PROBLEM STATEMENT :

Creation of a model that can predict the number of followers of a user on twitter, given a twitter handle.

IMPROVEMENTS:

As seen from the previous report, the data was not following any linear model and there was no correlation between the independent variable i.e FOLLOWERS and the dependent variables i.e FOLLOWING , STATUSES, LISTED ,DAYS etc.

Therefore , to improve upon the model the following things were done : -

1) New feature was analyzed – Total number of retweet of a user.
2) New custom features were analyzed – Number of tweet per day , retweet per status etc
3) New non linear Logarithmic model was user

NEW MODEL :

Multivariate Non Linear Logarithmic Regression –

Method –

Create new variables from the data. The new variables are nonlinear functions of the variables in our data. If we construct your new variables properly, the curved function of our original variables can be expressed as a linear function of your new variables.

Log() function was applied on the variables both dependent and the independent variables resulting in a variable that are non linear in nature. Then regression equation was applied using the new logarithmic variables.

$$\mathbf{Log(Y) = b_0 + b_1\, Log(X_1) + b_2\, Log(X_2) + b_3\, Log(X_3)}$$

Y is the value of the Dependent variable (Y), what is being predicted or explained
b_0 is the Constant or intercept
b_1 is the Slope (Beta coefficient) for Log(X_1)
$\mathbf{X_1}$ First independent variable that is explaining the variance in Y
b_2 is the Slope (Beta coefficient) for Log(X_2)
$\mathbf{X_2}$ Second independent variable that is explaining the variance in Y
b_3 is the Slope (Beta coefficient) for Log (X_3)
$\mathbf{X_3}$ Third independent variable that is explaining the variance in Y

TECHNOLOGIES USED:

1) Python – For data extraction
2) R – For applying the model.

Note - Due to Library installation issues I was not able to continue with python and therefore I had to use R for applying the model. You can use anything you are comfortable with.

DATA EXTRACTION:

1) Twython (open source library in python) - https://twython.readthedocs.org/en/latest/ for extracting Retweet Data
2) Twitter API
3) Tweepy (open source library in python) - https://github.com/tweepy/tweepy

CODE FOR DATA EXTRACTION:

Tweet.py – Same as the previous report

TwitterJSON.py- Same as the previous report

Retweetextract.py –

In this script, we use the Twitter IDs that were extracted in Tweet.py script to fetch all the Tweets tweeted by a given user. All the tweets were analyzed to find how many times were they retweeted. The tweets that the user retweeted were ignored.

```
try:
#file that contained the stream data extracted in tweey.py
  with open('E:/IIIT delhi/Twitter_file/twitterData.csv', "rb") as csvfile:
    datareader = csv.reader(csvfile)
    for data in datareader:
      count = 0
      count_all =
 # we can extract 200 tweets per page. So variable 'iterate' was used for count the number of total number of pages – (Total number of tweet)/200
      iterate = (int(data[7])/200) + 1
        try:
          for i in range(iterate):
            timeline = twitter.get_user_timeline(screen_name=str(data[2]),count=200,page = i+1)
# We ignore the tweet that starts with 'RT' as it was retweeted by the user
            for tweet in timeline:
              if len(tweet['text']) > 2:
```

```
                    if tweet['text'][0] != 'R':
                        if tweet['text'][1] != 'T':
#We only count the retweets that were done for a users own status
                            count = tweet['retweet_count'] + count
                        count_all = tweet['retweet_count'] + count_all
                    else:
                        count = tweet['retweet_count'] + count
                        count_all = tweet['retweet_count'] + count_all
                saveFile = open ('E:/IIIT delhi/Twitter_file/twitterData_retweet.csv','a')
                saveFile.write(str(data[0]) + ',' +str(data[1]) + ',' + str(data[2]) + ',' + str(data[3]) + ',' +
str(data[4]) + ',' + str(data[5]) + ',' + str(data[6]) + ',' + str(data[7]) + ',' + str(data[8])+ ',' + str(count_all)+
',' + str(count))
                saveFile.write('\n')
                saveFile.close()
                #data_save = [str(data[0]) ,
str(data[1]),str(data[2]),str(data[3]),str(data[4]),str(data[5]),str(data[6]),str(data[7]),str(data[8]),str(coun
t_all),str(count)]
            except TwythonError as e:
                print e
#As twitter has a request limit , therefore we get an - Twitter API returned a 429 (Too Many Requests),
Rate limit exceeded error. We can only resume after 5mins.
                time.sleep(310)



except TwythonError as e:
    print e
    #time.sleep(310)
```

New features considered:

1. retweet_count
2. Retweet per status
3. Status per day

List of all the data extracted :

1. **handle** - twitter username | string
2. **name** - full name of the twitter user | string
3. **Days** - number of days the user has existed on twitter | number
4. **Statuses**- number of tweets this user has created (includes retweets) | number
5. **following** - number of other twitter users, this user is following | number
6. **favorites** - number of tweets the user has favorited | number
7. **lists** - number of public lists this user has been added to | number
8. **followers** - number of other users following this user | number

9. **retweet** – Total number of retweets for all the users tweet
10. **Retweet per statuses(RpS)** – number of retweets per tweet
11. **Statuses per days(SpD)** - number of statuses per days

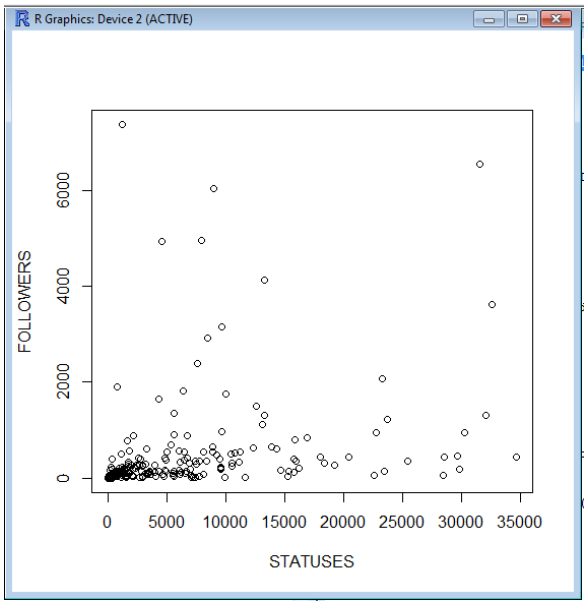
ID	FOLLOWERS	FOLLOWING	LISTED	FAVOURIT	STATUSES	DAYS	countall	retweet	RpS	SpD
2917930622	6	73	0	0	6655	52	1	1	0.00015	127.9808
153753502	241	237	12	20599	18559	1690	300052	34	0.001832	10.98166
2942240530	443	321	0	0	774	29	2	2	0.002584	26.68966
472979726	1878	538	11	4044	12973	1096	395157	75	0.005781	11.83668
194583855	197	248	1	190	586	1583	69293	134	0.228669	0.370183
2850036578	35	584	2	100	634	106	33614	0	0	5.981132
196270821	125	114	1	50	4612	1579	175746	165	0.035776	2.920836
2545851438	63	49	4	1735	11214	234	14319	155	0.013822	47.92308
2956467215	207	102	0	0	926	22	5	5	0.0054	42.09091
362481565	319	2001	5	24	16233	1247	42	15	0.000924	13.01764
2414086476	551	128	15	0	597	303	33670	723	1.211055	1.970297
2883120413	154	58	7	0	28653	67	8	8	0.000279	427.6567
2696111005	966	2001	10	857	4701	177	427187	322	0.068496	26.55932
2868759687	55	66	2	86	441	76	7600	97	0.219955	5.802632
2199500008	122	498	0	691	4068	422	2722287	82	0.020157	9.63981
181119311	30	95	0	2	222	1617	4133	6	0.027027	0.137291
1656630072	11	79	0	38	234	533	156393	0	0	0.439024
2914847320	25	26	0	0	14876	44	1	1	6.72E-05	338.0909
90136351	167	1797	1	8	1535	1896	44334	1	0.000651	0.809599
2786134751	475	1255	10	1607	2193	120	238210	182	0.082991	18.275

Data Analyses :

Now we analyze all the data features mentioned in the list above and find the relevant features. To find the relevant features we use correlation and scatter plot –

FOLLOWERS - STATUSES

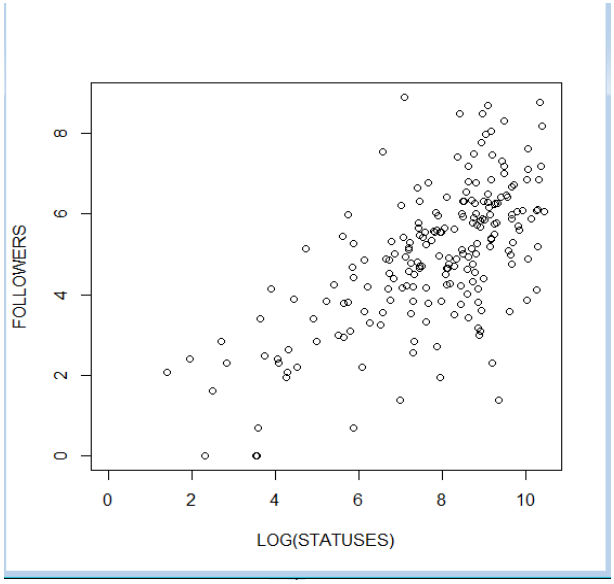
> cor(FOLLOWING,FOLLOWERS)
[1] 0.5027552



There seems to be little correlation between the 2 features.

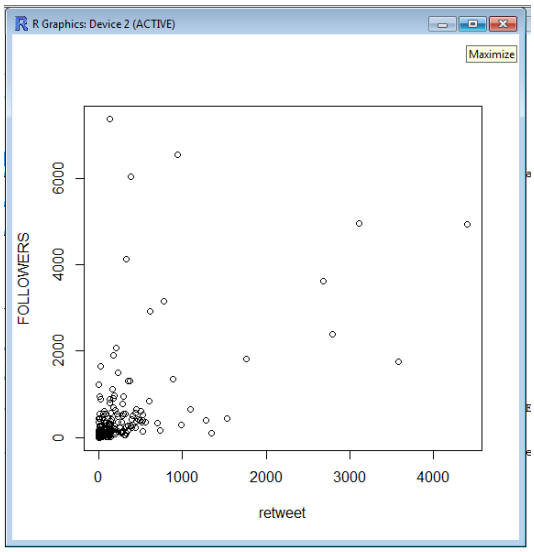
Now using Log(FOLLOWERS) – LOG(STATUSES) – It can be seen from the Scatter plot and the correlation value that there seems to be a relation between LOG(FOLLOWERS) and LOG(STATUSES)

> cor(STATUSES,FOLLOWERS)
[1] 0.6736216

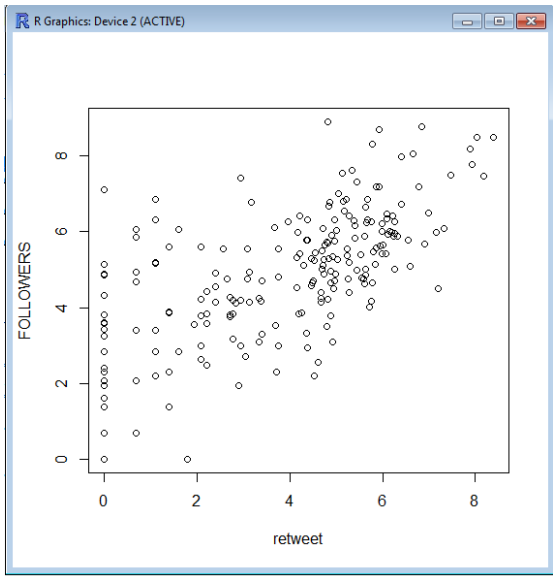


Similarly for other features :

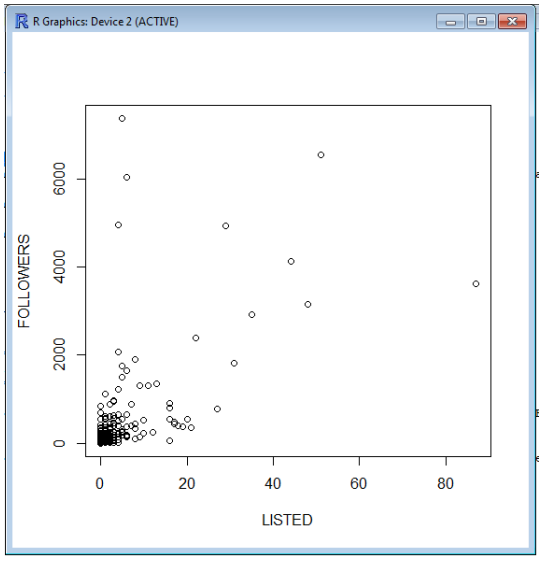
cor(retweet,FOLLOWERS)
[1] 0.5304777



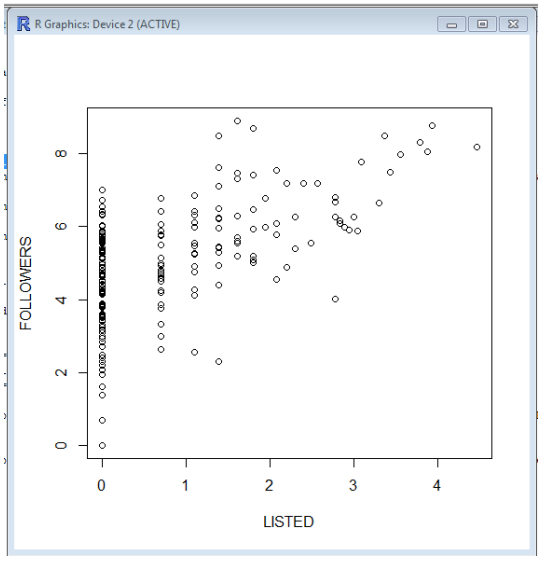

```
cor(log(retweet),log(FOLLOWERS))
[1] 0.6689346
```



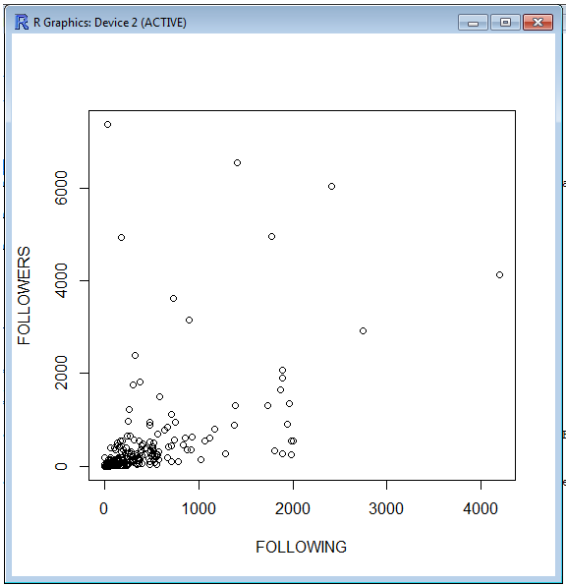

```
cor(LISTED,FOLLOWERS)
[1] 0.3931089
```



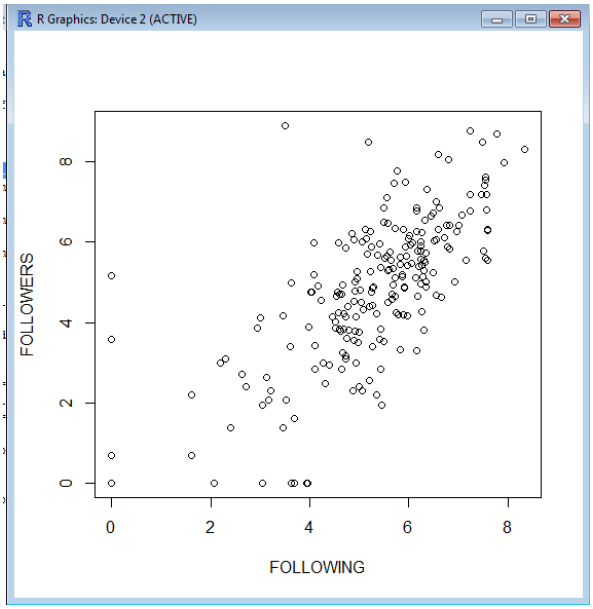
cor(log(LISTED),log(FOLLOWERS))
[1] 0.591715



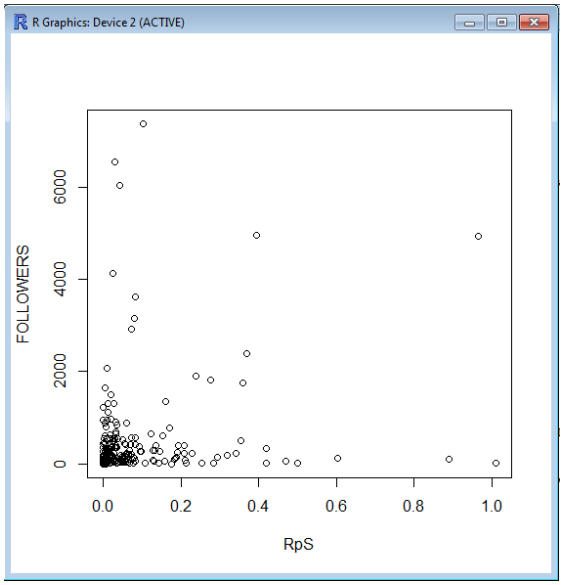
cor(FOLLOWING,FOLLOWERS)
[1] 0.5027552



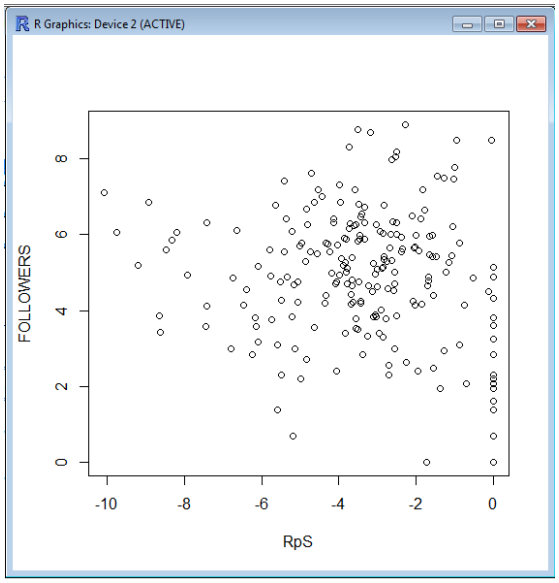
cor(log(FOLLOWING),log(FOLLOWERS))
[1] 0.6924032



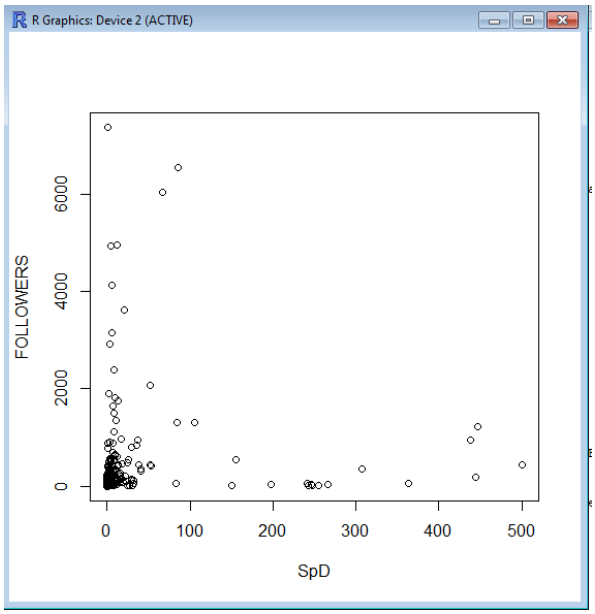
cor(RpS,FOLLOWERS)
[1] 0.1767746



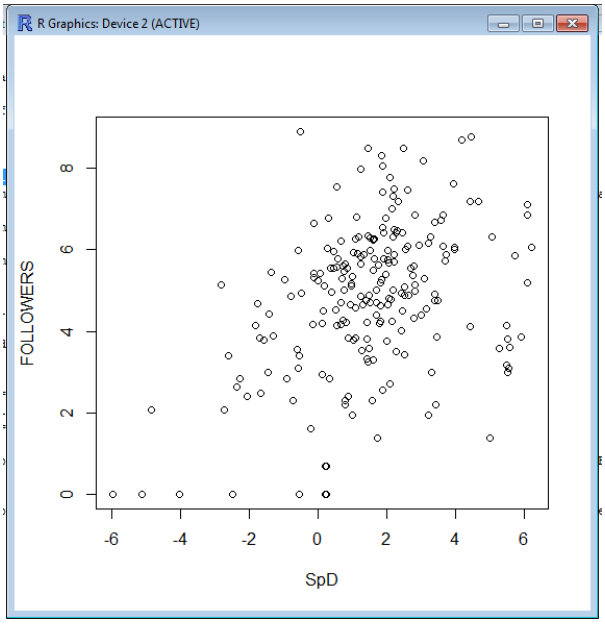
cor(log(RpS),log(FOLLOWERS))
[1] -0.1997462



cor(SpD,FOLLOWERS)
[1] 0.02598865



cor(log(SpD),log(FOLLOWERS))
[1] 0.3854226

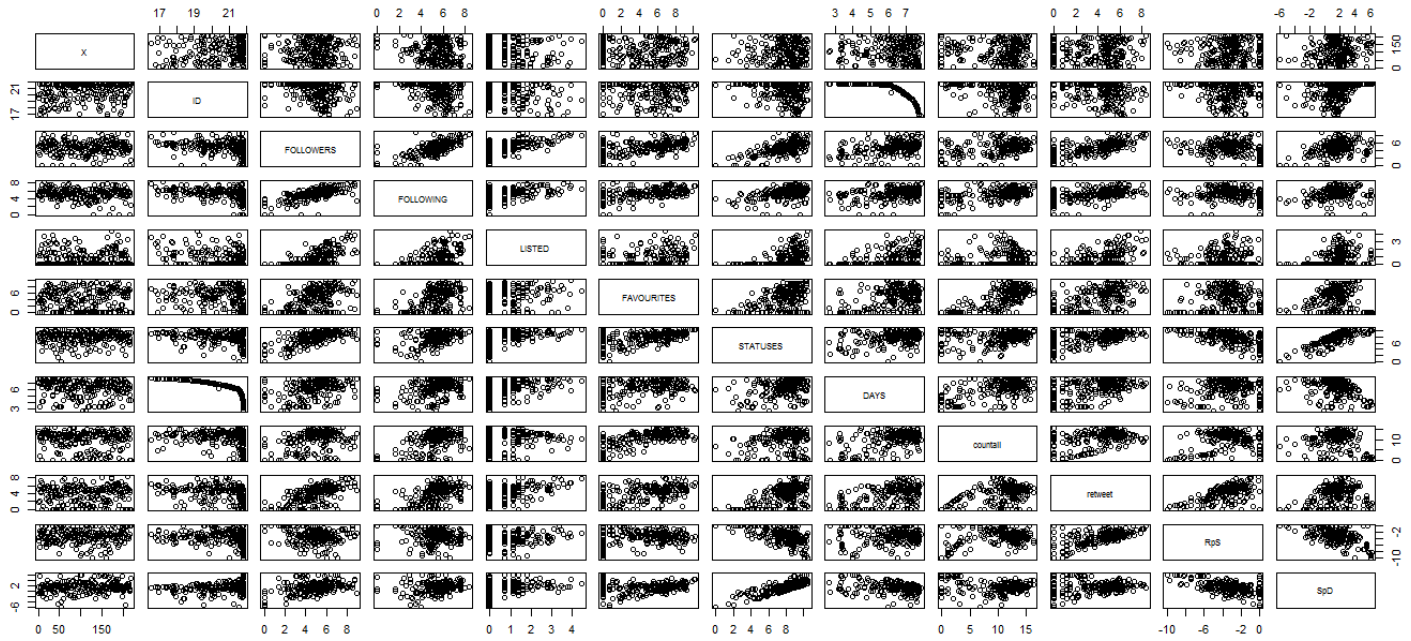


FULL Correlation table:

	X	ID	FOLLOWERS	FOLLOWING	LISTED
X	1.00000000	0.15569860	0.02860711	-0.14153907	-0.03070289
ID	0.15569860	1.00000000	-0.32526193	-0.33455199	-0.33731056
FOLLOWERS	0.02860711	-0.32526193	1.00000000	0.69240323	0.59171497
FOLLOWING	-0.14153907	-0.33455199	0.69240323	1.00000000	0.39817817
LISTED	-0.03070289	-0.33731056	0.59171497	0.39817817	1.00000000
FAVOURITES	0.04168759	-0.26495238	0.48490156	0.47311821	0.28725815
STATUSES	0.07978311	-0.21444878	0.67362160	0.41552692	0.45764827
DAYS	-0.08835435	-0.79779439	0.40560685	0.36203989	0.33167989
countall	-0.06946556	-0.34412537	0.37971919	0.49546342	0.13699750
retweet	0.05540159	-0.33536547	0.66893460	0.49083791	0.48661055
RpS	0.04578158	-0.01945093	-0.19974619	-0.05720585	-0.07595872
SpD	0.13387631	0.30983181	0.38542263	0.16540611	0.22555085

	FAVOURITES	STATUSES	DAYS	countall	retweet
X	0.04168759	0.07978311	-0.08835435	-0.06946556	0.05540159
ID	-0.26495238	-0.21444878	-0.79779439	-0.34412537	-0.33536547
FOLLOWERS	0.48490156	0.67362160	0.40560685	0.37971919	0.66893460
FOLLOWING	0.47311821	0.41552692	0.36203989	0.49546342	0.49083791
LISTED	0.28725815	0.45764827	0.33167989	0.13699750	0.48661055
FAVOURITES	1.00000000	0.38323520	0.47002915	0.74275742	0.54995548
STATUSES	0.38323520	1.00000000	0.27586310	0.24235100	0.51975536
DAYS	0.47002915	0.27586310	1.00000000	0.54307496	0.48415085
countall	0.74275742	0.24235100	0.54307496	1.00000000	0.51560264
retweet	0.54995548	0.51975536	0.48415085	0.51560264	1.00000000
RpS	0.01264703	-0.52402088	0.06454525	0.10469682	0.18573526
SpD	0.06449919	0.78321150	-0.38157021	-0.11823288	0.18664980

	RpS	SpD
X	0.04578158	0.13387631
ID	-0.01945093	0.30983181
FOLLOWERS	-0.19974619	0.38542263
FOLLOWING	-0.05720585	0.16540611
LISTED	-0.07595872	0.22555085
FAVOURITES	0.01264703	0.06449919
STATUSES	-0.52402088	0.78321150
DAYS	0.06454525	-0.38157021
countall	0.10469682	-0.11823288
retweet	0.18573526	0.18664980
RpS	1.00000000	-0.54567875



Data analyses Inference :

As can be seen from the scatter plots and the correlation table we find –

1 . There is a better correlation between Log(FOLLOWERS) – log(independent variable) than just normal FOLLOWERS – Independent variable

2. Custom variables - RpS and SpD doesn't correlate well with FOLLOWERS even in non linear Logarithmic form.

The following features were selected based on correlation value and scatter plot

1. Retweet
2. LISTED
3. STATUSES
4. FOLLOWINGS

Moreover, The erroneous data, missing data was removed from the file

The data was split into 2 parts –

1) Training Data - Consisting of 265 unique user
2) Testing Data - Consisting of 53 rows

Method Followed :

1) A non linear logarithmic model is created using the dependent variables – retweet , FOLLOWING , LISTED, STATUSES , and then the model is analyzed using R-Squared , P-VALUE for every variable . Those variables are removed which do not provide good P-VALUE i.e <0.05 . After removing the variables , A new model is created using the remaining variables and evaluated accordingly.
2) Correlation values between the variable provide a good evidence of the relationship between the FOLLOWERS and the dependent variables. Based on the value of correlation coefficient, variable are added into the model and analyzed using R-Square and P-Value.

STEPS -

Loading data into an object

```
> datavarTest <- read.csv('E:/IIIT delhi/Twitter_file/TwitterFinalTrain.csv')
>st = data.frame(datavar)
```

Lm() function was used for applying to fit a multiple linear regression model.

1)

MODEL1: Using all the dependent variable.

```
model1 = lm(formula = FOLLOWERS ~ retweet + FOLLOWING + STATUSES  + LISTED, data =
datavar_log)

>model1
```

Below are the values of the coefficient - b_0 , b_1 , b_3 , b_4

```
Call:

lm(formula = FOLLOWERS ~ retweet + FOLLOWING + STATUSES + LISTED,
   data = datavarTrainlog)

Coefficients:
(Intercept)     retweet   FOLLOWING    STATUSES      LISTED
   -0.8258      0.2233      0.4747      0.2695      0.3013
```

> summary(model1) – will show the analysis of the model providing various parameters to access the model.

```
> summary(model1)

Call:
lm(formula = FOLLOWERS ~ retweet + FOLLOWING + STATUSES + LISTED,
    data = datavarTrainlog)

Residuals:
    Min      1Q  Median      3Q     Max
-2.9092 -0.5061 -0.0020  0.5012  4.5981

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.82584    0.29133  -2.835  0.00494 **
retweet      0.22332    0.03128   7.139 9.03e-12 ***
FOLLOWING    0.47469    0.04559  10.412  < 2e-16 ***
STATUSES     0.26951    0.03548   7.597 5.22e-13 ***
LISTED       0.30127    0.06554   4.597 6.65e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9201 on 265 degrees of freedom
Multiple R-squared:  0.7384,    Adjusted R-squared:  0.7345
F-statistic:   187 on 4 and 265 DF,  p-value: < 2.2e-16
```

ANALYSIS :

Now we analyse this model using the R-squared , P value etc.

R-Squared value - 0.7384 i.e 73.84% of the points can be explained by this model.

As we are considering a confidence interval of 95%, the P-Value of the dependent variables should be <0.05 and P-Value of all the variables is < 0.05.

MODEL1 INFERENCE:

Therefore , Model1 accepted

PREDICTION :

pred = coef(model1)[1] + coef(model1)[2]*retweet+ coef(model1)[3]*FOLLOWING + coef(model1)[4]*STATUSES + coef(model1)[5]*LISTED

As we are using LOG(Y) , we need to exponentiate the output variable pred to get Y.
As Exp(log(Y)) = Y

The below table presents the output of the model. This gives the predicted value , predicted value within 10%,15%,20% range along with the absolute error for all the test data.

Exp(pred)

ID	Predicted	Actual	10%	-10%	15%	-15%	20%	-20%	Absolute error = |Predicted – actual|
306720597	40.75679	35	44.83247	36.68111	46.87031	34.64328	48.90815	32.60543542	5.75679429
469305978	242.8487	194	267.1336	218.5639	279.2761	206.4214	291.4185	194.2789914	48.84873931
2943623928	79.63326	20	87.59658	71.66993	91.57825	67.68827	95.55991	63.70660567	59.63325708
23409507	328.711	226	361.5821	295.8399	378.0176	279.4043	394.4532	262.9687915	102.7109894
573662950	98.18802	206	108.0068	88.36922	112.9162	83.45982	117.8256	78.55041804	107.8119775
36154505	402.1824	395	442.4006	361.9642	462.5098	341.855	482.6189	321.7459156	7.182394468
51968791	285.547	189	314.1017	256.9923	328.3791	242.715	342.6564	228.4376107	96.54701339
2892201369	6.576087	1	7.233696	5.918478	7.5625	5.589674	7.891305	5.260869696	5.57608712
1659705086	1611.902	1342	1773.093	1450.712	1853.688	1370.117	1934.283	1289.521845	269.9023057
2907155647	132.9402	131	146.2342	119.6462	152.8812	112.9991	159.5282	106.3521389	1.940173692
216372416	442.0466	181	486.2513	397.842	508.3536	375.7396	530.456	353.6373022	261.0466278
60201182	55.85514	109	61.44065	50.26963	64.23341	47.47687	67.02617	44.68411202	53.14485996
29724534	3369.051	4133	3705.956	3032.146	3874.409	2863.693	4042.861	2695.2407	763.949127
706705454	286.755	554	315.4305	258.0795	329.7682	243.7417	344.106	229.4039994	267.2450009
2715518479	106.4164	82	117.0581	95.7748	122.3789	90.45398	127.6997	85.13315608	24.41644512
554874349	133.1021	129	146.4123	119.7919	153.0674	113.1368	159.7225	106.481673	4.102091253
19554936	1230.731	534	1353.805	1107.658	1415.341	1046.122	1476.878	984.5851374	696.7314217
201923649	540.2652	450	594.2917	486.2387	621.305	459.2254	648.3183	432.2121721	90.26521497
1391524304	103.066	43	113.3726	92.75936	118.5259	87.60606	123.6792	82.45276701	60.06595875
91794873	25.32138	173	27.85352	22.78925	29.11959	21.52318	30.38566	20.25710796	147.678615
2169089640	1182.291	2073	1300.52	1064.062	1359.635	1004.947	1418.749	945.8327076	890.7091146
713992237	205.6322	288	226.1954	185.069	236.4771	174.7874	246.7587	164.5057779	82.36777757
701836814	531.8603	879	585.0463	478.6742	611.6393	452.0812	638.2323	425.4882009	347.1397491
48054700	653.1703	373	718.4873	587.8533	751.1458	555.1947	783.8043	522.5362232	280.1702789
1576796352	60.39215	68	66.43137	54.35294	69.45098	51.33333	72.47058	48.31372193	7.607847578
342393846	339.1582	272	373.0741	305.2424	390.032	288.2845	406.9899	271.3265994	67.15824928
55472318	528.3135	357	581.1448	475.4821	607.5605	449.0664	633.9762	422.6507681	171.3134601
471472576	114.9818	68	126.48	103.4836	132.2291	97.73454	137.9782	91.98545223	46.98181529
262532488	16.88233	17	18.57056	15.1941	19.41468	14.34998	20.25879	13.50586234	0.117672069
1642334412	1152.659	804	1267.925	1037.393	1325.558	979.7605	1383.191	922.1274962	348.6593704
732427747	429.9789	328	472.9767	386.981	494.4757	365.482	515.9746	343.9830807	101.978851
189865165	234.1753	258	257.5928	210.7577	269.3015	199.049	281.0103	187.3402023	23.82474708
2785959955	11.40516	1	12.54568	10.26465	13.11594	9.69439	13.6862	9.124131684	10.4051646
2741099997	109.6106	119	120.5717	98.64957	126.0522	93.16904	131.5328	87.68850863	9.389364203

234389725	1853.555	3145	2038.911	1668.2	2131.589	1575.522	2224.266	1482.844294	1291.444632
103570566	653.9898	768	719.3887	588.5908	752.0882	555.8913	784.7877	523.1918179	114.0102275
2953995122	14.02189	11	15.42408	12.6197	16.12517	11.91861	16.82627	11.21751164	3.021889548
59344219	341.594	301	375.7534	307.4346	392.8331	290.3549	409.9128	273.2751791	40.59397382
52644084	135.2885	139	148.8174	121.7597	155.5818	114.9952	162.3462	108.23081	3.711487524
376510560	412.1576	613	453.3734	370.9419	473.9813	350.334	494.5892	329.726115	200.8423563
218727806	490.3743	438	539.4117	441.3369	563.9305	416.8182	588.4492	392.2994465	52.37430835
2929192061	11.98116	17	13.17928	10.78305	13.77834	10.18399	14.37739	9.58492998	5.018837524
2475742249	38.91038	20	42.80142	35.01934	44.74694	33.07382	46.69245	31.12830319	18.91037898
317889040	94.52074	67	103.9728	85.06866	108.6988	80.34263	113.4249	75.61659039	27.52073801
1884326047	316.0163	55	347.6179	284.4146	363.4187	268.6138	379.2195	252.8130154	261.0162692
35328164	376.3635	246	413.9999	338.7272	432.818	319.909	451.6362	301.0908015	130.3635019
1729747776	3.961012	8	4.357113	3.56491	4.555163	3.36686	4.753214	3.168809239	4.038988454
55181323	183.4286	10	201.7714	165.0857	210.9429	155.9143	220.1143	146.742867	173.4285838
2950665182	34.8005	69	38.28055	31.32045	40.02058	29.58043	41.7606	27.84040158	34.19949806
2158210285	31.4728	31	34.62008	28.32552	36.19372	26.75188	37.76735	25.17823661	0.472795773
593512124	84.75127	132	93.2264	76.27615	97.46397	72.03858	101.7015	67.80101951	47.24872567
2943504482	96.25958	22	105.8855	86.63363	110.6985	81.82065	115.5115	77.00766718	74.25958398
2272823593	2.856972	1	3.142669	2.571275	3.285518	2.428426	3.428367	2.28557776	1.8569722

CONCLUSION :

1. There is a significant improvement in prediction when using we use a non linear model rather than a linear model
2. The model was able to predict the number of FOLLOWERS with some error but the errors were not extreme. The Absolute error of the prediction was not very large.

REFERENCES

[1] http://rstudio-pubs-static.s3.amazonaws.com/10578_319e4083c11341cfb8a7d79b65665f6f.html

[2] Video explaining regression https://www.youtube.com/user/BCFoltz

[3] A Longitudinal Study of Follow Predictors on Twitter - http://comp.social.gatech.edu/papers/follow_chi13_final.pdf