Influential Analysis

Table of Contents

There are many important factors in building a better model, one of such factors is analysis of Influences. When we think of influence, we think about Regression line and its balancing act between different data points. It's important to understand different type of influencers that we might come across.

Outlier: These are the data points with extreme value of the response variable (Y)

Leverage: These are the data points with extrema value of predictor variable (X)

Influential: Some combination of Outlier and Leverage makes data point influential.

A datapoint, which can change (pull toward / push away) the regression line if added or removed in the model is influential data point. It's very easy to identify such data points in simple linear model, but it gets complex as our model becomes more and more complex.

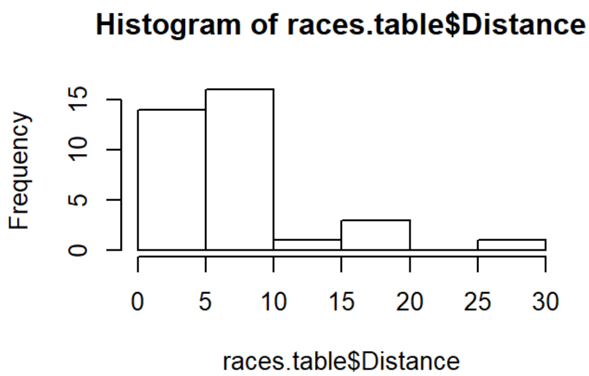
I will see how to identify such outliers and influential data points using some mathematical technique.

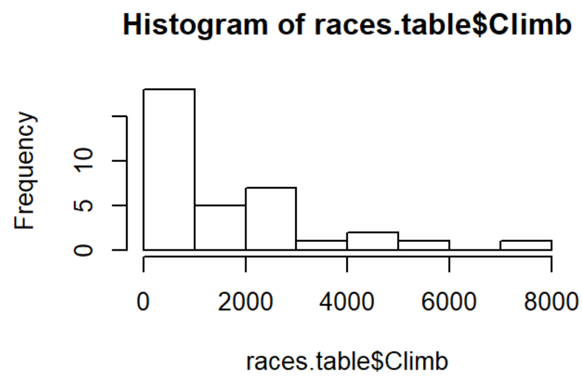
With below set of data we will try to predict TIME~ . by using all the predictors.

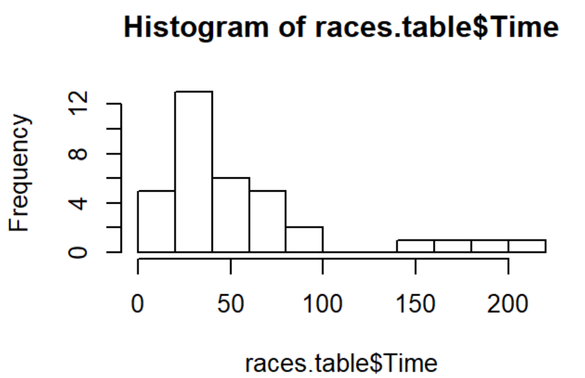
```
url = 'http://www.statsci.org/data/general/hills.txt'
races.table = read.table(url, header=TRUE, sep='\t')


head(races.table)
##           Race Distance Climb   Time
## 1 Greenmantle      2.5   650 16.083
## 2    Carnethy      6.0  2500 48.350
## 3 CraigDunain      6.0   900 33.650
## 4       BenRha      7.5   800 45.600
## 5    BenLomond      8.0  3070 62.267
```

```
## 6     Goatfell       8.0   2866 73.217
```

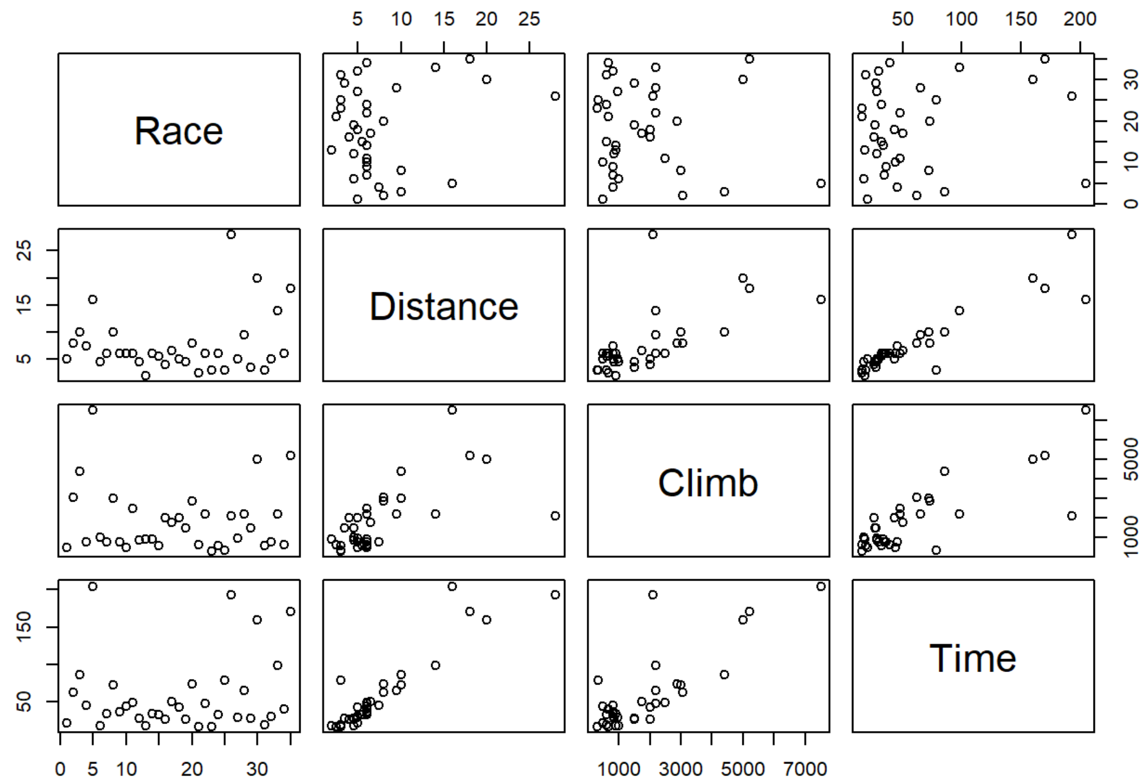
Histogram of each data field is as shown below:







Multivariate plot showing scatterplot of each data fields with others.



Model

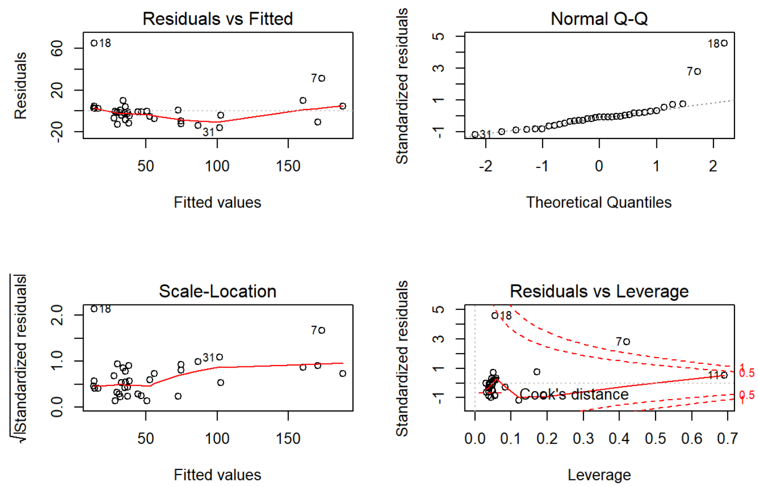
I created 1st model1, and then I can see from the Residual vs Fitted plot that data point 7,18 and 31 are outlier , lets build another model by dropping 7th and 18th data points.

We also see the residual is right skewed, is not giving normalize residual.

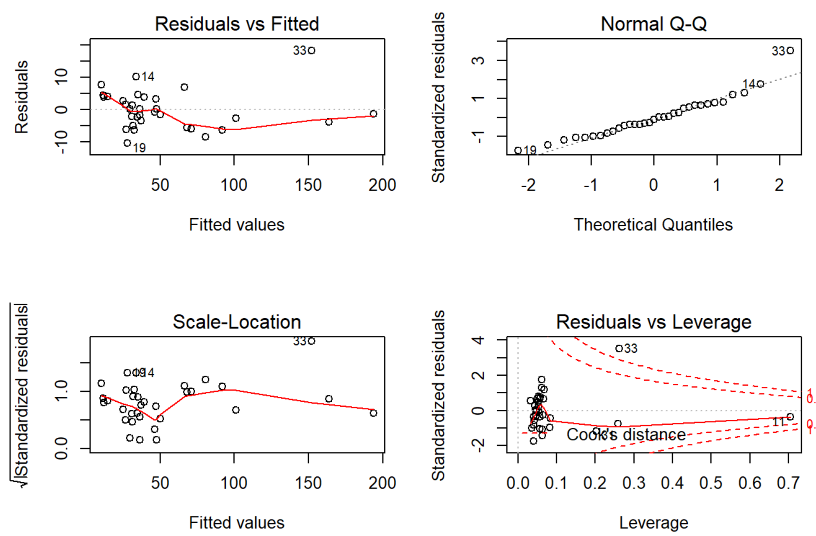
From Model 2, dropping 14 and 33 data points to build model3.

Model	**Plot/ Summary**

model1 <- lm(Time~Distance+Climb, data = races.table)

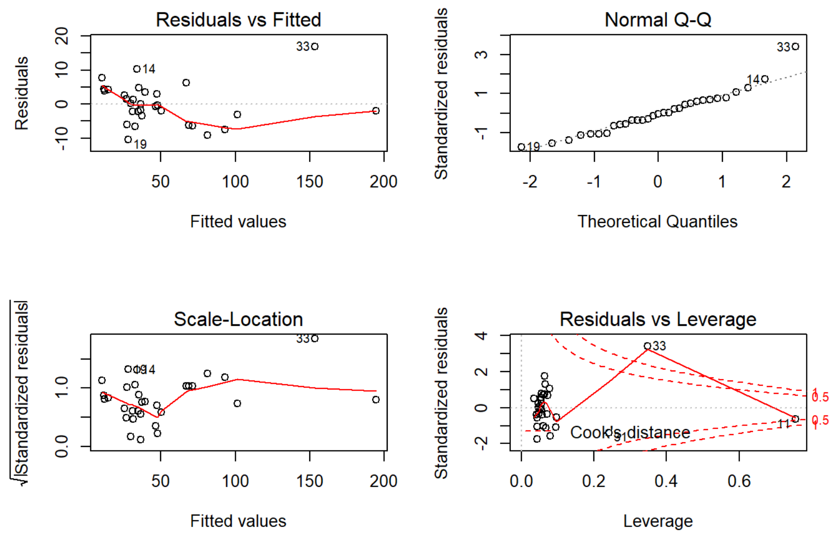


model2 <- lm(Time~Distance+Climb, data = races.table2)




```
summary(model2)

##
## Call:
## lm(formula = Time ~ Distance + Climb, data = races.table2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3827  -3.8818  -0.6667   3.8213  18.3101
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.361646   1.897608  -5.460 6.35e-06 ***
## Distance      6.692114   0.254338  26.312  < 2e-16 ***
## Climb         0.008047   0.001063   7.573 1.91e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.054 on 30 degrees of freedom
## Multiple R-squared:  0.9824, Adjusted R-squared:  0.9812
## F-statistic: 835.3 on 2 and 30 DF,  p-value: < 2.2e-16
```

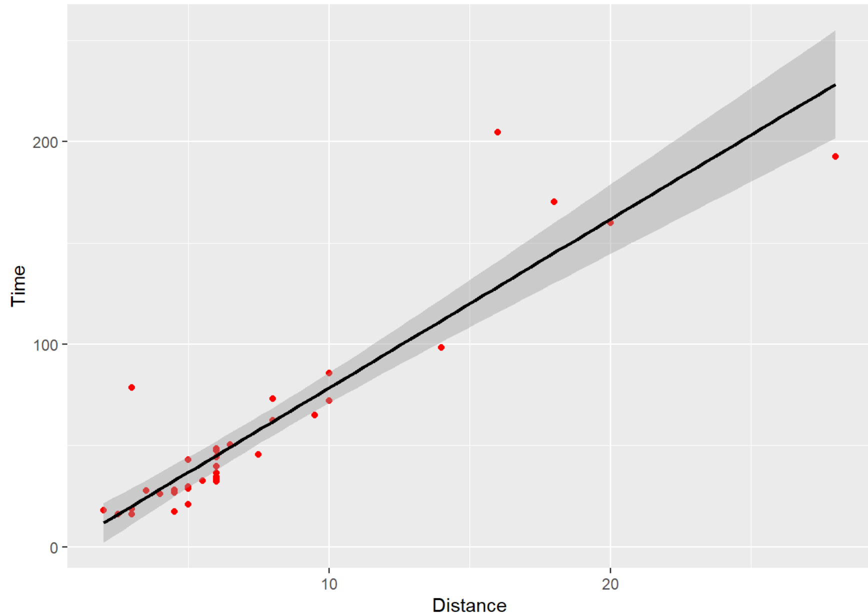
races.table3 <- races.table2[-c(14,33),]
model3 <- lm(Time~Distance+Climb, data = races.table3)




```
summary(model3)
##
## Call:
## lm(formula = Time ~ Distance + Climb, data = races.table3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4730  -3.3118  -0.2929   3.7159  16.9104
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.600885   2.056544  -5.155 1.82e-05 ***
## Distance      6.705906   0.263589  25.441  < 2e-16 ***
## Climb         0.008314   0.001121   7.415 4.48e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.136 on 28 degrees of freedom
## Multiple R-squared:  0.9789, Adjusted R-squared:  0.9774
## F-statistic: 649.9 on 2 and 28 DF,  p-value: < 2.2e-16
```

Below plots how data stand along the regression line , in red.

```
ggplot(races.table, aes(Distance, Time)) +
  geom_point(fill = "indianred4", color="red") +
  geom_smooth(method = "lm",  color = "black")
```

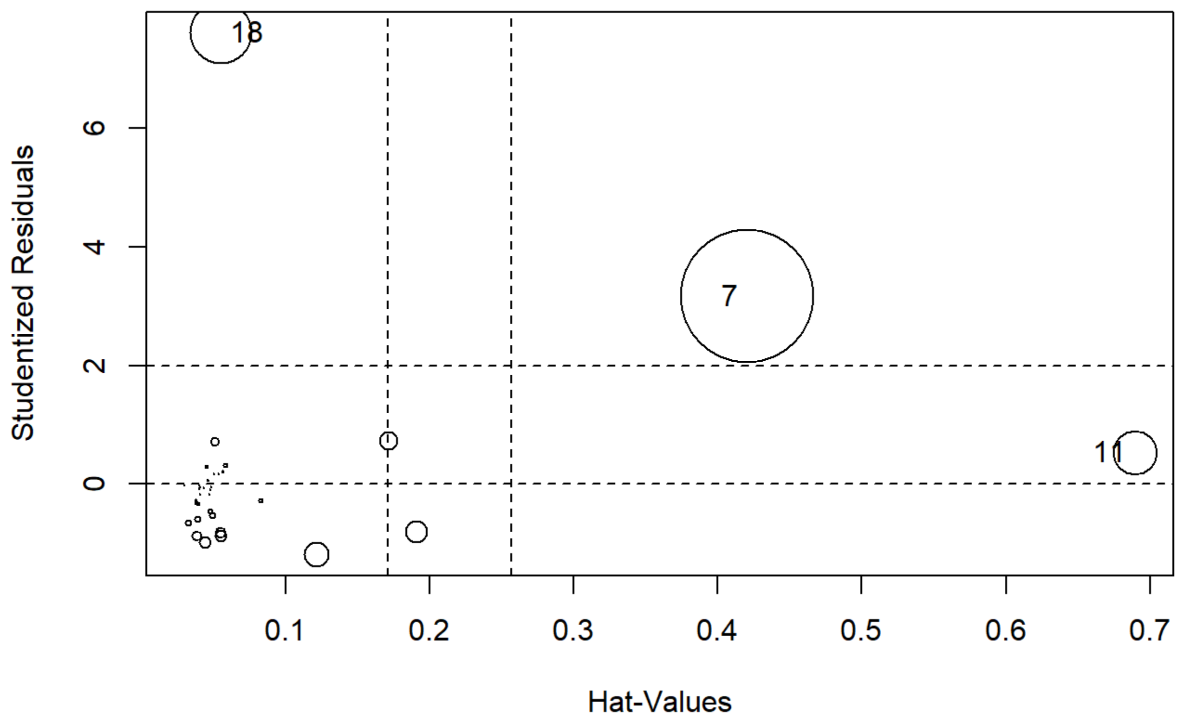


Standardized Residual

: The standardized residual is the residual divided by its standard deviation.

The influence package rom car library also gives 7,18 and 11.

```
library(MASS)
car::influencePlot(model1)
```




```
##       StudRes        Hat      CookD
## 7   3.1689798 0.42043463 1.8933487
## 11 0.5268576 0.68981613 0.2105214
## 18 7.6108449 0.05535523 0.4071560
plot(resid(model1))
```

Creating a new table to store all the info so that we can calculate if proposed data point is possible influential data point or not, from the below result which uses studentized Residual to check if the data point is influential or no.,

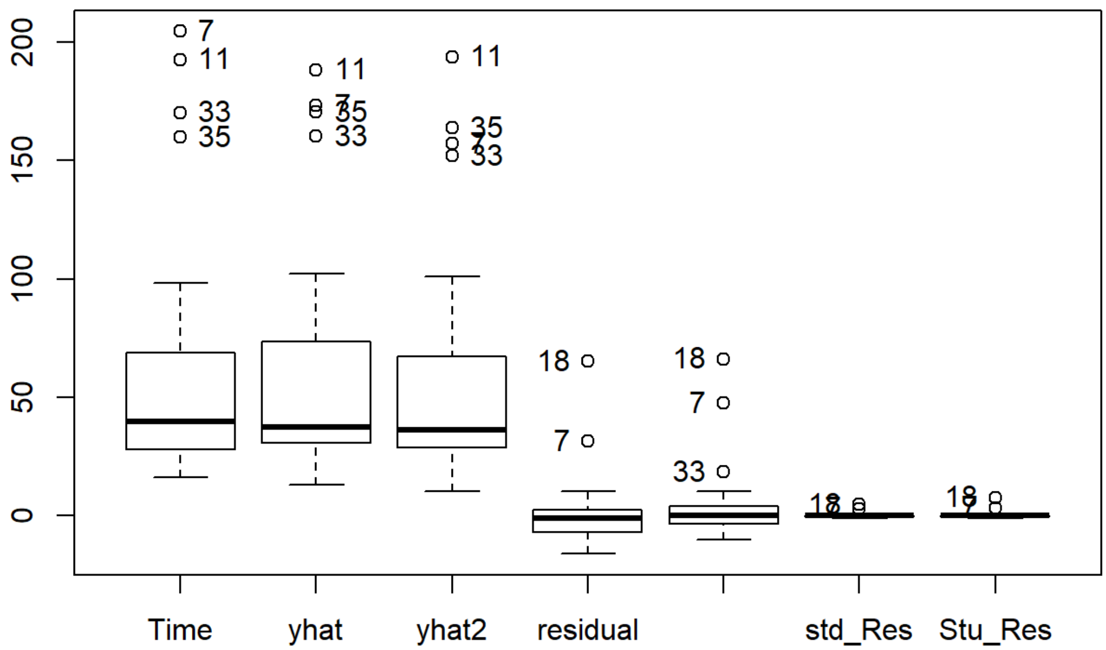
"Time" = races.table$Time   Actual Time taken from the races.table
"yhat" =predict(model1)   Predicted value from Model1on races.table
"yhat2" =predict(model2,races.table)   Predicted value from Model2 on races.table
"residual" = resid(model1) ,   Residuals from Model1
"deleted_residual" = races.table$Time - predict(model2,races.table),
. deleted Residual = Actual Value – Predicted Value from Model 2
"std_Res" = rstandard(model1), #std = residual divided by its standard deviation of residual
"Stu_Res" = studres(model1))   Calculated Stud Residual

```
Result<- data.frame(
            "Time" = races.table$Time,
```

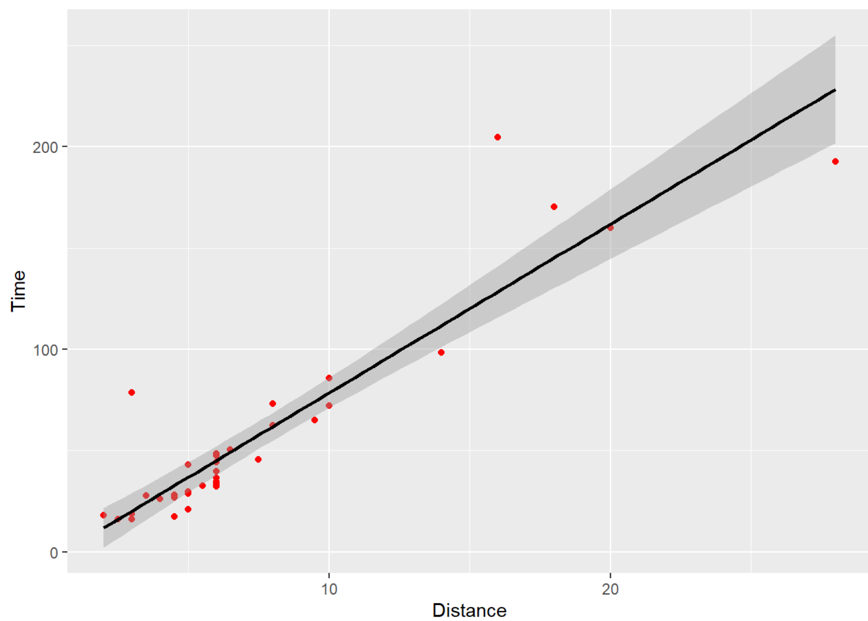
```
            "yhat" =predict(model1),
            "yhat2" =predict(model2,races.table),
            "residual" = resid(model1) ,
            "deleted_residual" = races.table$Time - predict(model2,races.table
),
            "std_Res" = rstandard(model1),  #std = residual divided by its sta
ndard deviation of residual
            "Stu_Res" = studres(model1))
```

Identify all outliers by above table :

car::Boxplot(Result, id=list(n=Inf)) # identify all outliers



Is this a large deleted residual? (data point after residual) Well, we can tell from the plot in this simple linear regression case that the red data points are influential, and so this deleted residual must be considered large.



Studentized residuals

But, in general, how large is large? Unfortunately, there's not a straightforward answer to that question. Deleted residuals depend on the units of measurement just as the ordinary residuals do. We can solve this problem by dividing each deleted residual by an estimate of its standard deviation. That's where "studentized residuals" come into play.

A studentized residual (sometimes referred to as an "externally studentized residual" or a "deleted t residual") is:

```
df = 34-2-2
Result<- mutate(Result,
                cstud = deleted_residual/sd(deleted_residual),
                tval= paste(round(qt(c(.025, .999), df),3) ,collapse="--"),
         posib_inf =
         ifelse( (qt(.025, df)< cstud) & (cstud < qt(.975, df)) , "T " ,"F"))
```

Above we saw in the plots for Model2 and Model3 , we didn't notice much difference in the models output. we see that data point 7 and 18 are the only influential data points.

Result out put:

CSTUD = Calculated studentized residual

Tval = T value between the CI of .025-.999 with degree of freedom 30.
`posib_inf = T if not influential else "F"`

Filter

	Time	yhat	yhat2	residual	deleted_residual	std_Res	Stu_Res	cstud	tval	posib_inf
1	16.083	13.73399	11.59907	2.3490079	4.4839250	0.16454678	0.16202389	0.304691440	-2.042--3.385	T
2	48.350	55.93547	49.90810	-7.5854713	-1.5581003	-0.53015742	-0.52411478	-0.105875950	-2.042--3.385	T
3	33.650	38.25881	37.03318	-4.6088146	-3.3831789	-0.32025768	-0.31572031	-0.229893595	-2.042--3.385	T
4	45.600	46.48096	46.26667	-0.8809572	-0.6666666	-0.06153956	-0.06057395	-0.045301294	-2.042--3.385	T
5	62.267	74.66869	67.87902	-12.4016916	-5.6120181	-0.86942853	-0.86602567	-0.381347565	-2.042--3.385	T
6	73.217	72.41492	66.23747	0.8020821	6.9795343	0.05598020	0.05510126	0.474272956	-2.042--3.385	T
7	204.617	173.35458	157.06337	31.2624197	47.5536348	2.79819456	3.16897977	3.231362124	-2.042--3.385	F
8	36.367	37.15402	36.22850	-0.7870236	0.1385037	-0.05476367	-0.05390372	0.009411598	-2.042--3.385	T
9	29.750	30.93607	29.53638	-1.1860679	0.2136173	-0.08249712	-0.08120651	0.014515710	-2.042--3.385	T
10	39.750	35.49684	35.02147	4.2531630	4.7285276	0.29667307	0.29240316	0.321312662	-2.042--3.385	T
11	192.667	188.31133	193.91587	4.3556673	-1.2488680	0.53290726	0.52685755	-0.084863012	-2.042--3.385	T
12	43.050	44.19356	39.19257	-1.1435604	3.8574262	-0.07967326	-0.07842627	0.262119625	-2.042--3.385	T
13	65.000	74.38394	70.91645	-9.3839431	-5.9164499	-0.65001853	-0.64404747	-0.402034299	-2.042--3.385	T
14	44.133	33.83965	33.81445	10.2933495	10.3185515	0.72009604	0.71456843	0.701165675	-2.042--3.385	T
15	26.933	35.56063	31.82310	-8.6276273	-4.8901041	-0.59963166	-0.59353199	-0.332292098	-2.042--3.385	T
16	72.250	86.33125	80.69997	-14.0812493	-8.4499674	-0.98152450	-0.98094545	-0.574191747	-2.042--3.385	T
17	98.417	102.36474	101.03096	-3.9477438	-2.6139609	-0.28093370	-0.27685089	-0.177623735	-2.042--3.385	T
18	78.650	13.52860	12.53108	65.1214032	66.1189160	4.56558067	7.61084489	4.492909148	-2.042--3.385	F
19	17.417	30.03667	27.79969	-12.6196721	-10.3826911	-0.87696131	-0.87371292	-0.705524089	-2.042--3.385	T
20	32.567	31.83546	31.27307	0.7315363	1.2939257	0.05103257	0.05023090	0.087924770	-2.042--3.385	T
21	15.950	12.97620	12.12874	2.9737987	3.8212573	0.20862403	0.20547819	0.259661878	-2.042--3.385	T
22	27.900	29.34267	25.13099	-1.4426716	2.7690095	-0.10076608	-0.09919485	0.188159588	-2.042--3.385	T
23	47.633	52.62110	47.49405	-4.9880982	0.1389475	-0.34686042	-0.34204132	0.009441752	-2.042--3.385	T
24	17.933	13.38699	10.26472	4.5460082	7.6682753	0.31923417	0.31470807	0.521074247	-2.042--3.385	T
25	18.683	16.29057	14.54279	2.3924256	4.1402095	0.16732023	0.16475719	0.281335304	-2.042--3.385	T
26	26.217	37.97560	32.50046	-11.7586047	-6.2834602	-0.82422465	-0.81999469	-0.426973364	-2.042--3.385	T
27	34.433	37.15402	36.22850	-2.7210236	-1.7954963	-0.18933769	-0.18646028	-0.122007470	-2.042--3.385	T
28	28.567	32.59325	30.74341	-4.0262545	-2.1764066	-0.27965703	-0.27558968	-0.147891069	-2.042--3.385	T
29	50.500	50.75852	47.21904	-0.2585163	3.2809624	-0.01788515	-0.01760356	0.222947784	-2.042--3.385	T
30	20.950	27.62169	27.12233	-6.6716948	-6.1723350	-0.46599435	-0.46021957	-0.419422185	-2.042--3.385	T
31	85.583	101.79832	91.96552	-16.2153238	-6.3825237	-1.17891252	-1.18639577	-0.433704916	-2.042--3.385	T
32	32.383	34.94444	34.61913	-2.5614415	-2.2361311	-0.17883396	-0.17610553	-0.151949464	-2.042--3.385	T
33	170.250	160.38030	151.93989	9.8697022	18.3101073	0.73890152	0.73354928	1.244207457	-2.042--3.385	T
34	28.100	28.37949	26.59267	-0.2794856	1.5073328	-0.01944034	-0.01913429	0.102426197	-2.042--3.385	T
35	159.833	170.60663	163.71475	-10.7736271	-3.8817547	-0.81619003	-0.81183050	-0.263772791	-2.042--3.385	T

Now we just have to decide if this is studentized residual is large enough to deem the data point influential. To do that we rely on the fact that, in general, studentized residuals follow a t distribution with (n-k-2) degrees of freedom. That is, all we need to do is compare the studentized residuals to the t distribution with (n-k-2) degrees of freedom. If a data point's studentized residual is extreme-that is the data point is deemed influential.

As you can see, the studentized residual for the point 7 data point is which is not in the range of CI of T-test hence we can consider it as influential data points.

7	204.617	173.35458	157.06337	31.2624197	47.5536348	2.79819456	3.16897977	3.231362124	-2.042--3.385	F

We can use these tests to make our decision easy while trying to find right influential data point.

Ref:

T-value range for different DF.

```
t.values <- seq(-4,4,.1)


plot(x = t.values,y = dt(t.values,21), type = "l", lty = "dotted", ylim = c(0
,.4), xlab = "t", ylab = "f(t)")
lines(t.values,dt(t.values,3),lty = "dashed")
lines(t.values,dt(t.values,31),lty = "solid")
lines(t.values,dt(t.values,51),lty = "solid")
lines(t.values,dt(t.values,1),lty = "dashed")
```