Regression

Code ▾

Linear Regression is a method used to find similarities between a dependent variable and 1 or more independent variable. The goal of linear regression model is to predict the dependent value by using the independent valeus. Some strengths of linear regression may include that is widely used and it can be used to understand relationships betwen objects. One weakness may include that it assumes a linear relationship but it may not be the case all the time.

Hide

```
set.seed(3)
data <- read.csv(file = 'desktop/creditcard.csv')
dim(data)
```

```
[1] 284807     31
```

Hide

```
head(data)
```

	T...	V1	V2	V3	V4	V5	V6	V7
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	0	-1.3598071	-0.07278117	2.5363467	1.3781552	-0.33832077	0.46238778	0.23959855
2	0	1.1918571	0.26615071	0.1664801	0.4481541	0.06001765	-0.08236081	-0.07880298
3	1	-1.3583541	-1.34016307	1.7732093	0.3797796	-0.50319813	1.80049938	0.79146096
4	1	-0.9662717	-0.18522601	1.7929933	-0.8632913	-0.01030888	1.24720317	0.23760894
5	2	-1.1582331	0.87773675	1.5487178	0.4030339	-0.40719338	0.09592146	0.59294075
6	2	-0.4259659	0.96052304	1.1411093	-0.1682521	0.42098688	-0.02972755	0.47620095

6 rows | 1-9 of 31 columns

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

Hide

```
set.seed(3)
i<-sample(1:nrow(data), nrow(data)*0.8, replace = FALSE)
train<-data[i,]
test<-data[-i,]
```

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

Hide

```
dim(train)
```

```
[1] 227845     31
```

Hide

```
head(train)
```

	Time <dbl>	V1 <dbl>	V2 <dbl>	V3 <dbl>	V4 <dbl>	V5 <dbl>	V6 <dbl>	
218087	141166	1.7933193	0.01984035	-1.9296190	1.34779239	0.5610834	-1.3402134	1.0
236218	148714	-0.8220090	0.72748442	-0.8316150	0.98524199	2.2795032	-0.5769322	0.9
229706	146020	2.0397723	0.61324571	-2.3616822	0.76673928	0.5995394	-1.7172058	0.3
83869	60061	-0.5231397	0.58988326	3.2996937	3.05804626	-0.4433193	1.5899776	-0.2
197613	132097	1.8249885	-0.68401799	-0.7133819	0.09200484	-0.3199603	0.1725305	-0.4
268264	163135	-0.2646561	1.00158569	-1.1063368	-1.25530667	1.5773387	-0.5076532	1.6

6 rows | 1-9 of 31 columns

Hide

```
names(train)
```

```
 [1] "Time"   "V1"     "V2"     "V3"     "V4"     "V5"     "V6"    
 [8] "V7"     "V8"     "V9"     "V10"    "V11"    "V12"    "V13"   
[15] "V14"    "V15"    "V16"    "V17"    "V18"    "V19"    "V20"   
[22] "V21"    "V22"    "V23"    "V24"    "V25"    "V26"    "V27"   
[29] "V28"    "Amount" "Class" 
```

Hide

```
colSums(is.na(train))
```

```
  Time     V1     V2     V3     V4     V5     V6     V7     V8     V9 
     0      0      0      0      0      0      0      0      0      0 
   V10    V11    V12    V13    V14    V15    V16    V17    V18    V19 
     0      0      0      0      0      0      0      0      0      0 
   V20    V21    V22    V23    V24    V25    V26    V27    V28 Amount 
     0      0      0      0      0      0      0      0      0      0 
 Class 
     0 
```

Hide

```
str(train)
```

```
'data.frame':	227845 obs. of  31 variables:
 $ Time  : num  141166 148714 146020 60061 132097 ...
 $ V1    : num  1.793 -0.822 2.04 -0.523 1.825 ...
 $ V2    : num  0.0198 0.7275 0.6132 0.5899 -0.684 ...
 $ V3    : num  -1.93 -0.832 -2.362 3.3 -0.713 ...
 $ V4    : num  1.348 0.985 0.767 3.058 0.092 ...
 $ V5    : num  0.561 2.28 0.6 -0.443 -0.32 ...
 $ V6    : num  -1.34 -0.577 -1.717 1.59 0.173 ...
 $ V7    : num  1.081 0.938 0.392 -0.201 -0.491 ...
 $ V8    : num  -0.619 -0.118 -0.411 0.362 0.133 ...
 $ V9    : num  -0.356 -0.25 0.439 -0.294 0.779 ...
 $ V10   : num  0.267 0.222 -1.221 0.575 0.123 ...
 $ V11   : num  -0.4948 0.4926 0.3239 0.0458 0.5658 ...
 $ V12   : num  0.8447 -0.9152 0.0279 0.3468 0.8999 ...
 $ V13   : num  1.1326 -2.2665 0.4242 -0.3725 0.0711 ...
 $ V14   : num  0.604 -0.742 -2.618 -0.885 0.137 ...
 $ V15   : num  0.1103 -0.487 1.0684 -1.4965 -0.0971 ...
 $ V16   : num  -0.512 -0.397 0.692 -0.442 0.758 ...
 $ V17   : num  -0.448 0.94 1.8 0.255 -0.902 ...
 $ V18   : num  -0.635 1.131 1.126 0.572 0.174 ...
 $ V19   : num  -0.509 0.996 -0.773 1.813 0.379 ...
 $ V20   : num  0.0696 0.1527 -0.1313 0.272 0.041 ...
 $ V21   : num  0.2258 -0.0806 0.1085 -0.0874 -0.188 ...
 $ V22   : num  0.519 0.237 0.546 0.209 -0.718 ...
 $ V23   : num  -0.1445 -0.4062 -0.0462 -0.2926 0.3409 ...
 $ V24   : num  0.0275 0.1415 -0.2154 0.0288 0.3405 ...
 $ V25   : num  0.4489 0.0203 0.2581 0.0884 -0.6436 ...
 $ V26   : num  -0.5 -0.413 -0.099 0.294 0.193 ...
 $ V27   : num  -0.0471 0.4572 0.0262 -0.0263 -0.066 ...
 $ V28   : num  -0.03701 0.09948 0.00825 -0.13511 -0.03495 ...
 $ Amount: num  144 21 1 29 98 ...
 $ Class : int  0 0 0 0 0 0 0 0 0 0 ...
```

Hide

```
install.packages("vioplot")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/contrib/4.2/vioplot_0.4.0.tgz'
Content type 'application/x-gzip' length 1385815 bytes (1.3 MB)
==================================================
downloaded 1.3 MB
```

```


The downloaded binary packages are in
	/var/folders/fx/fht6tvb95xldy2488rl0ctj00000gn/T//RtmpNP8aNE/downloaded_packages
```

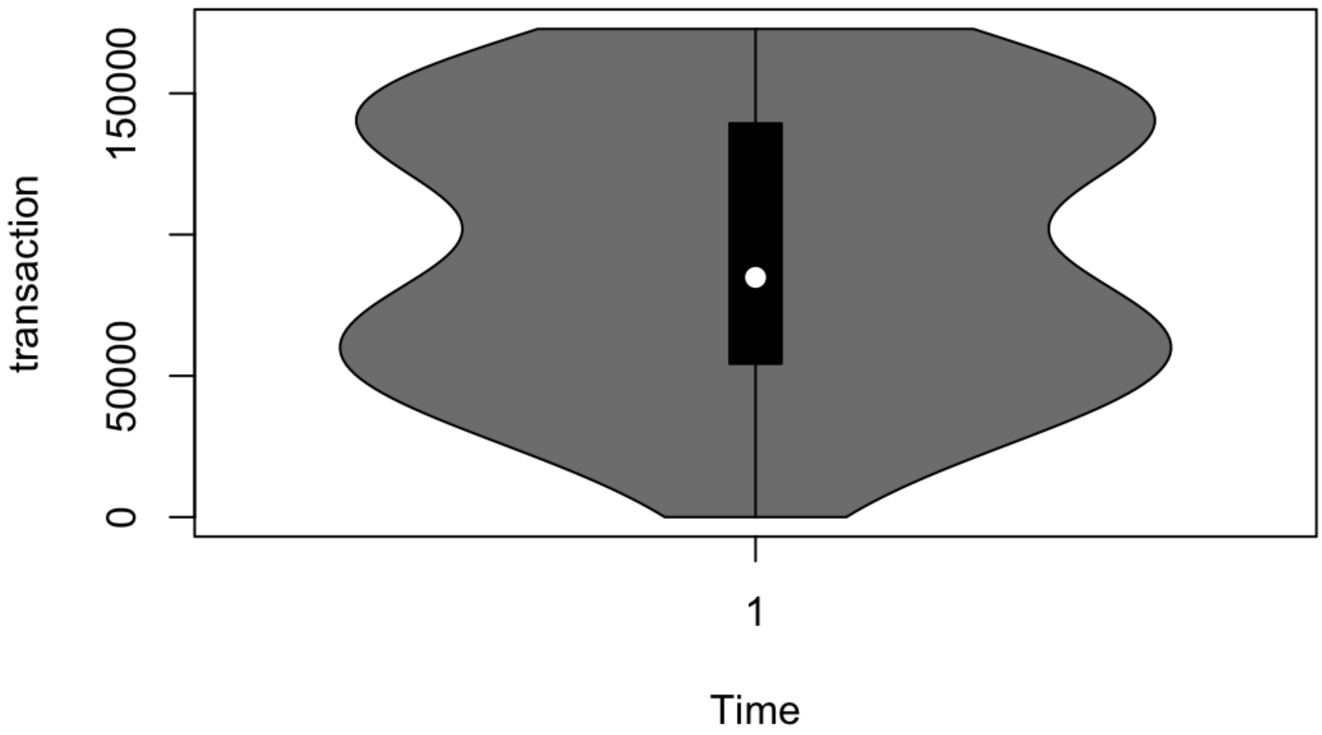
Hide

```
library("vioplot")
```

```
Loading required package: sm
Package 'sm', version 2.2-5.7: type help(sm) for summary information
Loading required package: zoo

Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric
```

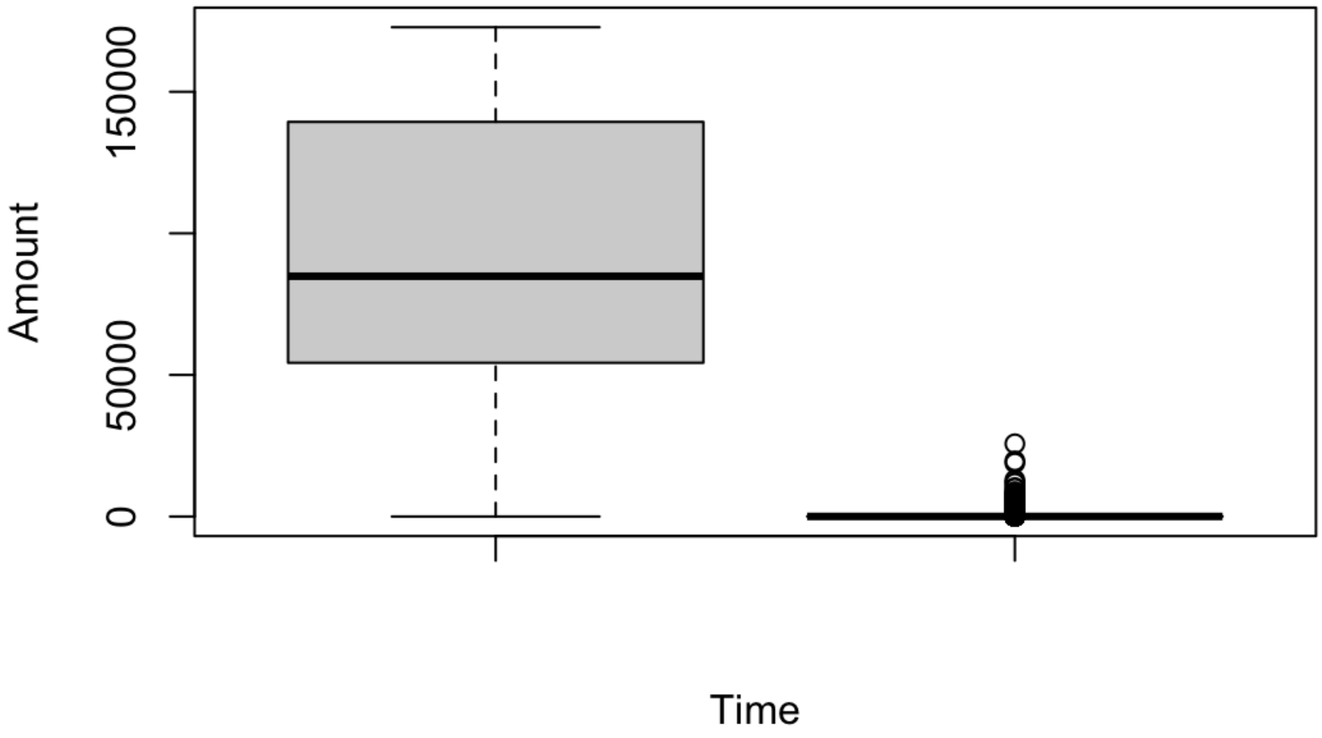
Hide

```
vioplot(train$Time, xlab = "Time", ylab = "transaction")
```



Hide

```
boxplot(train$Time, train$Amount, xlab="Time", ylab = "Amount")
```



Hide

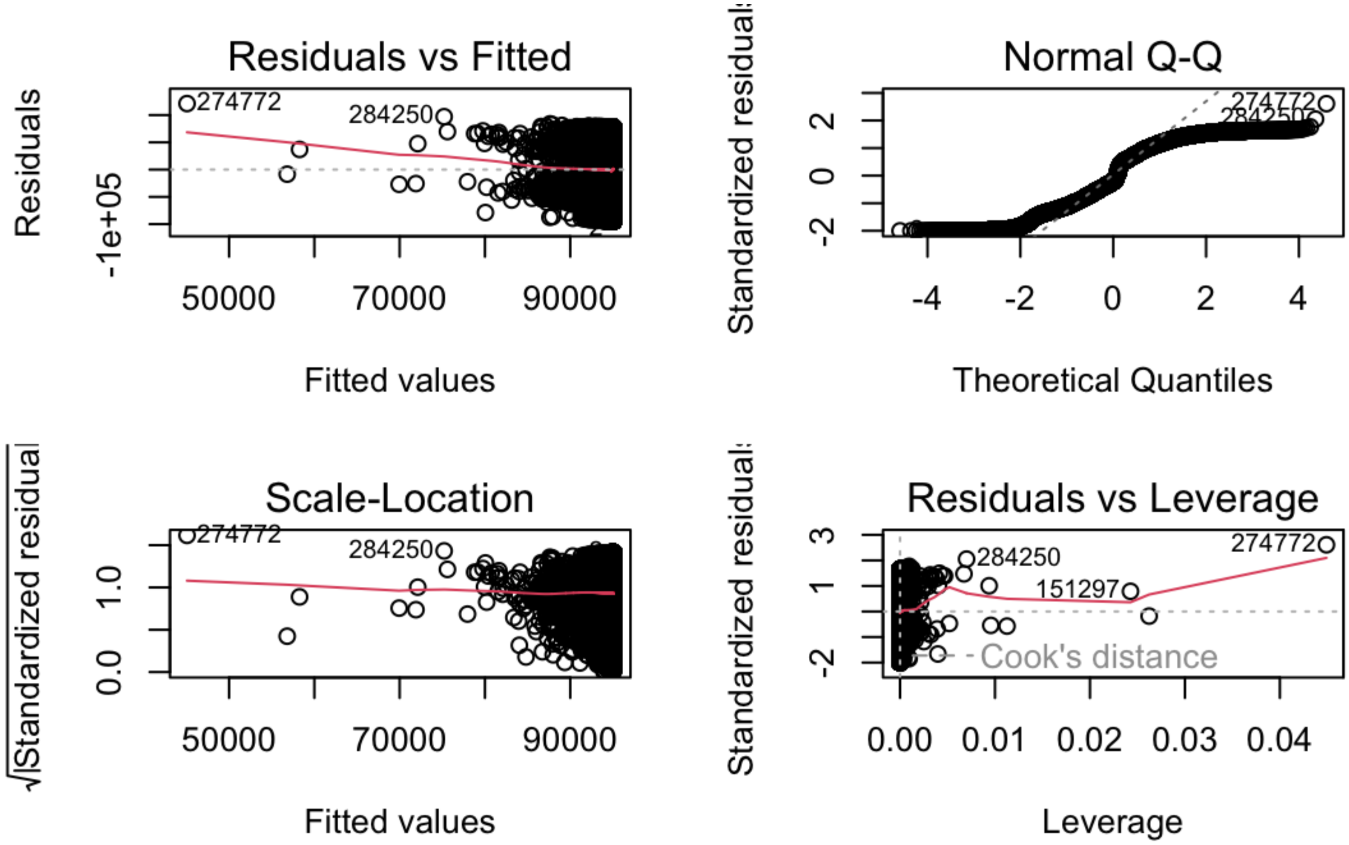
```
lm1<-lm(Time~Amount, data = train)
summary(lm1)
```

```

Call:
lm(formula = Time ~ Amount, data = train)

Residuals:
   Min     1Q Median     3Q    Max 
-95058 -40574 -10074  44482 121125 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 95062.9008   105.3560 902.302  < 2e-16 ***
Amount         -1.9458     0.3929  -4.952 7.35e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 47480 on 227843 degrees of freedom
Multiple R-squared:  0.0001076, Adjusted R-squared:  0.0001032 
F-statistic: 24.52 on 1 and 227843 DF,  p-value: 7.352e-07
```

From this summary we can learn that the p value is extremely small which represents that there is a trong relationship between time and amount. the multiple R squared value is very low which suggests that the model explains very little of the variance of time. the residual standard error is high which indicates that the prediction would not be as accurate.

Hide

```
par(mfrow=c(2,2))
plot(lm1)
```



Hide

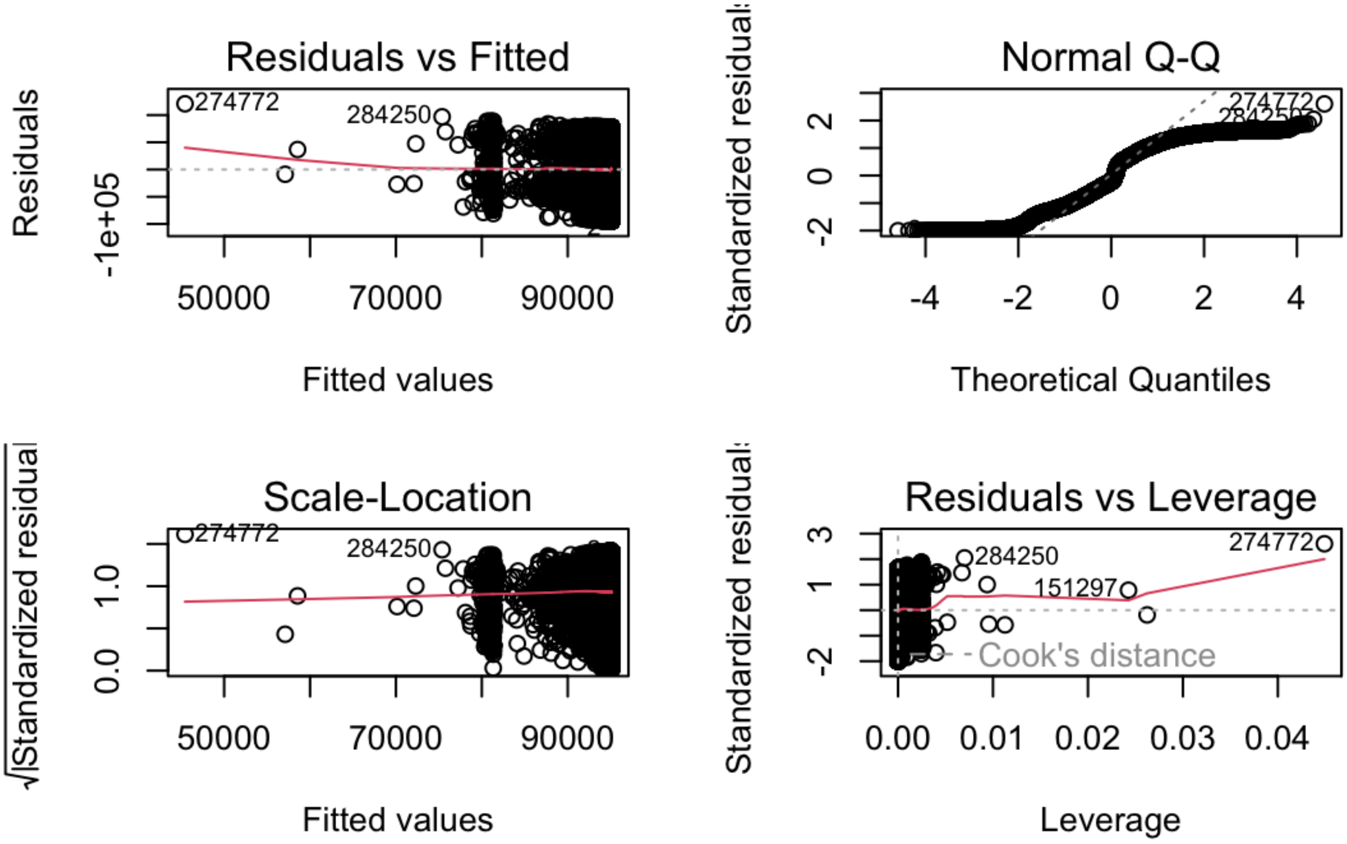
```
lm2<-lm(Time~Amount+Class, data = train)
summary(lm2)
```

]

```

Call:
lm(formula = Time ~ Amount + Class, data = train)

Residuals:
   Min     1Q Median     3Q    Max 
-95081 -40570 -10082  44481 120752 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  9.509e+04  1.054e+02 901.939  < 2e-16 ***
Amount      -1.932e+00  3.929e-01  -4.918 8.77e-07 ***
Class       -1.374e+04  2.355e+03  -5.835 5.39e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 47470 on 227842 degrees of freedom
Multiple R-squared:  0.000257,	Adjusted R-squared:  0.0002482 
F-statistic: 29.29 on 2 and 227842 DF,  p-value: 1.918e-13
```

Hide

```
par(mfrow=c(2,2))
plot(lm2)
```



Hide

```
lm3<-lm(Time ~ Time+Amount+Class,  data = train)
```

```
Warning: the response appeared on the right-hand side and was droppedWarning: problem wi
th term 1 in model.matrix: no columns are assigned
```

Hide

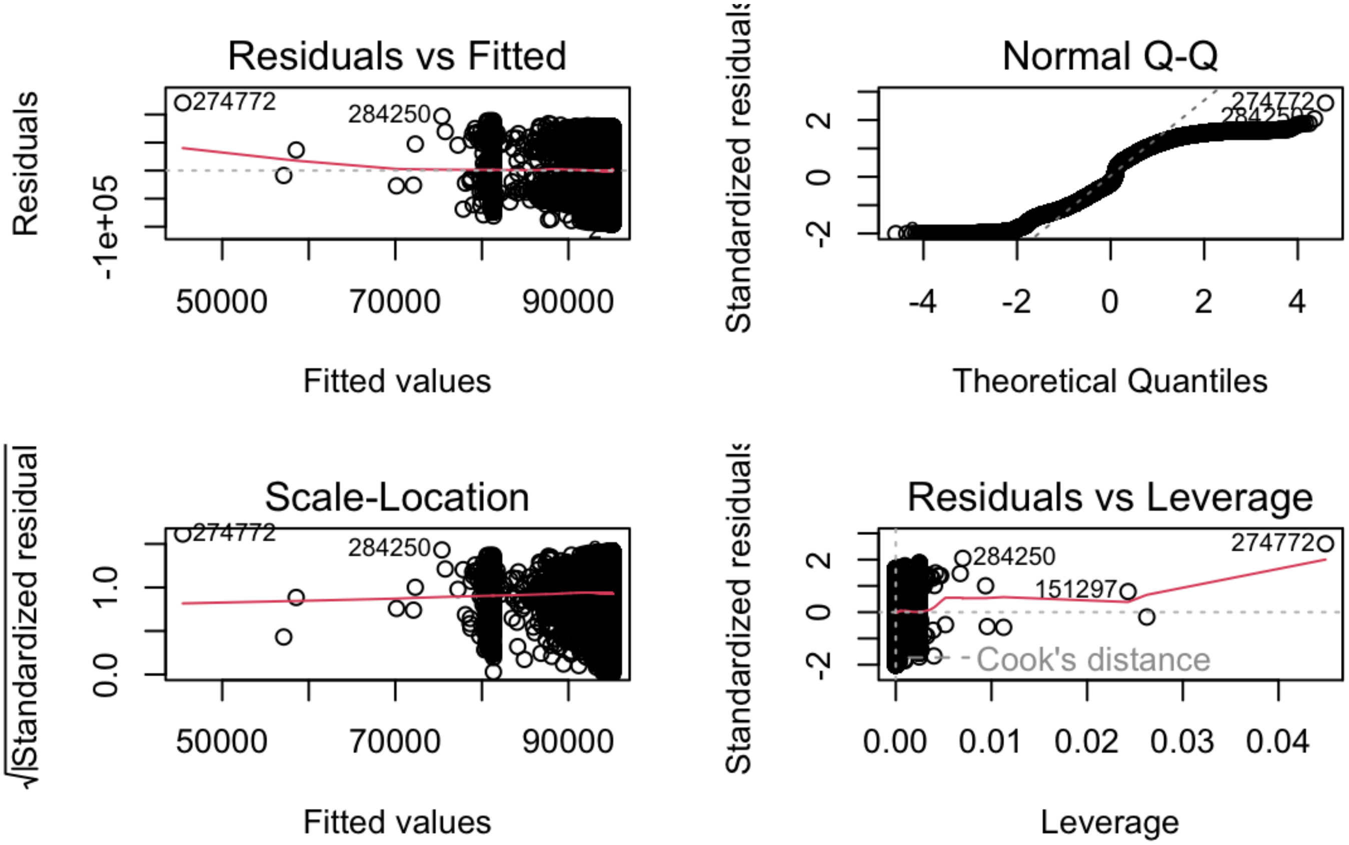
```
summary(lm3)
```

```


Call:
lm(formula = Time ~ Time + Amount + Class, data = train)

Residuals:
   Min     1Q Median     3Q    Max 
-95081 -40570 -10082  44481 120752 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  9.509e+04  1.054e+02 901.939  < 2e-16 ***
Amount      -1.932e+00  3.929e-01  -4.918 8.77e-07 ***
Class       -1.374e+04  2.355e+03  -5.835 5.39e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 47470 on 227842 degrees of freedom
Multiple R-squared:  0.000257,  Adjusted R-squared:  0.0002482 
F-statistic: 29.29 on 2 and 227842 DF,  p-value: 1.918e-13
```

Hide

```
par(mfrow=c(2,2))
plot(lm3)
```

```
Warning: the response appeared on the right-hand side and was droppedWarning: problem wi
th term 1 in model.matrix: no columns are assigned
```


Residuals vs Fitted
Residuals
-1e+05
Fitted values
50000
70000
90000
274772
284250
Normal Q-Q
Standardized residuals
Theoretical Quantiles
-4
-2
0
2
4
274772
284250
Scale-Location
√|Standardized residuals|
0.0
1.0
Residuals vs Leverage
Leverage
0.00
0.01
0.02
0.03
0.04
284250
151297
Cook's distance

Hide

```
pred1<-predict(lm1, newdata = test)
cor1 <- cor(pred1, test$Time)
mse1 <- mean ((pred1-test$Time)^1)
rmse1 <-sqrt(mse1)
print(paste('correlation: ',cor1))
```

```
[1] "correlation:  0.0115679423668671"
```

Hide

```
print(paste('mse: ',mse1))
```

```
[1] "mse:  385.643485196248"
```

Hide

```
print(paste('rmse: ',rmse1))
```

```
[1] "rmse:  19.6378075455548"
```

Hide

```
pred2<-predict(lm2, newdata = test)
cor2 <- cor(pred2, test$Time)
mse2 <- mean ((pred2-test$Time)^2)
rmse2 <-sqrt(mse2)
print(paste('correlation: ',cor2))
```

```
[1] "correlation:  0.0170486625902318"
```

Hide

```
print(paste('mse: ',mse2))
```

```
[1] "mse:  2257396538.92481"
```

Hide

```
print(paste('rmse: ',rmse2))
```

```
[1] "rmse:  47512.0672979488"
```

Hide

```
pred3<-predict(lm3, newdata = test)
```

```
Warning: prediction from a rank-deficient fit may be misleading
```

Hide

```
cor3 <- cor(pred3, test$Time)
mse3 <- mean ((pred3-test$Time)^2)
rmse3 <-sqrt(mse3)
print(paste('correlation: ',cor3))
```

```
[1] "correlation:  0.0170486625902318"
```

Hide

```
print(paste('mse: ',mse3))
```

```
[1] "mse:  2257396538.92481"
```

Hide

```
print(paste('rmse: ',rmse3))
```

```
[1] "rmse:  47512.0672979488"
```

Output: Fromt the correlation, mse and rmse we notice that the first linear regression model is the best one. We see that because of the low mse and rmse of the the first one compared to the other two. this may happen because the variables in the second and third models may not connect with the prediction. They divert the data in some other way.