.   :0.004609   Min.   :0.003536
##  1st Qu.:0.261979   1st Qu.:0.244459   1st Qu.:0.257612   1st Qu.:0.220893
##  Median :0.497332   Median :0.489092   Median :0.521636   Median :0.479497
##  Mean   :0.489600   Mean   :0.495319   Mean   :0.510449   Mean   :0.491783
##  3rd Qu.:0.720958   3rd Qu.:0.739969   3rd Qu.:0.775596   3rd Qu.:0.731610
##  Max.   :0.992616   Max.   :0.999524   Max.   :0.996372   Max.   :0.999274
##        x7                 x8                 x9                x10
##  Min.   :0.009345   Min.   :0.003516   Min.   :0.001152   Min.   :0.0000653
##  1st Qu.:0.240440   1st Qu.:0.242365   1st Qu.:0.277551   1st Qu.:0.2255640
##  Median :0.497762   Median :0.518984   Median :0.505419   Median :0.5012204
##  Mean   :0.498774   Mean   :0.507688   Mean   :0.503866   Mean   :0.5019742
##  3rd Qu.:0.744130   3rd Qu.:0.760059   3rd Qu.:0.741093   3rd Qu.:0.7772592
##  Max.   :0.999651   Max.   :0.988818   Max.   :0.999500   Max.   :0.9975694
##       x11                 y
##  Min.   :0.000829   Min.   :-0.9416
```

```
##  1st Qu.:0.270312   1st Qu.: 0.2947
##  Median :0.495292   Median : 0.9060
##  Mean   :0.495588   Mean   : 0.9408
##  3rd Qu.:0.730391   3rd Qu.: 1.5042
##  Max.   :0.999941   Max.   : 3.2256
```

**lets try the linear model with the data.**

```
# Fit the linear model
model <- lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11, data = data)

summary(model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 +
##     x10 + x11, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.57560 -0.22931 -0.08698  0.35059  1.15462
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.17316    0.16231   7.228 2.51e-12 ***
## x1           0.77010    0.10111   7.617 1.90e-13 ***
## x2           0.09865    0.10036   0.983   0.3262
## x3           0.08017    0.10278   0.780   0.4358
## x4           1.03880    0.10054  10.333  < 2e-16 ***
## x5          -0.04974    0.10009  -0.497   0.6195
## x6          -0.05759    0.09651  -0.597   0.5510
## x7          -0.17037    0.10156  -1.678   0.0942 .
## x8          -0.02678    0.09980  -0.268   0.7886
## x9          -1.93250    0.10169 -19.005  < 2e-16 ***
## x10         -0.13771    0.09497  -1.450   0.1478
## x11         -0.05209    0.10361  -0.503   0.6154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5784 on 400 degrees of freedom
## Multiple R-squared:  0.5684, Adjusted R-squared:  0.5565
## F-statistic: 47.89 on 11 and 400 DF,  p-value: < 2.2e-16
```

**From above we can see the most important vaiables they are x1,x4,x9,x7 , so lets try to see model with them.**

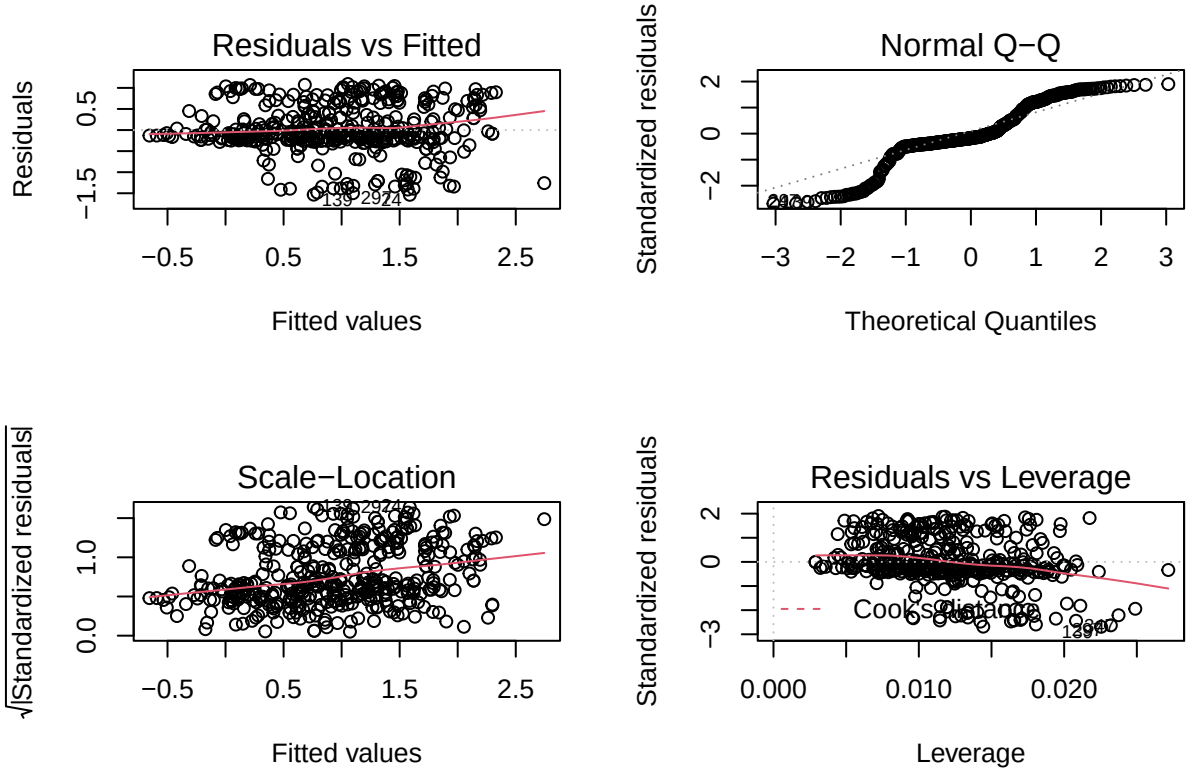
```
# Fit the linear model
model <- lm(y ~ x1 + x4 + x7 + x9 , data = data)

# Check the summary of the model
summary(model)
```

```
##
## Call:
```

```
## lm(formula = y ~ x1 + x4 + x7 + x9, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.53603 -0.22475 -0.09897  0.33487  1.09214
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.09081    0.10221  10.672  < 2e-16 ***
## x1           0.75684    0.10021   7.552 2.84e-13 ***
## x4           1.05737    0.09877  10.706  < 2e-16 ***
## x7          -0.15967    0.09976  -1.601     0.11
## x9          -1.93116    0.10082 -19.154  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5765 on 407 degrees of freedom
## Multiple R-squared:  0.5636, Adjusted R-squared:  0.5593
## F-statistic: 131.4 on 4 and 407 DF,  p-value: < 2.2e-16
```

**Ok we can see how the effect of x7 dispaear on y.**

```
par(mfrow = c(2, 2))  # Set up the plotting area to display 4 plots in a 2x2 grid
plot(model)
```




**Ok there is no pattern but there is a Lot of points not explained by the model.**

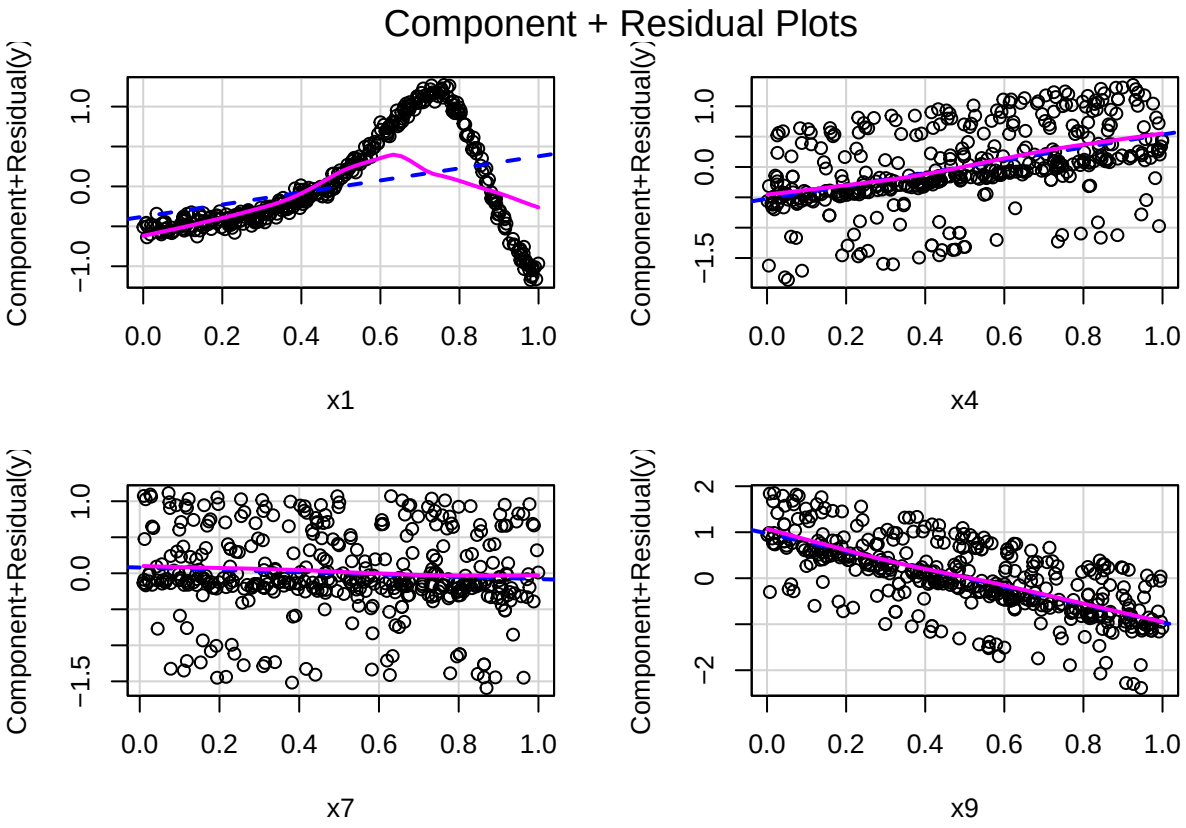
```
install.packages("car")
```

```
## Installation du package dans '/home/o/othmanma/R/x86_64-pc-linux-gnu-library/4.1'
## (car 'lib' n'est pas spécifié)
```

```
library(car)
```

```
## Le chargement a nécessité le package : carData
```

```
crPlots(model)
```




**Here we can see each individual variable, we can see how the X7 is almost a straight line.**

```
max_value <- max(data$y, na.rm = TRUE)
max_value
```

```
## [1] 3.225577
```

```
data_subset <- data[c("x1", "x4","x7","x9", "y")]
data_sorted_desc <- data_subset[order(data_subset$y, decreasing = TRUE), ]
head(data_sorted_desc)
```

```
##            x1        x4        x7          x9        y
## 183 0.7082903 0.8824230 0.9619054 0.039946946 3.225577
## 175 0.6682846 0.6856713 0.1149009 0.006897165 3.169531
## 246 0.7245543 0.7538998 0.7997378 0.080061785 3.124880
## 263 0.6741868 0.8108895 0.7331638 0.064845172 3.053524
## 82  0.6680556 0.8527804 0.3099936 0.145648298 3.001998
## 301 0.7845753 0.9380283 0.4417920 0.241493882 2.898522
```

**we can see the highest value for y is 3.225577 , with specific combination of values for the four variables.**

**Now let see another data.**

**Above is random uniform sampling let see if we add to this data my old data that I made it by my hand in the first place.**

```
data2 <- read.csv("NewDatawithmyhandData.csv", header = TRUE, stringsAsFactors = FALSE, sep = ",")


data2$Date <- as.POSIXct(data2$Date, format = "%Y-%m-%d-%H:%M:%S")

head(data2)
```

```
##                  Date   x1   x2   x3   x4   x5   x6   x7   x8   x9  x10  x11
## 1 2023-12-24 22:09:55 0.10 0.20 0.20 0.30 0.50 0.60 0.70 0.01 0.40 0.44 0.60
## 2 2024-01-11 01:54:36 0.90 0.40 0.99 0.90 0.88 0.44 0.55 0.70 0.80 0.70 0.90
## 3 2024-01-11 01:55:42 0.90 0.40 0.99 0.90 0.88 0.44 0.55 0.70 0.80 0.70 0.90
## 4 2024-01-11 01:55:43 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99
## 5 2024-01-11 01:56:50 0.90 0.90 0.90 0.00 0.90 0.90 0.90 0.90 0.90 0.90 0.90
## 6 2024-01-11 01:58:10 0.40 0.40 0.40 0.50 0.40 0.50 0.70 0.80 0.30 0.70 0.60
##            y
## 1  0.4769565
## 2  0.3593838
## 3  0.3560769
## 4 -0.8278516
## 5 -0.5988448
## 6  1.1523425
```

```
# Fit the linear model
model2 <- lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11, data = data2)

summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 +
##     x10 + x11, data = data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.87920 -0.27821 -0.03508  0.49460  1.20271
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.46868    0.11423  12.858  < 2e-16 ***
## x1           0.82584    0.09918   8.327 9.33e-16 ***
## x2          -0.15921    0.09601  -1.658   0.0979 .
## x3          -0.20371    0.09687  -2.103   0.0360 *
## x4           1.10997    0.09752  11.382  < 2e-16 ***
## x5          -0.19120    0.09780  -1.955   0.0512 .
## x6           0.02366    0.09539   0.248   0.8042
## x7          -0.23407    0.09834  -2.380   0.0177 *
## x8           0.03703    0.09722   0.381   0.7034
## x9          -2.03569    0.09506 -21.414  < 2e-16 ***
```

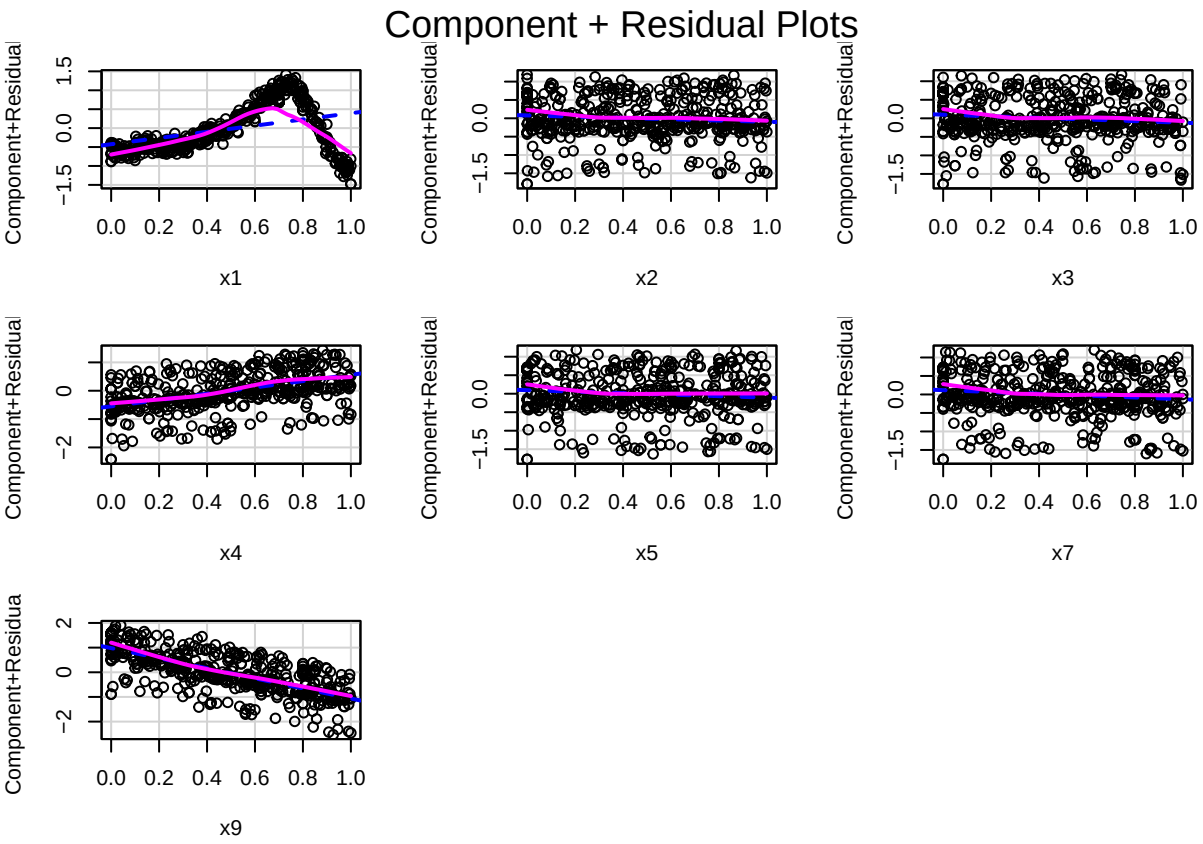
```
## x10         -0.14260    0.09292  -1.535   0.1256
## x11          0.01509    0.10082   0.150   0.8811
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6068 on 466 degrees of freedom
## Multiple R-squared:  0.6257, Adjusted R-squared:  0.6168
## F-statistic: 70.81 on 11 and 466 DF,  p-value: < 2.2e-16
```

**This data made add more variable that effect the y let us see if these impact is significant by plotting each variable.**

```
# Fit the linear model
model3 <- lm(y ~ x1 + x2 + x3 + x4 + x5 +x7+ x9, data = data2)
summary(model3)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x7 + x9, data = data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.87102 -0.27407 -0.05358  0.51319  1.24268
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.46829    0.09924  14.795  < 2e-16 ***
## x1           0.81806    0.09845   8.310 1.04e-15 ***
## x2          -0.17381    0.09509  -1.828   0.0682 .
## x3          -0.21737    0.09555  -2.275   0.0234 *
## x4           1.11203    0.09536  11.661  < 2e-16 ***
## x5          -0.20248    0.09634  -2.102   0.0361 *
## x7          -0.24640    0.09754  -2.526   0.0119 *
## x9          -2.04301    0.09414 -21.702  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6059 on 470 degrees of freedom
## Multiple R-squared:  0.6237, Adjusted R-squared:  0.6181
## F-statistic: 111.3 on 7 and 470 DF,  p-value: < 2.2e-16
```

```
crPlots(model3)
```




**From above we can see how the just three variable have real impact(x1,x4,x9) and the rest like a straight line, like the model without the hand data.**

```
max_value <- max(data2$y, na.rm = TRUE)
max_value
```

```
## [1] 3.356876
```

**We get the biggest for y is 3.356876 , now let see the value for each varaible.**

```
data_subset <- data2[c("x1", "x4","x7","x9", "y")]
data_sorted_desc2 <- data_subset[order(data_subset$y, decreasing = TRUE), ]
head(data_sorted_desc2)
```

```
##     x1  x4 x7 x9        y
## 39 0.7 0.7  0  0 3.356876
## 29 0.7 0.7  0  0 3.356482
## 35 0.7 0.7  0  0 3.355810
## 36 0.7 0.7  0  0 3.354668
## 30 0.7 0.7  0  0 3.354628
## 42 0.7 0.7  0  0 3.354090
```

**We can see the data biased to 0.7 since it was emprical by me additional to the random data.**