MID-TERM

Name: Ajinkya Rasam

RUID: 172002767

E-mail: rasam.ajinkya23@rutgers.edu

PART 1(UNIX Programming)

Answer 01)
```
tail -n+2 ChicagoCrimes.csv| cut -d',' -f6|uniq|wc -l

-→30
```

Answer 02)
```
grep -w "ATTEMPT THEFT" ChicagoCrimes.csv|cut -d"," -f1-11|grep -wi
"True"|wc -l

→ 7
```

Answer 03)
Reference:
http://unix.stackexchange.com/questions/104525/sort-based-on-the-third-column
http://stackoverflow.com/questions/17842903/delete-specific-rows-based-on-specific-word-in-column

```
grep -v "CRIM SEXUAL ASSAULT" ChicagoCrimes.csv > output.csv
grep -wi "ASSAULT" output.csv| sort -nk 12 >assault.csv
```

Answer 04)
LOWEST CRIME
```
cut -d"," -f12  ChicagoCrimes.csv |sort -n|uniq -c|tail -n+4|head -
22|sort|head -3
→290 020
 528 024
 544 017
```

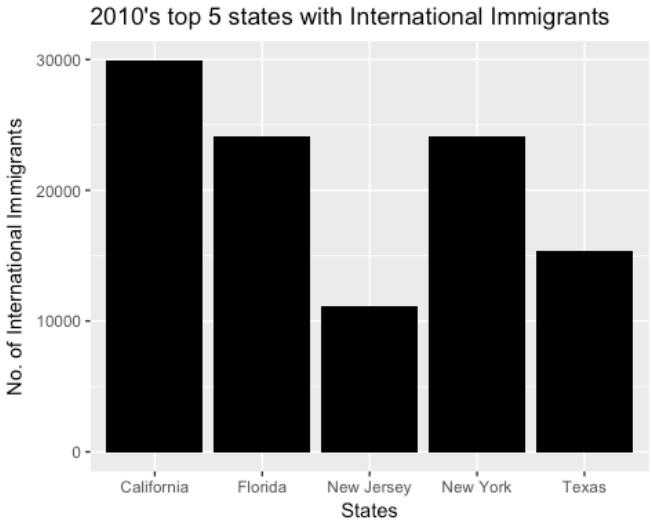
HIGHEST CRIME
```
cut -d"," -f12  ChicagoCrimes.csv |sort -n|uniq -c|tail -n+4|head --
22|sort|tail -3
1119 006
1183 008
1305 011
```

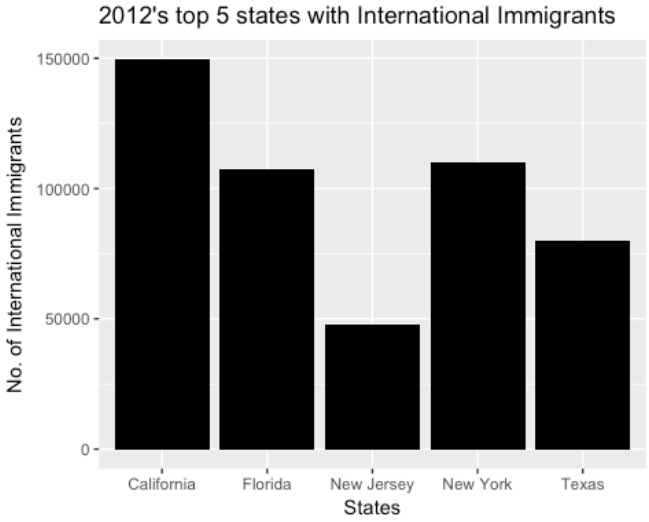
PART 2(R Programming)

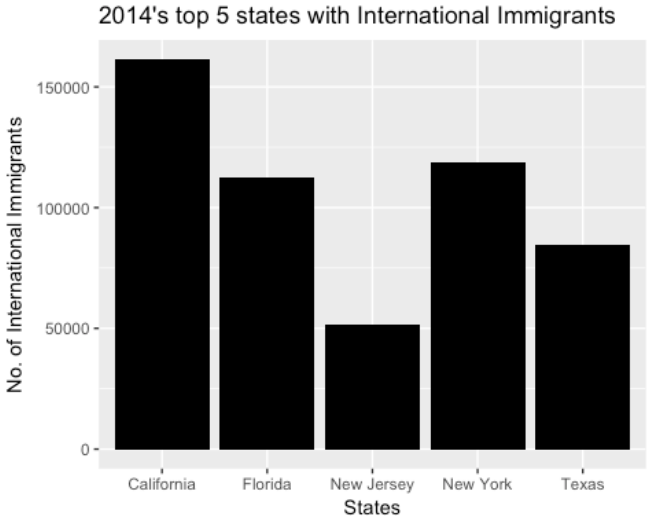
Answer 1)

```r
popDF=read.table('/Users/rasam/Google Drive/Spring17/PSWD/PSW_mid_term/Population/Population.csv',header=T,sep=',')
#--------------ANSWER_01----------------------------------------
#Ref:http://stackoverflow.com/questions/3445590/how-to-extract-a-subset-of-a-data-frame-based-on-a-condition-involving-a-field
states <- subset(popDF,(popDF$SUMLEV == '40'))
#https://www.r-bloggers.com/r-sorting-a-data-frame-by-the-contents-of-a-column/
#Year 2010
max_2010 =states[order(-states$INTERNATIONALMIG2010)[1:5],]
library(ggplot2)
#Ref:http://docs.ggplot2.org/0.9.3.1/geom_bar.html
ggplot(max_2010, aes(x = max_2010$NAME, y = max_2010$INTERNATIONALMIG2010)) +
  geom_bar(stat = "identity",fill='black') + xlab("States") +
  ylab("No. of International Immigrants") +
  ggtitle("2010's top 5 states with International Immigrants")
#Year 2012
max_2012 =states[order(-states$INTERNATIONALMIG2012)[1:5],]
ggplot(max_2012, aes(x = max_2012$NAME, y = max_2012$INTERNATIONALMIG2012)) +
  geom_bar(stat = "identity",fill='black') + xlab("States") +
  ylab("No. of International Immigrants") +
  ggtitle("2012's top 5 states with International Immigrants")
#Year 2014
max_2014 =states[order(-states$INTERNATIONALMIG2014)[1:5],]
ggplot(max_2014, aes(x = max_2014$NAME, y = max_2014$INTERNATIONALMIG2014)) +
  geom_bar(stat = "identity",fill='black') + xlab("States") +
  ylab("No. of International Immigrants") +
  ggtitle("2014's top 5 states with International Immigrants")

```


2010's top 5 states with International Immigrants
No. of International Immigrants
30000
20000
10000
0
California
Florida
New Jersey
New York
Texas
States


2012's top 5 states with International Immigrants
No. of International Immigrants
150000
100000
50000
0
California
Florida
New Jersey
New York
Texas
States


2014's top 5 states with International Immigrants
No. of International Immigrants
150000
100000
50000
0
California
Florida
New Jersey
New York
Texas
States

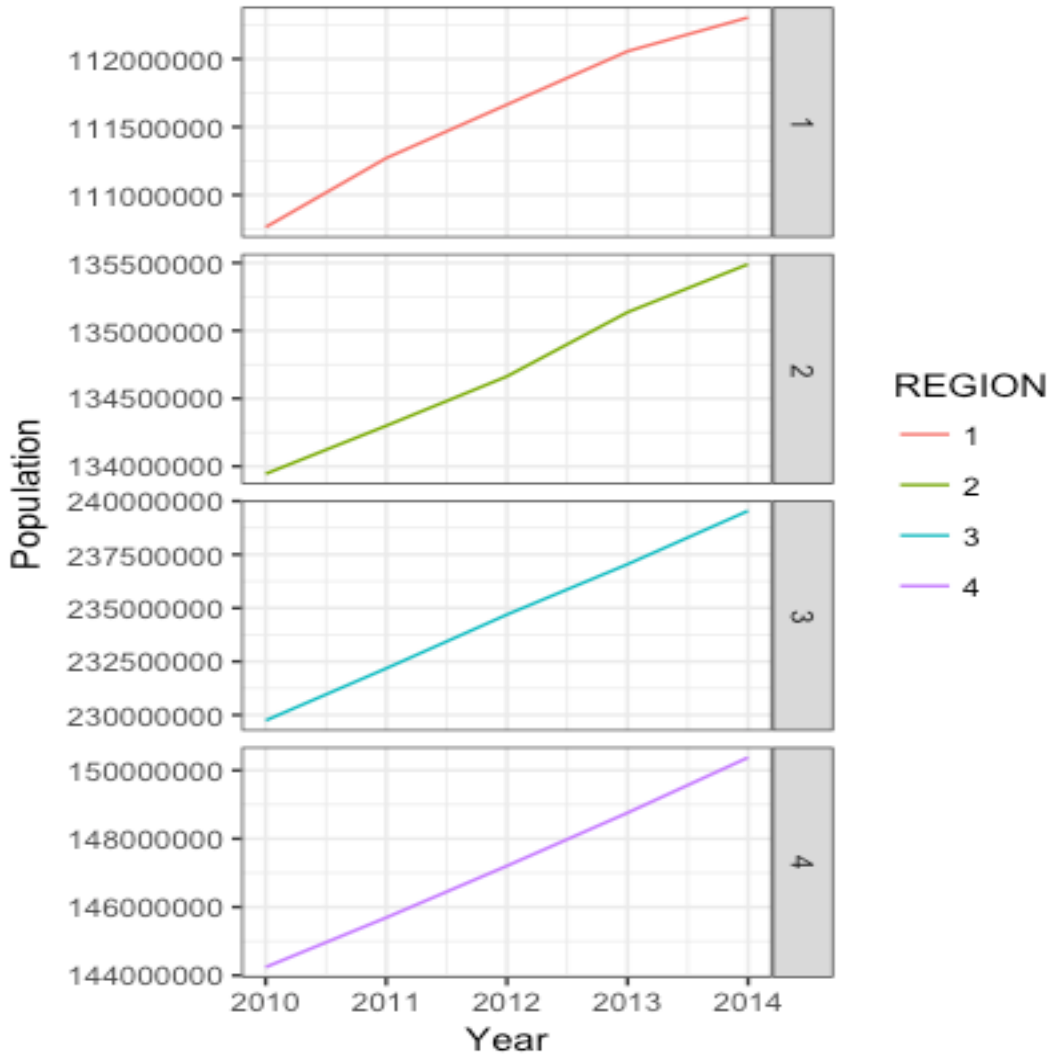
Answer 02)

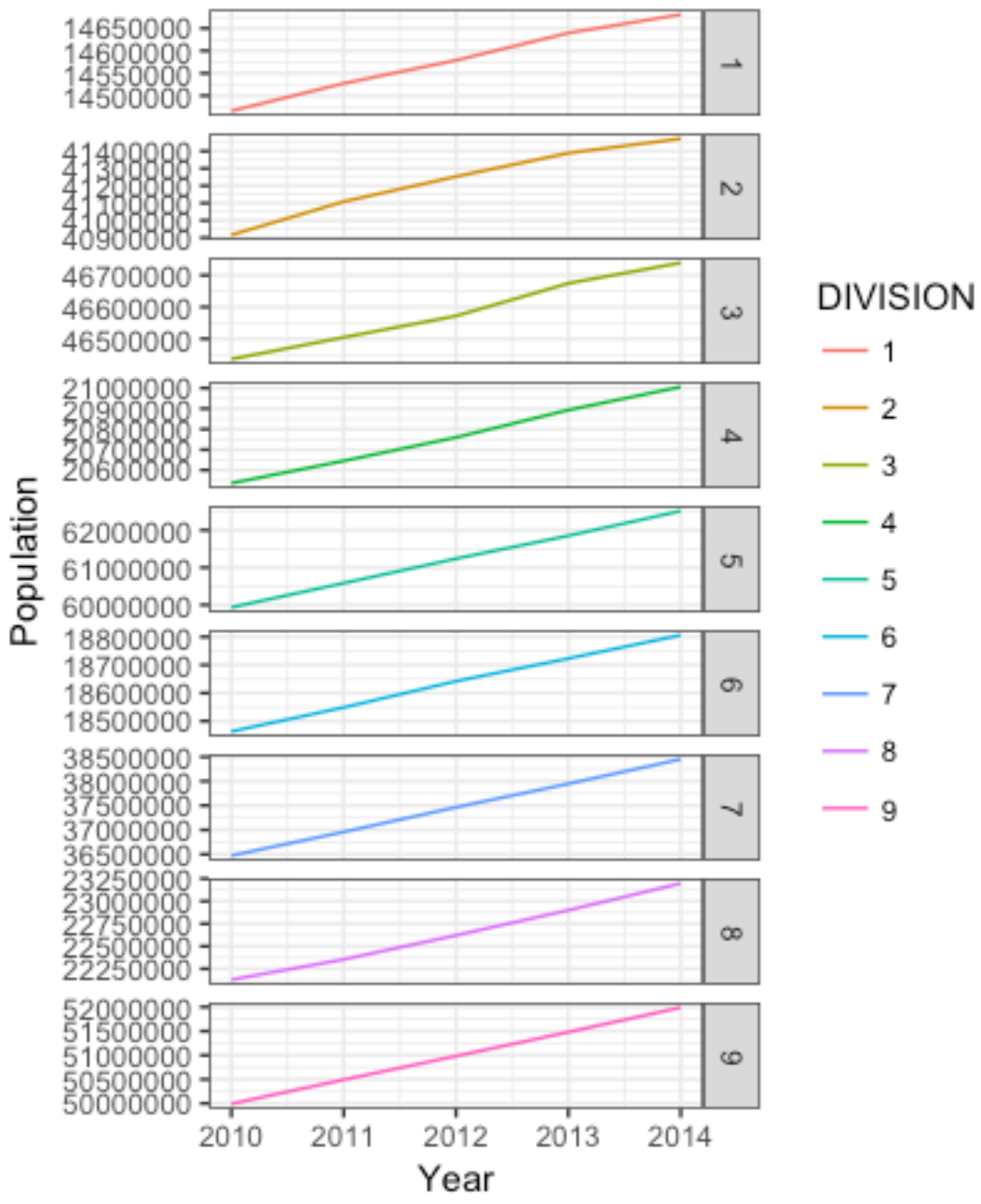
```r
#-------------ANSWER_02---------------------------------------
name(popDF)
#For Region
#Ref:http://stackoverflow.com/questions/10085806/extracting-specific-columns-from-a-data-frame
region_DF = popDF[,c("REGION","POPESTIMATE2010","POPESTIMATE2011","POPESTIMATE2012","POPESTIMATE2013","POPESTIMATE2014")]
#Removing rows with irrelevant data like region '0' and region 'X'
region_DF=subset(region_DF,(region_DF$REGION != 'X'& region_DF$REGION != 0))
#Ref:http://stackoverflow.com/questions/18799901/data-frame-group-by-column
regional_pop = aggregate(. ~ REGION, region_DF, sum)
library(reshape2)
traff2 <- melt(regional_pop,id=c("REGION"),variable.name = "Year")
#Remove the X in the Year column and convert it to number
traff2$Year <- as.numeric(gsub(pattern="POPESTIMATE",replacement = "",x = as.character(traff2$Year)))
options(scipen=10)
ggplot(traff2, aes(x = Year, y = value, color = REGION))+
  facet_grid(facets = REGION~., scales = "free_y")+
  geom_line()+theme_bw()+ylab('Population')

#For Division
div_DF=popDF[,c("DIVISION","POPESTIMATE2010","POPESTIMATE2011","POPESTIMATE2012",
                "POPESTIMATE2013","POPESTIMATE2014")]
#Removing rows with irrelevant data like region '0' and region 'X'
div_DF=subset(div_DF,(div_DF$DIVISION!=0 & div_DF$DIVISION!='X'))
#Group population by Division
div_pop=aggregate(. ~ DIVISION, div_DF, sum)
traff22 <- melt(div_pop,id=c("DIVISION"),variable.name = "Year")

#Remove the X in the Year column and convert it to number
traff22$Year <- as.numeric(gsub(pattern="POPESTIMATE",replacement = "",x = as.character(traff22$Year)))
options(scipen=10)
ggplot(traff22, aes(x = Year, y = value, color = DIVISION))+
  facet_grid(facets = DIVISION~., scales = "free_y")+
  geom_line()+theme_bw()+ylab('Population')
#Ref:http://stackoverflow.com/questions/27382649/a-line-graph-for-each-row

```


Population
112000000
111500000
111000000
135500000
135000000
134500000
134000000
240000000
237500000
235000000
232500000
230000000
150000000
148000000
146000000
144000000
1
2
3
4
2010
2011
2012
2013
2014
Year
REGION
1
2
3
4


Population
14650000
14600000
14550000
14500000
1
41400000
41300000
41200000
41100000
41000000
40900000
2
46700000
46600000
46500000
3
21000000
20900000
20800000
20700000
20600000
4
62000000
61000000
60000000
5
18800000
18700000
18600000
18500000
6
38500000
38000000
37500000
37000000
36500000
7
23250000
23000000
22750000
22500000
22250000
8
52000000
51500000
51000000
50500000
50000000
9
2010
2011
2012
2013
2014
Year
DIVISION
1
2
3
4
5
6
7
8
9

Answer 03)

```r
#-------------ANSWER_03-----------------------------------------
x = popDF[,c("DIVISION","NPOPCHG_2012","NPOPCHG_2014")]
x=subset(x,(x$DIVISION!=0 & x$DIVISION!='X'))
#Group population by Division
x_div_pop=aggregate(. ~ DIVISION, x, sum)

#x_div_pop['DIVISION'][which.max(x_div_pop$NPOPCHG_2012)]
#Ref:http://gis.stackexchange.com/questions/97310/return-column-number-of-min-value-in-dataframe
max_pop_rate_2012 = x_div_pop$DIVISION[x_div_pop$NPOPCHG_2012 == max(x_div_pop$NPOPCHG_2012)]
max_pop_rate_2012 = as.character(max_pop_rate_2012)
cat("Division that show the highest increasing rate of population between 2011 and 2012 is: ", max_pop_rate_2012)

max_pop_rate_2014 = x_div_pop$DIVISION[x_div_pop$NPOPCHG_2014 == max(x_div_pop$NPOPCHG_2014)]
max_pop_rate_2014 = as.character(max_pop_rate_2014)
cat("Division that show the highest increasing rate of population between 2013 and 2014 is: ", max_pop_rate_2014)


```

Division that show the highest increasing rate of population between 2011 and 2012 is: 5
Division that show the highest increasing rate of population between 2013 and 2014 is: 5

PART 3

1. The data set is of online news and its present in .csv file. The problem is of regression type. The target variable is ‘shares’ so let’s try to find out if there is a attribute who has got a strong relationship with our target variable.

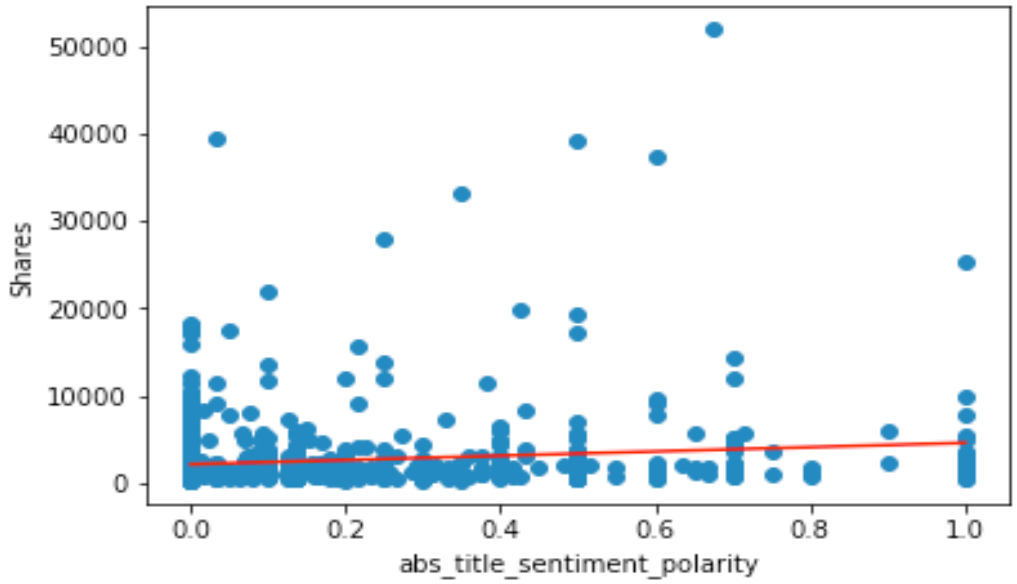
Key	Type	Size	Value
LDA_00__shares	float64	(2, 2)	array([[1. , -0.01516345], [-0.01516345, 1.]])
LDA_01__shares	float64	(2, 2)	array([[1. , 0.01471218], [0.01471218, 1.]])
LDA_02__shares	float64	(2, 2)	array([[1. , -0.06755339], [-0.06755339, 1.]])
LDA_03__shares	float64	(2, 2)	array([[1. , 0.03760529], [0.03760529, 1.]])
LDA_04__shares	float64	(2, 2)	array([[1. , 0.02485299], [0.02485299, 1.]])
abs_title_sentiment_polarity__shares	float64	(2, 2)	array([[1. , 0.13846235], [0.13846235, 1.]])
abs_title_subjectivity__shares	float64	(2, 2)	array([[1. , -0.0287263], [-0.0287263, 1.]])
average_token_length__shares	float64	(2, 2)	array([[1. , -0.05799609], [-0.05799609, 1.]])
avg_negative_polarity__shares	float64	(2, 2)	array([[1. , -0.02129341], [-0.02129341, 1.]])
avg_positive_polarity__shares	float64	(2, 2)	array([[1. , 0.03832744], [0.03832744, 1.]])
global_rate_negative_words__shares	float64	(2, 2)	array([[1. , -0.04160722], [-0.04160722, 1.]])
global_rate_positive_words__shares	float64	(2, 2)	array([[1. , -0.00544765], [-0.00544765, 1.]])
global_sentiment_polarity__shares	float64	(2, 2)	array([[1. , 0.02989428], [0.02989428, 1.]])
global_subjectivity__shares	float64	(2, 2)	array([[1. , 0.05931331], [0.05931331, 1.]])
is_weekend__shares	float64	(2, 2)	array([[1. , 0.12891116], [0.12891116, 1.]])
max_negative_polarity__shares	float64	(2, 2)	array([[1. , -0.06733751], [-0.06733751, 1.]])
max_positive_polarity__shares	float64	(2, 2)	array([[1. , 0.02477787], [0.02477787, 1.]])
min_negative_polarity__shares	float64	(2, 2)	array([[1. , 0.00757691], [0.00757691, 1.]])
min_positive_polarity__shares	float64	(2, 2)	array([[1. , 0.05525644], [0.05525644, 1.]])
n_non_stop_unique_tokens__shares	float64	(2, 2)	array([[1. , 0.02681801], [0.02681801, 1.]])
n_non_stop_words__shares	float64	(2, 2)	array([[1. , 0.00459737], [0.00459737, 1.]])
n_tokens_content__shares	float64	(2, 2)	array([[1. , 0.00222506], [0.00222506, 1.]])
n_tokens_title__shares	float64	(2, 2)	array([[1. , 0.01907454], [0.01907454, 1.]])
n_unique_tokens__shares	float64	(2, 2)	array([[1. , 0.01774484], [0.01774484, 1.]])
num_hrefs__shares	float64	(2, 2)	array([[1. , -0.02044388], [-0.02044388, 1.]])
num_imgs__shares	float64	(2, 2)	array([[1.00000000e+00, 2.22644407e-04], [2.22644407e-04, 1.00000

Cancel OK

Key	Type	Size	Value
max_positive_polarity__shares	float64	(2, 2)	array([[1. , 0.02477787], [0.02477787, 1.]])
min_negative_polarity__shares	float64	(2, 2)	array([[1. , 0.00757691], [0.00757691, 1.]])
min_positive_polarity__shares	float64	(2, 2)	array([[1. , 0.05525644], [0.05525644, 1.]])
n_non_stop_unique_tokens__shares	float64	(2, 2)	array([[1. , 0.02681801], [0.02681801, 1.]])
n_non_stop_words__shares	float64	(2, 2)	array([[1. , 0.00459737], [0.00459737, 1.]])
n_tokens_content__shares	float64	(2, 2)	array([[1. , 0.00222506], [0.00222506, 1.]])
n_tokens_title__shares	float64	(2, 2)	array([[1. , 0.01907454], [0.01907454, 1.]])
n_unique_tokens__shares	float64	(2, 2)	array([[1. , 0.01774484], [0.01774484, 1.]])
num_hrefs__shares	float64	(2, 2)	array([[1. , -0.02044388], [-0.02044388, 1.]])
num_imgs__shares	float64	(2, 2)	array([[1.00000000e+00, 2.22644407e-04], [2.22644407e-04, 1.00000 ...
num_keywords__shares	float64	(2, 2)	array([[1. , 0.03598556], [0.03598556, 1.]])
num_self_hrefs__shares	float64	(2, 2)	array([[1. , -0.02378883], [-0.02378883, 1.]])
rate_negative_words__shares	float64	(2, 2)	array([[1. , -0.02982619], [-0.02982619, 1.]])
rate_positive_words__shares	float64	(2, 2)	array([[1. , 0.03058665], [0.03058665, 1.]])
self_reference_avg_sharess__shares	float64	(2, 2)	array([[1. , 0.0740404], [0.0740404, 1.]])
self_reference_max_shares__shares	float64	(2, 2)	array([[1. , 0.04538243], [0.04538243, 1.]])
self_reference_min_shares__shares	float64	(2, 2)	array([[1. , 0.08629688], [0.08629688, 1.]])
title_sentiment_polarity__shares	float64	(2, 2)	array([[1. , 0.0243873], [0.0243873, 1.]])
title_subjectivity__shares	float64	(2, 2)	array([[1. , 0.11427668], [0.11427668, 1.]])
weekday_is_friday__shares	float64	(2, 2)	array([[1. , 0.03681992], [0.03681992, 1.]])
weekday_is_monday__shares	float64	(2, 2)	array([[1. , -0.06353339], [-0.06353339, 1.]])
weekday_is_saturday__shares	float64	(2, 2)	array([[1. , 0.07876239], [0.07876239, 1.]])
weekday_is_sunday__shares	float64	(2, 2)	array([[1. , 0.09871089], [0.09871089, 1.]])
weekday_is_thursday__shares	float64	(2, 2)	array([[1. , -0.02038031], [-0.02038031, 1.]])
weekday_is_tuesday__shares	float64	(2, 2)	array([[1. , -0.04402719], [-0.04402719, 1.]])
weekday_is_wednesday__shares	float64	(2, 2)	array([[1. , 0.00772271], [0.00772271, 1.]])

Cancel OK

From observing this array of all the correlations, we can conclude that the attribute ‘abs_title_sentiment_polarity’ has maximum correlation with ‘shares’ attribute. It has a positive correlation of 0.138. Although this is not a significant correlation, let’s go ahead and try to develop a regression model based on this attribute. We have ‘shares’ on the x –axis and ‘online_reviews.abs_title_sentiment_polarity’ on y-axis.



OLS Regression Results

Dep. Variable:	shares	R-squared:	0.019
Model:	OLS	Adj. R-squared:	0.018
Method:	Least Squares	F-statistic:	19.51
Date:	Wed, 01 Mar 2017	Prob (F-statistic):	1.11e-05
Time:	16:05:41	Log-Likelihood:	-9690.3
No. Observations:	1000	AIC:	1.938e+04
Df Residuals:	998	BIC:	1.939e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]	
const	2140.9826	148.759	14.392	0.000	1849.065	2432.900
abs_title_sentiment_polarity	2446.9301	554.014	4.417	0.000	1359.763	3534.097

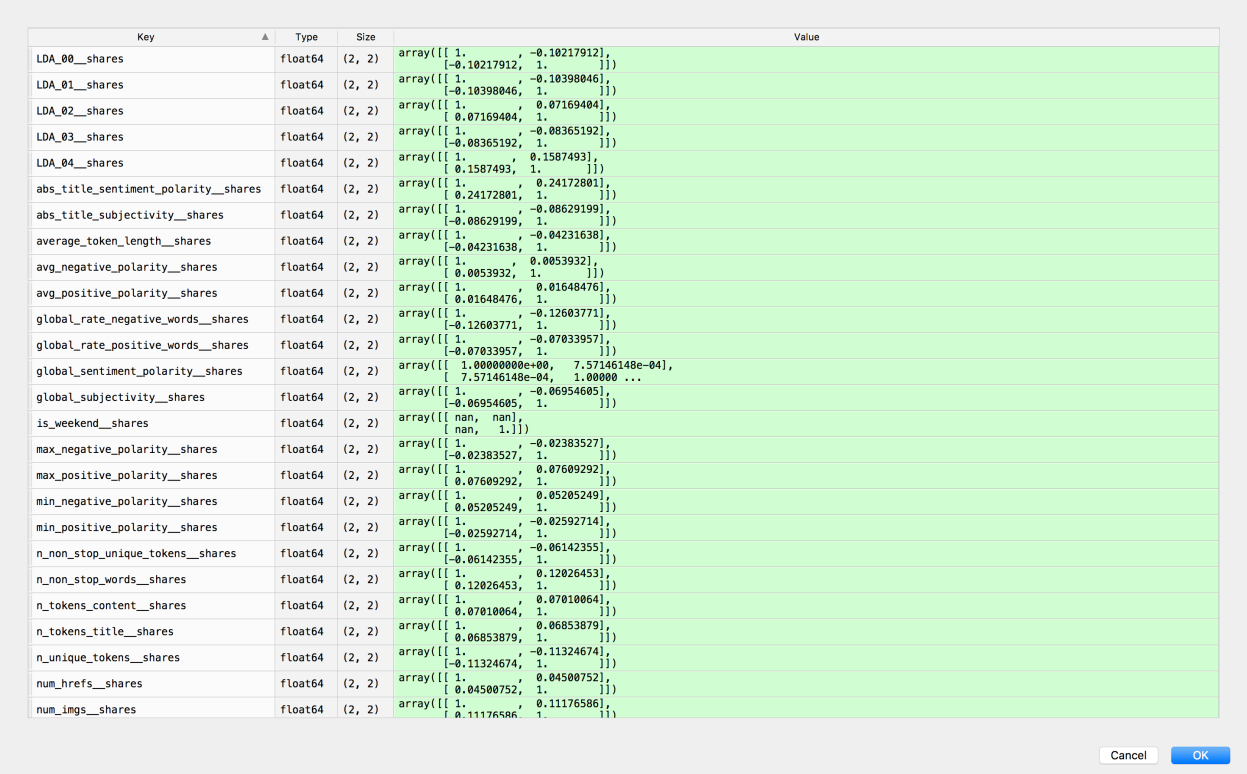
Omnibus:	1189.008	Durbin-Watson:	1.875
Prob(Omnibus):	0.000	Jarque-Bera (JB):	112540.828
Skew:	6.034	Prob(JB):	0.00
Kurtosis:	53.551	Cond. No.	4.58

Regression Equation: shares = 2140.9826 + 2446.9301*(abs_title_sentiment_polarity)

2. (WEEKEND MODEL)

Now, let's make a subset of data with only news that was published on weekends.
We get around '84' rows in the dataset. Using the same method in the previous problem, let's try to find correlation between 'shares' and other variables.

From looking at the array of correlations we can observe that 'title_sentiment_polarity' has the maximum correlation with 'shares' attribute of 0.263 and abs_title_sentiment_polarity is 0.2417 are positively correlated.



Key	Type	Size	Value
LDA_00__shares	float64	(2, 2)	array([[1. , -0.10217912], [-0.10217912, 1.]])
LDA_01__shares	float64	(2, 2)	array([[1. , -0.10398046], [-0.10398046, 1.]])
LDA_02__shares	float64	(2, 2)	array([[1. , 0.07169404], [0.07169404, 1.]])
LDA_03__shares	float64	(2, 2)	array([[1. , -0.08365192], [-0.08365192, 1.]])
LDA_04__shares	float64	(2, 2)	array([[1. , 0.1587493], [0.1587493, 1.]])
abs_title_sentiment_polarity__shares	float64	(2, 2)	array([[1. , 0.24172801], [0.24172801, 1.]])
abs_title_subjectivity__shares	float64	(2, 2)	array([[1. , -0.08629199], [-0.08629199, 1.]])
average_token_length__shares	float64	(2, 2)	array([[1. , -0.04231638], [-0.04231638, 1.]])
avg_negative_polarity__shares	float64	(2, 2)	array([[1. , 0.0053932], [0.0053932, 1.]])
avg_positive_polarity__shares	float64	(2, 2)	array([[1. , 0.01648476], [0.01648476, 1.]])
global_rate_negative_words__shares	float64	(2, 2)	array([[1. , -0.12603771], [-0.12603771, 1.]])
global_rate_positive_words__shares	float64	(2, 2)	array([[1. , -0.07033957], [-0.07033957, 1.]])
global_sentiment_polarity__shares	float64	(2, 2)	array([[1.00000000e+00, 7.57146148e-04], [7.57146148e-04, 1.00000 ...
global_subjectivity__shares	float64	(2, 2)	array([[1. , -0.06954605], [-0.06954605, 1.]])
is_weekend__shares	float64	(2, 2)	array([[nan, nan], [nan, 1.]])
max_negative_polarity__shares	float64	(2, 2)	array([[1. , -0.02383527], [-0.02383527, 1.]])
max_positive_polarity__shares	float64	(2, 2)	array([[1. , 0.07609292], [0.07609292, 1.]])
min_negative_polarity__shares	float64	(2, 2)	array([[1. , 0.05205249], [0.05205249, 1.]])
min_positive_polarity__shares	float64	(2, 2)	array([[1. , -0.02592714], [-0.02592714, 1.]])
n_non_stop_unique_tokens__shares	float64	(2, 2)	array([[1. , -0.06142355], [-0.06142355, 1.]])
n_non_stop_words__shares	float64	(2, 2)	array([[1. , 0.12026453], [0.12026453, 1.]])
n_tokens_content__shares	float64	(2, 2)	array([[1. , 0.07010064], [0.07010064, 1.]])
n_tokens_title__shares	float64	(2, 2)	array([[1. , 0.06853879], [0.06853879, 1.]])
n_unique_tokens__shares	float64	(2, 2)	array([[1. , -0.11324674], [-0.11324674, 1.]])
num_hrefs__shares	float64	(2, 2)	array([[1. , 0.04500752], [0.04500752, 1.]])
num_imgs__shares	float64	(2, 2)	array([[1. , 0.11176586], [0.11176586, 1.]])
num_keywords__shares	float64	(2, 2)	array([[1. , 0.07425952], [0.07425952, 1.]])
num_self_hrefs__shares	float64	(2, 2)	array([[1. , 0.01039508], [0.01039508, 1.]])
rate_negative_words__shares	float64	(2, 2)	array([[1. , -0.03056147], [-0.03056147, 1.]])
rate_positive_words__shares	float64	(2, 2)	array([[1. , 0.03056147], [0.03056147, 1.]])
self_reference_avg_sharess__shares	float64	(2, 2)	array([[1. , 0.17183951], [0.17183951, 1.]])
self_reference_max_shares__shares	float64	(2, 2)	array([[1. , 0.14857259], [0.14857259, 1.]])
self_reference_min_shares__shares	float64	(2, 2)	array([[1. , 0.18268482], [0.18268482, 1.]])
title_sentiment_polarity__shares	float64	(2, 2)	array([[1. , 0.2639681], [0.2639681, 1.]])
title_subjectivity__shares	float64	(2, 2)	array([[1. , 0.14096015], [0.14096015, 1.]])
weekday_is_friday__shares	float64	(2, 2)	array([[nan, nan], [nan, 1.]])
weekday_is_monday__shares	float64	(2, 2)	array([[nan, nan], [nan, 1.]])
weekday_is_saturday__shares	float64	(2, 2)	array([[1. , -0.01637014], [-0.01637014, 1.]])
weekday_is_sunday__shares	float64	(2, 2)	array([[1. , 0.01637014], [0.01637014, 1.]])
weekday_is_thursday__shares	float64	(2, 2)	array([[nan, nan], [nan, 1.]])
weekday_is_tuesday__shares	float64	(2, 2)	array([[nan, nan], [nan, 1.]])
weekday_is_wednesday__shares	float64	(2, 2)	array([[nan, nan], [nan, 1.]])

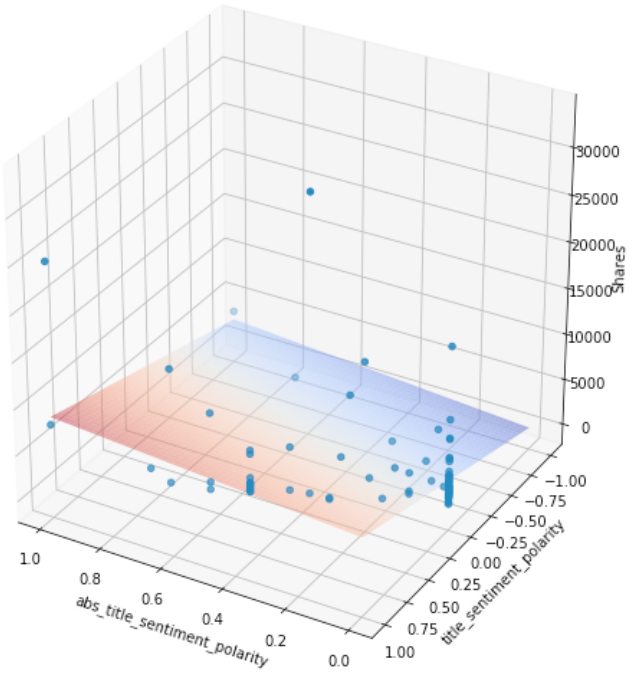
OLS Regression Results

Dep. Variable:	shares	R-squared:	0.081
Model:	OLS	Adj. R-squared:	0.059
Method:	Least Squares	F-statistic:	3.584
Date:	Fri, 03 Mar 2017	Prob (F-statistic):	0.0322
Time:	13:10:14	Log-Likelihood:	-832.89
No. Observations:	84	AIC:	1672.
Df Residuals:	81	BIC:	1679.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]	
const	3208.0244	679.407	4.722	0.000	1856.218	4559.830
abs_title_sentiment_polarity	2453.5596	2423.039	1.013	0.314	-2367.528	7274.647
title_sentiment_polarity	3083.4139	2171.116	1.420	0.159	-1236.425	7403.253

Omnibus:	82.503	Durbin-Watson:	1.706
Prob(Omnibus):	0.000	Jarque-Bera (JB):	700.402
Skew:	3.116	Prob(JB):	8.12e-153
Kurtosis:	15.700	Cond. No.	5.54

Regression Equation: shares = 3208.0244+ 2453.5596 *(abs_title_sentiment_polarity) + 3038.4139* (title_sentiment_polarity)

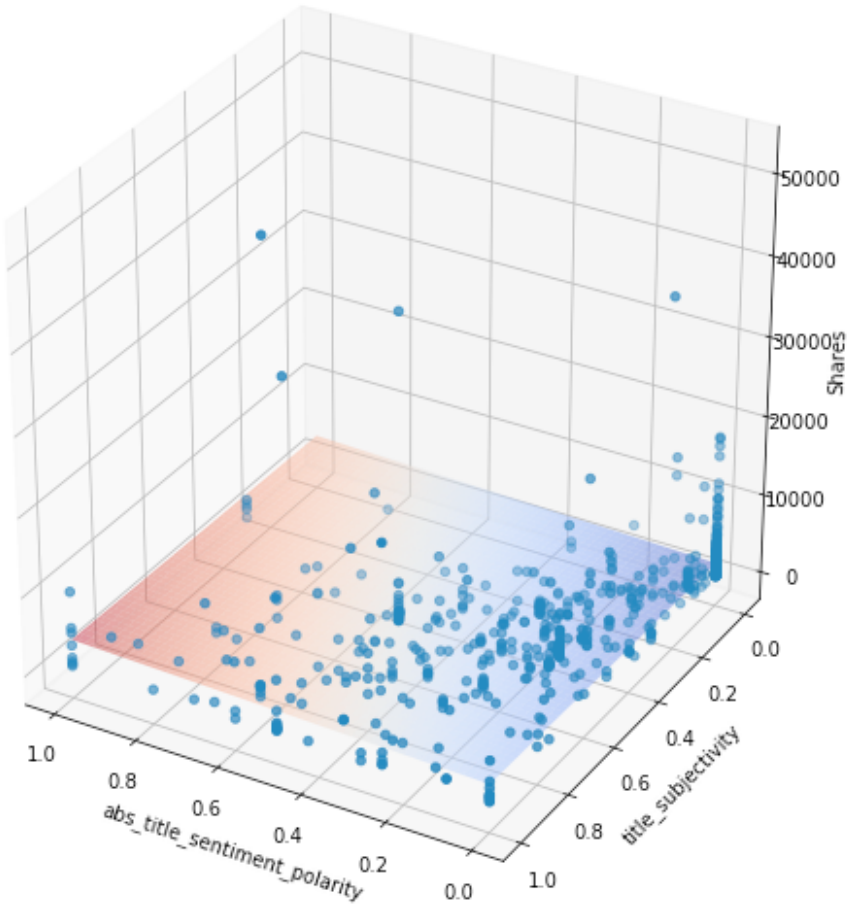


3. (WEEKDAY MODEL)
   Now, the requirement is to make subset of dataset of news which were published on weekdays. Using attribute ‘is_weekend’ is 0. We get 916 rows.

Finding correlation between ‘shares’ and rest of the variables we get.

Key	Type	Size	Value
LDA_00__shares	float64	(2, 2)	array([[1. , 0.00174045], [0.00174045, 1.]])
LDA_01__shares	float64	(2, 2)	array([[1. , 0.02882347], [0.02882347, 1.]])
LDA_02__shares	float64	(2, 2)	array([[1. , -0.07962403], [-0.07962403, 1.]])
LDA_03__shares	float64	(2, 2)	array([[1. , 0.04813263], [0.04813263, 1.]])
LDA_04__shares	float64	(2, 2)	array([[1. , 0.00377138], [0.00377138, 1.]])
abs_title_sentiment_polarity__shares	float64	(2, 2)	array([[1. , 0.11241458], [0.11241458, 1.]])
abs_title_subjectivity__shares	float64	(2, 2)	array([[1. , -0.01853944], [-0.01853944, 1.]])
average_token_length__shares	float64	(2, 2)	array([[1. , -0.06084639], [-0.06084639, 1.]])
avg_negative_polarity__shares	float64	(2, 2)	array([[1. , -0.00969725], [-0.00969725, 1.]])
avg_positive_polarity__shares	float64	(2, 2)	array([[1. , 0.0324618], [0.0324618, 1.]])
global_rate_negative_words__shares	float64	(2, 2)	array([[1. , -0.0281842], [-0.0281842, 1.]])
global_rate_positive_words__shares	float64	(2, 2)	array([[1. , -0.00156138], [-0.00156138, 1.]])
global_sentiment_polarity__shares	float64	(2, 2)	array([[1. , 0.02435579], [0.02435579, 1.]])
global_subjectivity__shares	float64	(2, 2)	array([[1. , 0.07276977], [0.07276977, 1.]])
is_weekend__shares	float64	(2, 2)	array([[nan, nan], [nan, 1.]])
max_negative_polarity__shares	float64	(2, 2)	array([[1. , -0.06023506], [-0.06023506, 1.]])
max_positive_polarity__shares	float64	(2, 2)	array([[1. , 0.00713136], [0.00713136, 1.]])
min_negative_polarity__shares	float64	(2, 2)	array([[1. , 0.01074737], [0.01074737, 1.]])
min_positive_polarity__shares	float64	(2, 2)	array([[1. , 0.06820185], [0.06820185, 1.]])
n_non_stop_unique_tokens__shares	float64	(2, 2)	array([[1. , 0.04442646], [0.04442646, 1.]])
n_non_stop_words__shares	float64	(2, 2)	array([[1. , 0.00310466], [0.00310466, 1.]])
n_tokens_content__shares	float64	(2, 2)	array([[1. , -0.01843227], [-0.01843227, 1.]])
n_tokens_title__shares	float64	(2, 2)	array([[1. , 0.00955027], [0.00955027, 1.]])
n_unique_tokens__shares	float64	(2, 2)	array([[1. , 0.04007775], [0.04007775, 1.]])
num_hrefs__shares	float64	(2, 2)	array([[1. , -0.05117318], [-0.05117318, 1.]])
num_imgs__shares	float64	(2, 2)	array([[1. , -0.01967419], [-0.01967419, 1.]])
num_keywords__shares	float64	(2, 2)	array([[1. , 0.01376926], [0.01376926, 1.]])
num_self_hrefs__shares	float64	(2, 2)	array([[1. , -0.03917522], [-0.03917522, 1.]])
rate_negative_words__shares	float64	(2, 2)	array([[1. , -0.02650169], [-0.02650169, 1.]])
rate_positive_words__shares	float64	(2, 2)	array([[1. , 0.02688492], [0.02688492, 1.]])
self_reference_avg_sharess__shares	float64	(2, 2)	array([[1. , 0.058373], [0.058373, 1.]])
self_reference_max_shares__shares	float64	(2, 2)	array([[1. , 0.02101424], [0.02101424, 1.]])
self_reference_min_shares__shares	float64	(2, 2)	array([[1. , 0.0690503], [0.0690503, 1.]])
title_sentiment_polarity__shares	float64	(2, 2)	array([[1. , -0.02565284], [-0.02565284, 1.]])
title_subjectivity__shares	float64	(2, 2)	array([[1. , 0.10384998], [0.10384998, 1.]])
weekday_is_friday__shares	float64	(2, 2)	array([[1. , 0.05513269], [0.05513269, 1.]])
weekday_is_monday__shares	float64	(2, 2)	array([[1. , -0.04705045], [-0.04705045, 1.]])
weekday_is_saturday__shares	float64	(2, 2)	array([[nan, nan], [nan, 1.]])
weekday_is_sunday__shares	float64	(2, 2)	array([[nan, nan], [nan, 1.]])
weekday_is_thursday__shares	float64	(2, 2)	array([[1. , -0.00284646], [-0.00284646, 1.]])
weekday_is_tuesday__shares	float64	(2, 2)	array([[1. , -0.02932356], [-0.02932356, 1.]])
weekday_is_wednesday__shares	float64	(2, 2)	array([[1. , 0.03341057], [0.03341057, 1.]])

By observation we can see that the attribute ‘abs_title_sentiment_polarity‘ with correlation followed by title_subjectivity.



Regression model SUMMARY:

OLS Regression Results

Dep. Variable:	shares	R-squared:	0.014
Model:	OLS	Adj. R-squared:	0.012
Method:	Least Squares	F-statistic:	6.358
Date:	Fri, 03 Mar 2017	Prob (F-statistic):	0.00181
Time:	14:13:42	Log-Likelihood:	-8840.6
No. Observations:	916	AIC:	1.769e+04
Df Residuals:	913	BIC:	1.770e+04
Df Model:	2		
Covariance Type:	nonrobust		

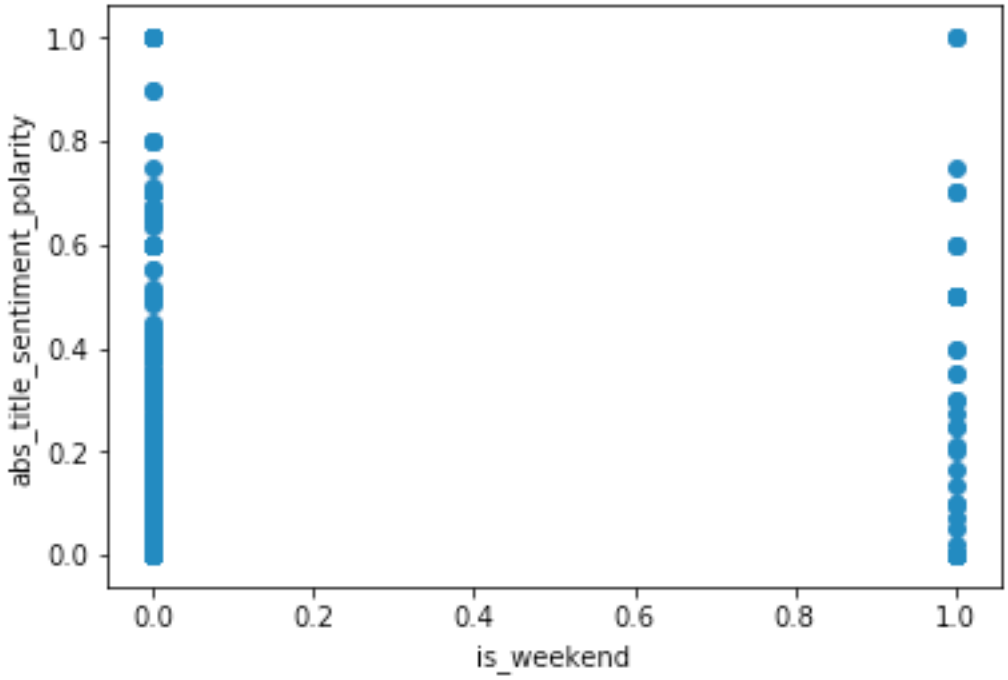
	coef	std err	t	P>|t|	[95.0% Conf.	Int.]
const	2004.3580	162.650	12.323	0.000	1685.146	2323.570
abs_title_sentiment_polarity	1362.8258	824.562	1.653	0.099	-255.431	2981.083
title_subjectivity	570.3097	565.501	1.009	0.313	-539.523	1680.142

Omnibus:	1165.573	Durbin-Watson:	1.938
Prob(Omnibus):	0.000	Jarque-Bera (JB):	149456.634
Skew:	6.645	Prob(JB):	0.00
Kurtosis:	64.149	Cond. No.	7.92

Regression Equation: shares =
2004.3580 + 1362.8258 *(abs_title_sentiment_polarity) + 570.3097*(title_subjectivity)

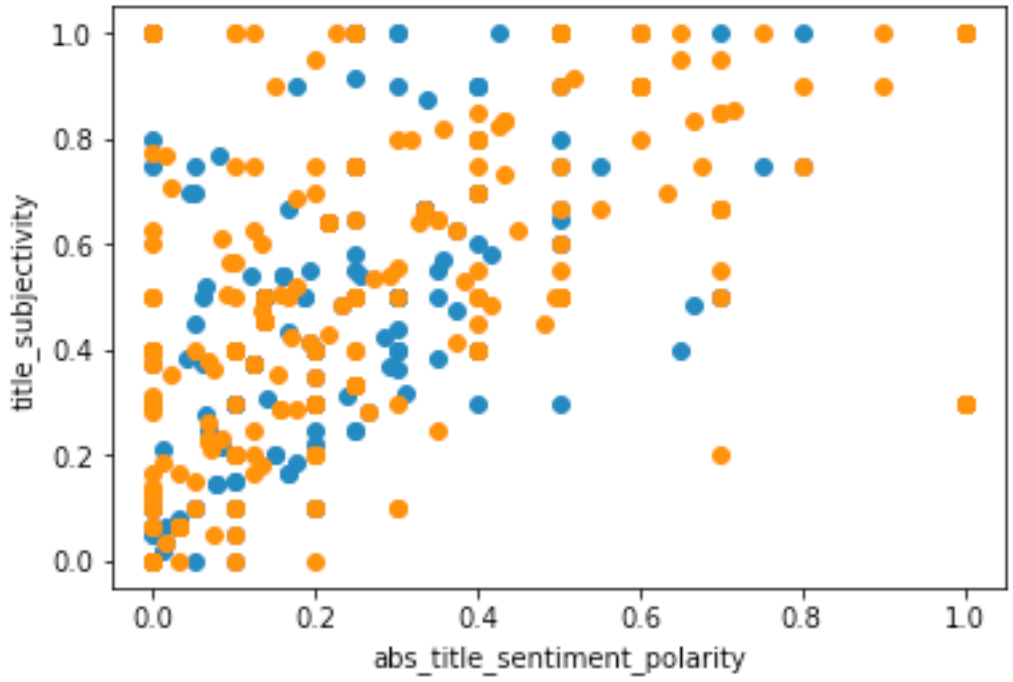
4. For weekend news model we selected 'title_sentiment_polarity' and 'abs_title_sentiment_polarity' as a predictors having correlation co-efficient 0.263 and 0.2417 for the 'weekday model' we selected 'abs_title_sentiment_polarity' ' as a predictor having correlation co-efficient 0.112 followed by title_subjectivity. With co-efficient 0.103.
   Following observations can be made about the correlation:
   - The correlation is better in weekend model than in weekday model.
   - Also, if you observe the F-score in the summary of both the models, F(prob)Weekday is 0.00181 as compared to 0.0322 that of Weekend model. The low F(prob) implies that the the independent variables (**abs_title_sentiment_polarity**, **title_subjectivity**) is not purely random with respect to the dependent variable(shares). Thus, the regression equation has some validity in fitting the data.
   - Moreover, these models does not take into account all the points in the scatter plot in a linear fashion. Thus, it can be concluded that a 'linear ' model will not perform well in this scenario. It is recommended to use non – linear model which will fit the data more accurately to get a better accuracy of predictive model.
   - Online news popularity is affected by the fact whether the news is published on weekends or weekdays.

5. In question 1, features have best two correlation with the target variable were 'abs_title_sentiment_polarity' and 'is_weekend'.

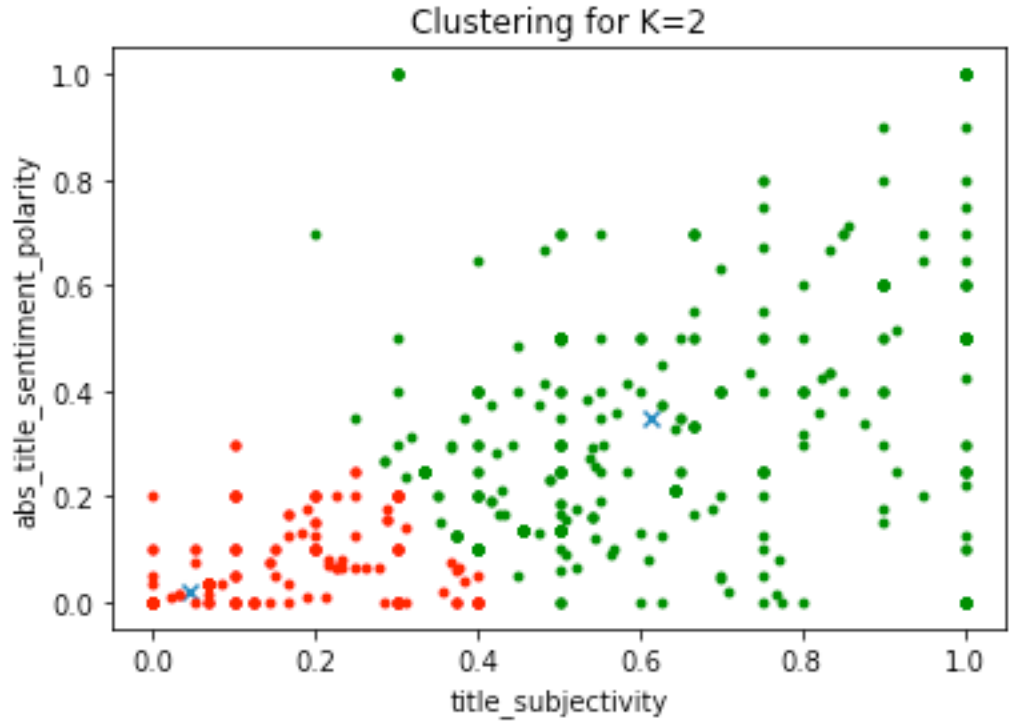


Here, is_weekend is a categorical variable and abs_title_sentiment_polarity is a continuous attribute.

So, selecting the next best correlated feature we get, title_subjectivity.

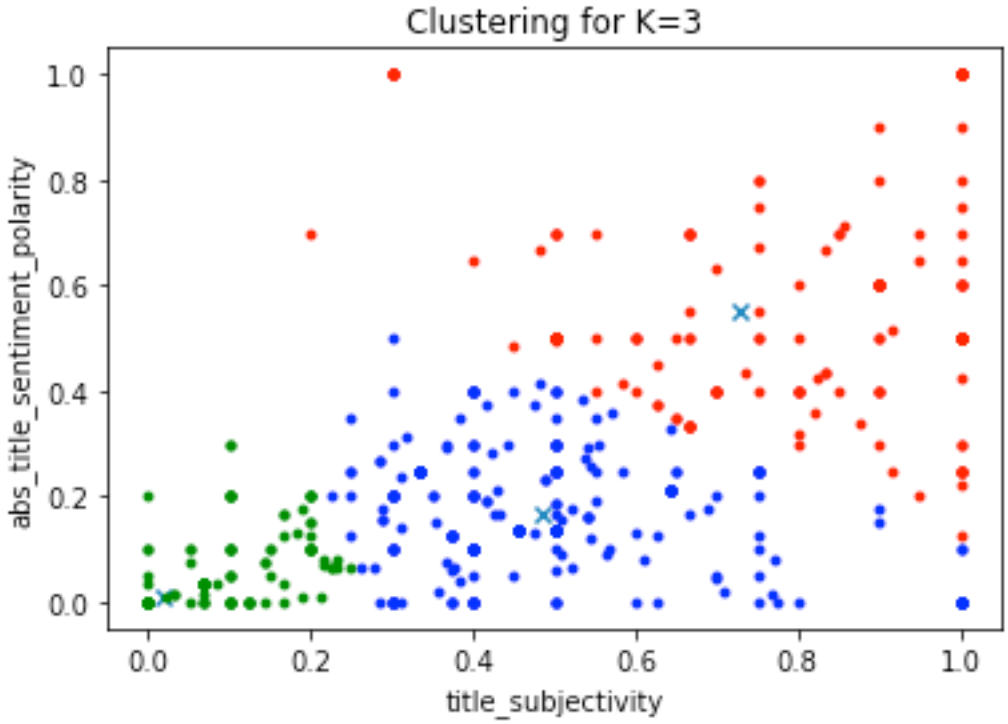


6. When k=2, we get 2 clusters using kmeans algorithm. From the scatterplot we can see that the inter-cluster distance is very small while intra cluster is higher.

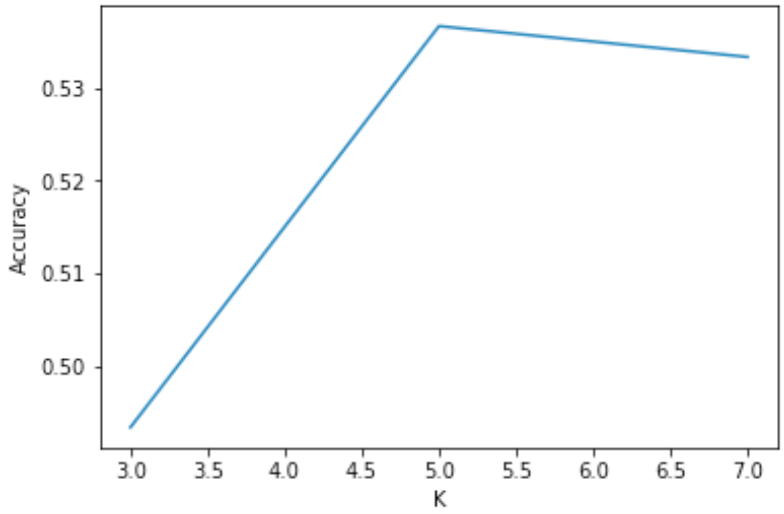


Also, much of the data is concentrated towards (0,0) which implies most of the titles are neutral and are less subjectivie.

When we take 3 clusters while unsupervised learning, the model is still not able to distinctly separate 3 regions. Some points still overlap which means kmeans is not performing while solving this problem.



7. Following graph shows the variance in Accuracy wrt K. It can be seen that the accuracy goes on increasing as the value of k increases upto a certain point and the starts decreasing. It means the model starts to over fit data after a certain point. The maximum accuracy however is around 55% which is not that great. Thus, it is recommended to try other algorithms such as SVM, Naïve – Bayes or maybe random forest.



References:
UNIX:
http://bconnelly.net/working-with-csvs-on-the-command-line/
http://www.panix.com/~elflord/unix/grep.html

Rest:
http://chrisalbon.com/python/pandas_indexing_selecting.html - For python coding.
http://www.nlreg.com/results.htm - For interpreting statistical parameters of models.