Insights from social media presence

```
Project Proposal for The Data Incubator Fellowship Program
By Muluemebet G Ayalew
Aug, 2019
```

```
In [1]: import pandas as pd    # data exploration and manupulation
        import matplotlib.pyplot as plt  # for ploting
        import seaborn as sns            # for visualization

        # to see the plot in the notebook
        % matplotlib inline
```

Read files

```
In [27]: # facebook dataframe
         fb =pd.read_csv("temp_datalab_records_social_facebook.zip", compression="zip")
```

```
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2
717: DtypeWarning: Columns (9) have mixed types. Specify dtype option on impo
rt or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

```
In [28]: #  linkedin dataframe
         ln=pd.read_csv("temp_datalab_records_linkedin_company.zip", compression="zip")
```

```
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2
717: DtypeWarning: Columns (9,10) have mixed types. Specify dtype option on i
mport or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Data Exploration

Facebook

```
In [4]: fb.shape # facebook dataframe shape
```

```
Out[4]: (3621391, 14)
```

In [5]:
```
fb.head()
```

Out[5]:

	dataset_id	time	username	checkins	has_added_app	were_here_c
0	53088	2015-01-01 05:00:00+00	SodaStream	0	f	0
1	52642	2015-01-01 05:00:00+00	ANSYSInc	148	f	0
2	53656	2015-01-01 05:00:00+00	MyAquaAmerica	0	f	0
3	53033	2015-01-01 05:00:00+00	Qualcomm	173	f	0
4	52783	2015-01-01 05:00:00+00	eaglepharmaceuticals	0	f	0

In [6]:
```
fb.info() # info about facebook dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3621391 entries, 0 to 3621390
Data columns (total 14 columns):
dataset_id             int64
time                   object
username               object
checkins               int64
has_added_app          object
were_here_count        int64
likes                  int64
talking_about_count    int64
facebook_id            int64
date_added             object
date_updated           object
entity_id              float64
cusip                  float64
isin                   float64
dtypes: float64(3), int64(6), object(5)
memory usage: 386.8+ MB
```

In [7]:
```
fb.username.nunique() # number of unique usernames in the dataframe
```

Out[7]:
```
4950
```

Linkedin

In [8]:
```
ln.shape # linkedin dataframe shape
```

Out[8]:
```
(2426196, 14)
```

In [9]:

```
ln.company_name.nunique() # number of unique companies in the dataframe
```

Out[9]:

```
5028
```

In [10]:

```
ln.head() # linkedin datafram
```

Out[10]:

	dataset_id	as_of_date	company_name	followers_count	employees_on_platform	
0	58329	2015-09-14	Goldman Sachs	552254	38124	htt
1	58329	2015-09-15	Goldman Sachs	552862	38141	htt
2	58363	2015-09-16	United Technologies	59157	14982	htt
3	58366	2015-09-16	Novo Nordisk	336175	26448	htt
4	58371	2015-09-16	Lowe's Companies, Inc.	134255	62574	htt

In [11]:

```
ln.info() #info about linkedin dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2426196 entries, 0 to 2426195
Data columns (total 14 columns):
dataset_id               int64
as_of_date               object
company_name             object
followers_count          int64
employees_on_platform    int64
link                     object
industry                 object
date_added               object
date_updated             object
description              object
website                  object
entity_id                float64
cusip                    float64
isin                     float64
dtypes: float64(3), int64(3), object(8)
memory usage: 259.1+ MB
```

```
In [12]: print(ln.columns)
         print(fb.columns)

         Index(['dataset_id', 'as_of_date', 'company_name', 'followers_count',
                'employees_on_platform', 'link', 'industry', 'date_added',
                'date_updated', 'description', 'website', 'entity_id', 'cusip', 'isi
         n'],
               dtype='object')
         Index(['dataset_id', 'time', 'username', 'checkins', 'has_added_app',
                'were_here_count', 'likes', 'talking_about_count', 'facebook_id',
                'date_added', 'date_updated', 'entity_id', 'cusip', 'isin'],
               dtype='object')
```

Data Preparation : Make the name case insensitive

```
In [13]: # remove space from company name in linkedin dataframe to have similar format
          as facebook
         lnname= ln.company_name.apply(lambda x: str(x).lower().replace(" ", ""))
```

```
In [14]: lnname.head()

Out[14]: 0             goldmansachs
         1             goldmansachs
         2       unitedtechnologies
         3              novonordisk
         4     lowe'scompanies,inc.
         Name: company_name, dtype: object
```

```
In [37]: ln.update(pd.DataFrame(lnname))  #update the naming format in the dataframe
```

```
In [38]: ln.shape

Out[38]: (2426196, 14)
```

```
In [39]: fbname=fb.username.apply(lambda x: str(x).lower()) # make the username into lo
         wercase
```

```
In [42]: fb["username"]=fbname
```

Data preparation : get similar date format

```
In [46]: fb["time"].head() # time format in facebbook dataframe

Out[46]: 0    2015-01-01 05:00:00+00
         1    2015-01-01 05:00:00+00
         2    2015-01-01 05:00:00+00
         3    2015-01-01 05:00:00+00
         4    2015-01-01 05:00:00+00
         Name: time, dtype: object
```

In [47]:

```
ln.as_of_date.head() # Linkedin date format
```

Out[47]:

```
0    2015-09-14
1    2015-09-15
2    2015-09-16
3    2015-09-16
4    2015-09-16
Name: as_of_date, dtype: object
```

In [48]:

```
# to extract only the date part of time column of the facebook dataframe
def get_date(date):
    date=str(date) # convert to string
    splited=date.split(" ") #this separates date and time
    date=splited[0] # get the date
    return date
```

In [50]:

```
fb["DATE"]=fb["time"].apply(get_date) # the DATE collumn is created on faceboo
k dataframe
```

Data Preparation : Extract Year, Month and Day

In [49]:

```
#functions to extract year, month and day from column name "time" in  facebook
dataframe
def get_year(date):
    date= str(date) # convert to string
    splited= date.split("-") # to get month, day, and year and time
    year= splited[0] # get year
    return year

def get_month(date):
    date= str(date) # convert to string
    splited= date.split("-") # to get month, day, and year and time
    month= splited[1] # get year
    return month
def get_day(date):
    date= str(date) # convert to string
    splited= date.split("-") # to get month, day, and year and time
    day= splited[2][:2] # extract only the day
    return day
```

In [97]:

```
ln.as_of_date.head()
```

Out[97]:

```
0    2015-09-14
1    2015-09-15
2    2015-09-16
3    2015-09-16
4    2015-09-16
Name: as_of_date, dtype: object
```

In [51]:

```
# add year, month and day collumns extracted from time column
fb["Year"]= fb["time"].apply(get_year)
```

In [52]:

```
fb["Month"]= fb["time"].apply(get_month)
fb["Day"]=fb["time"].apply(get_day)
```

In [53]:

```
fb.columns
```

Out[53]:

```
Index(['dataset_id', 'time', 'username', 'checkins', 'has_added_app',
       'were_here_count', 'likes', 'talking_about_count', 'facebook_id',
       'date_added', 'date_updated', 'entity_id', 'cusip', 'isin', 'DATE',
       'Year', 'Month', 'Day'],
      dtype='object')
```

In [54]:

```
fb.shape # four columns were added
```

Out[54]:

```
(3621391, 18)
```

In [ ]:

```
# convert year month and day into numeric
fb["Year"]= pd.to_numeric(fb["Year"])
fb["Month"]=pd.to_numeric(fb["Month"])
fb["Day"]=pd.to_numeric(fb["Day"])
```

Analyze Facebook Dataframe

In [74]:

```
def fb_statistics(name):
    '''The function that provides the date when a company ,its username is "na
me", was checkedin,
    liked and people talked about it most; and displays the trend in a single
 graph  '''
    name=str(name)

    if name in fb["username"].values: # check if the username is in the datafr
ame
        fb_data=fb[fb["username"]==name]

       # plot the number of likes, checkins and talking about counts in a sing
le graph
        fig =plt.figure(figsize=(14,20))

        # date vs numbers of likes
        plt.subplot(5,1,1)
        plt.plot_date(fb_data["DATE"].values ,fb_data["likes"].values , "b-")
        plt.ylabel("Number of likes")

        # date vs numbers of checkins
        plt.subplot(5,1,2)
        plt.plot_date(fb_data["DATE"].values ,fb_data["checkins"].values, "g-"
)
        plt.ylabel("Number of checkins")

         # date vs talking_about_count
        plt.subplot(5,1,3)
        plt.plot_date(fb_data["DATE"].values ,fb_data["talking_about_count"].v
alues, "r-")
        plt.ylabel("Talking_about_count")

        # date vs all
        plt.subplot(5,1,4)
        plt.plot_date(fb_data["DATE"].values ,fb_data["likes"].values , "b-")
        plt.plot_date(fb_data["DATE"].values ,fb_data["checkins"].values, "g-"
)
        plt.plot_date(fb_data["DATE"].values ,fb_data["talking_about_count"].v
alues, "r-")
        plt.legend(["likes", "checkins","talking_about"])
        plt.xlabel("Date")

        # month vs talking_about count
        plt.subplot(5,1,5)
        sns.boxplot(x="Month", y="talking_about_count", hue="Year", data=fb_da
ta)
        plt.ylabel("Talking_about_count")
        plt.xlabel("Month")

        # fig.show()


        highest_like= fb_data[fb_data["likes"]==fb_data["likes"].max()][["DAT
E", "likes"]]
        highest_checkins= fb_data[fb_data["checkins"]==fb_data["checkins"].max
()][["DATE", "checkins"]]
```

```
        highest_talking= fb_data[fb_data["talking_about_count"]==fb_data["talk
ing_about_count"].max()][["DATE", "talking_about_count"]]

        #print the text in bold and newline
        print("\n \033[1m" +"The following table shows the date when the highe
st likes, checkins and talking_about_counts were observed for "+ name.capitali
ze())

        return pd.concat((highest_like,highest_checkins,highest_talking), axis
=0)

    else:
        print("Username not found! Please try a different username")
```

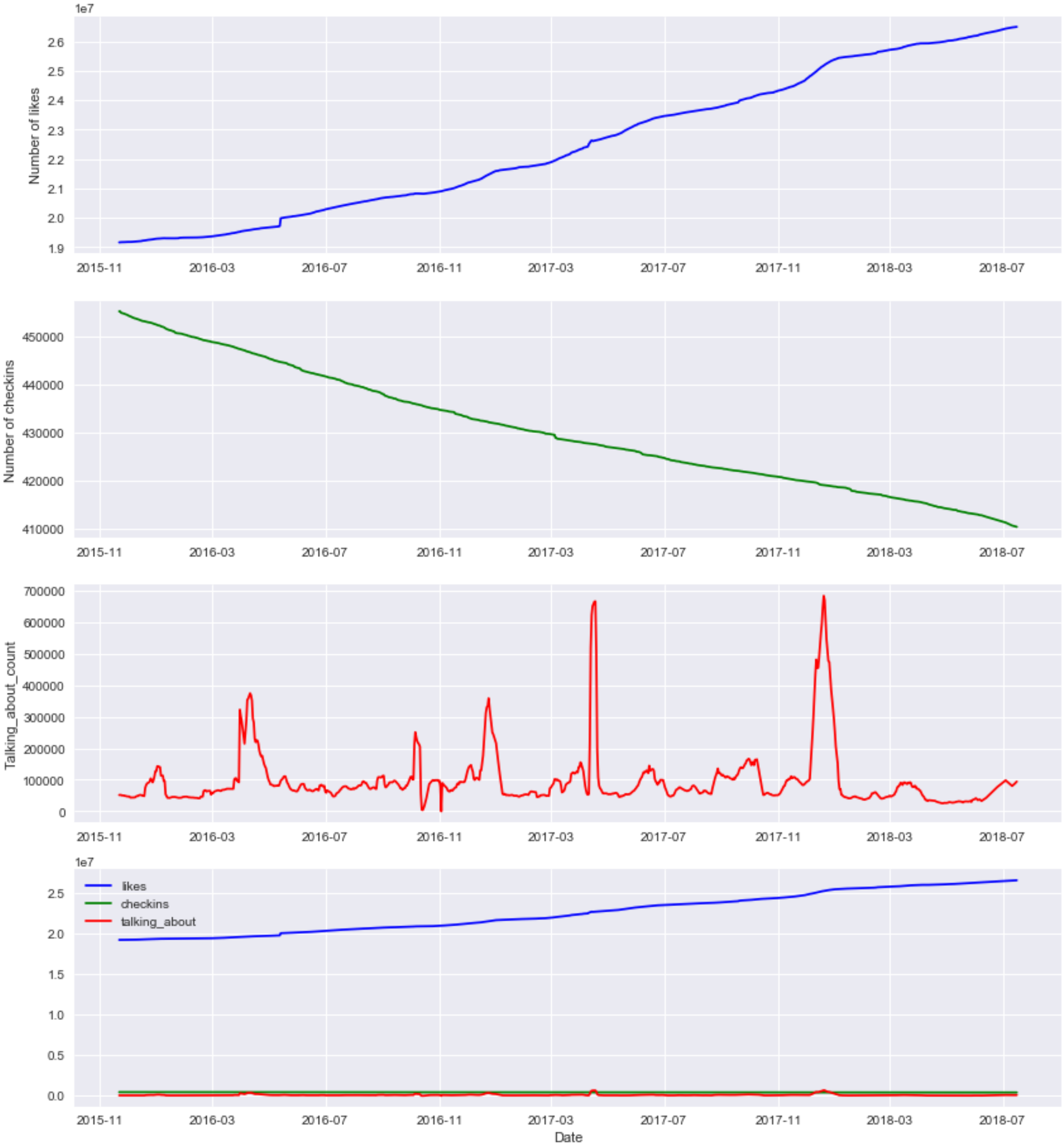
In [77]:

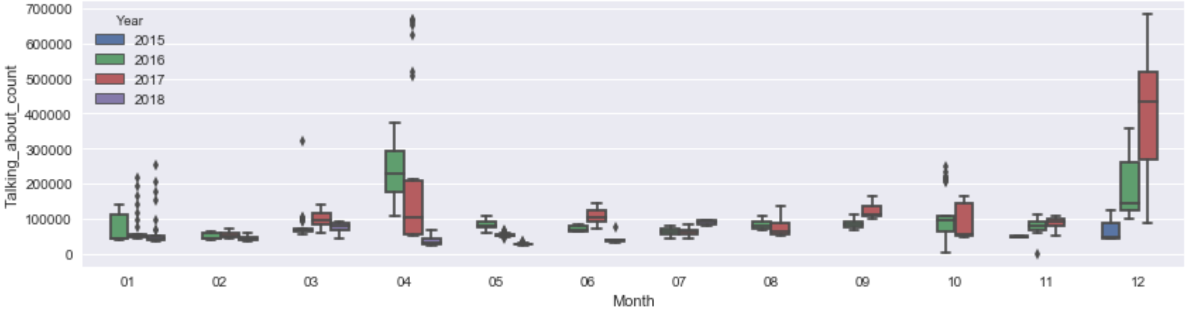
```
fb_statistics ("google")
```

The following table shows the date when the highest likes, checkins and talking_about_counts were observed for Google

Out[77]:

	DATE	checkins	likes	talking_about_count
3617742	2018-07-16	NaN	26496281.0	NaN
226343	2015-11-22	455244.0	NaN	NaN
2890282	2017-12-20	NaN	NaN	683040.0


1e7
Number of likes
2.6
2.5
2.4
2.3
2.2
2.1
2.0
1.9
2015-11
2016-03
2016-07
2016-11
2017-03
2017-07
2017-11
2018-03
2018-07
Number of checkins
450000
440000
430000
420000
410000
Talking_about_count
700000
600000
500000
400000
300000
200000
100000
0
1e7
likes
checkins
talking_about
2.5
2.0
1.5
1.0
0.5
0.0
Date

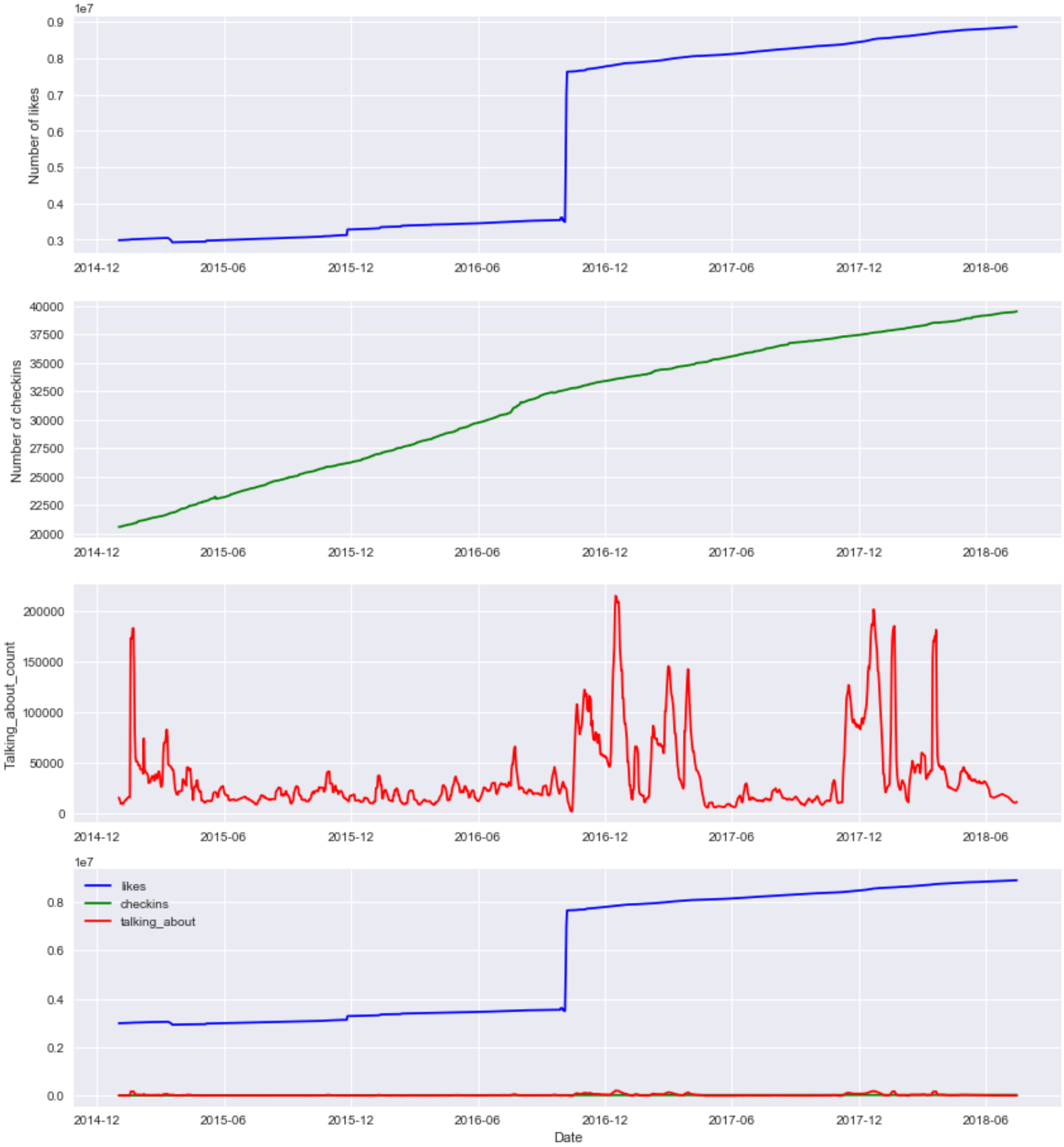

Year
2015
2016
2017
2018
Talking_about_count
700000
600000
500000
400000
300000
200000
100000
0
01
02
03
04
05
06
07
08
09
10
11
12
Month

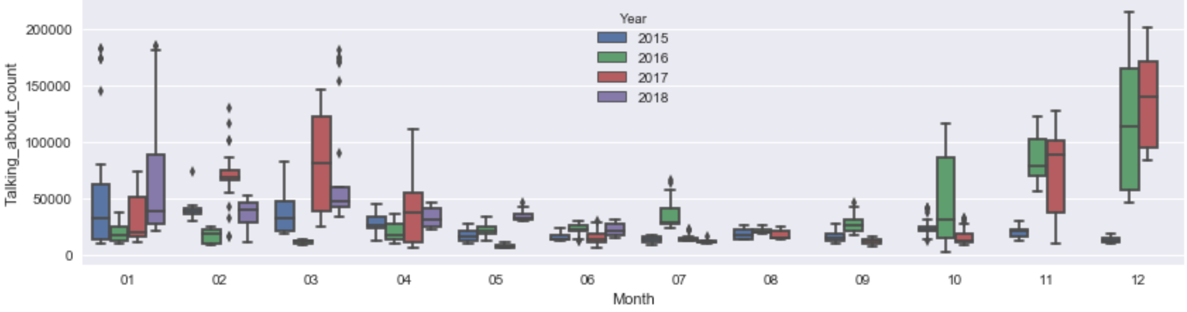
```
In [76]: fb_statistics("ford") #  Ford company
```

```
 The following table shows the date when the highest likes, checkins and talk
ing_about_counts were observed for Ford
```

Out[76]:

	DATE	checkins	likes	talking_about_count
3618029	2018-07-16	NaN	8869326.0	NaN
3618029	2018-07-16	39524.0	NaN	NaN
1313687	2016-12-16	NaN	NaN	214924.0





```
In [ ]:
```

Companies both in facebook and linkedin

In [78]:

```
fbln_merge=pd.merge(fb,ln, left_on=["DATE","username"], right_on=["as_of_date","company_name"])
```

In [ ]:

```
fbln_merge.sort_values(by=["username"])
```

In [79]:

```
fbln_merge.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 515671 entries, 0 to 515670
Data columns (total 32 columns):
dataset_id_x             515671 non-null int64
time                     515671 non-null object
username                 515671 non-null object
checkins                 515671 non-null int64
has_added_app            515671 non-null object
were_here_count          515671 non-null int64
likes                    515671 non-null int64
talking_about_count      515671 non-null int64
facebook_id              515671 non-null int64
date_added_x             379153 non-null object
date_updated_x           515671 non-null object
entity_id_x              0 non-null float64
cusip_x                  0 non-null float64
isin_x                   0 non-null float64
DATE                     515671 non-null object
Year                     515671 non-null object
Month                    515671 non-null object
Day                      515671 non-null object
dataset_id_y             515671 non-null int64
as_of_date               515671 non-null object
company_name             515671 non-null object
followers_count          515671 non-null int64
employees_on_platform    515671 non-null int64
link                     515671 non-null object
industry                 505864 non-null object
date_added_y             515671 non-null object
date_updated_y           515671 non-null object
description              127037 non-null object
website                  72971 non-null object
entity_id_y              0 non-null float64
cusip_y                  0 non-null float64
isin_y                   0 non-null float64
dtypes: float64(6), int64(9), object(17)
memory usage: 129.8+ MB
```

In [81]:

```
#remove the following columns from the dataframe
fbln_merge.drop(["cusip_x", 'isin_x','cusip_y', 'isin_y' ], axis=1, inplace=True)
```

In [ ]:

In [ ]:

In [ ]:

```

```

In [33]:

```
# datafram containing companies that are available in both fb and ln
lnfb= fb[(fb["username"].apply(lambda x: str(x).lower())).isin(lnname.values)]
```

In [ ]:

```
fbln=fb[fb["username"].isin(lnname.values)]
```

In [34]:

```
lnfb.username.nunique()
```

Out[34]:

```
1026
```

In [ ]:

```
lnfb=ln[ln.company_name.isin(fbname.values)]
```

In [ ]:

```
fb["username"].nunique() # 4950
```

In [ ]:

```
lnfb["username"].nunique() # 590
```

In [ ]:

```
lnfb["username"].nunique() # after space removed and made case insensitive , we found 1026 companies both in fb and ln
```

In [ ]:

```
ln.company_name.nunique() # 5028
```

In [ ]:

```
ln.company_name.nunique()  # 5025 unique companies
```

who has the most followers on Linkedin - google

In [86]:

```
fbln_merge[fbln_merge.followers_count==fbln_merge.followers_count.max()] # google has the most followers
```

Out[86]:

	dataset_id_x	time	username	checkins	has_added_app	were_here_cou
515003	62271	2018-07-16 04:00:00+00	google	410308	f	480

1 rows × 28 columns

In [138]:
```
#number of companies in each industry from merged dataframe
fbln_merge.groupby(by="industry")["company_name"].nunique().sort_values(ascend
ing=False).head()
```

Out[138]:
```
industry
Banking                                   100
Computer Software                          69
Information Technology and Services        60
Retail                                     57
Internet                                   57
Name: company_name, dtype: int64
```

Analyze Linkedin data

Which compnay has the most followers in linkedin - google

In [139]:
```
ln[ln.followers_count==ln.followers_count.max()] # google has the most followr
s on linkedin
```

Out[139]:

	dataset_id	as_of_date	company_name	followers_count	employees_on_platfo
2424659	58448	2018-07-17	google	7833967	140679

In [137]:
```
# number of companies in each industry
ln.groupby(by="industry")["company_name"].nunique().sort_values(ascending=Fals
e).head()
```

Out[137]:
```
industry
Biotechnology                             335
Banking                                   335
Financial Services                        302
Oil & Energy                              233
Information Technology and Services       205
Name: company_name, dtype: int64
```

Statistics about employee on platform by month and industry

In [99]:
```
ln["Month_l"]=ln.as_of_date.apply(get_month) # extract  the month
```

In [111]:
```
employee=pd.pivot_table(ln, index=["industry", "Month_l"] , values="employees_
on_platform",aggfunc=("min","mean", "median", "max"))
```

```
In [112]: employee
```

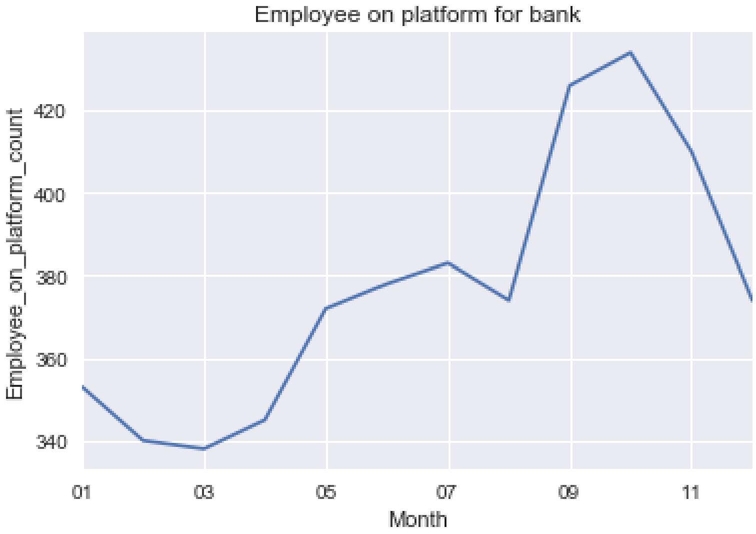
Out[112]:

		max	mean	median	min
industry	**Month_l**				
Accounting	**01**	6778	1344.496689	27.0	0
	02	6986	1980.900000	28.0	0
	03	7065	6652.523077	6970.0	0
	04	7103	6871.671429	6998.5	5790
	05	7090	6653.695652	7000.5	5800
	06	7108	6641.170455	7003.0	5760
	07	7099	6542.807692	7009.5	5624
	08	7095	6426.655172	7081.5	5634
	09	7102	6446.237288	5920.0	5642
	10	7105	6309.793651	6245.0	0
	11	7070	2106.242268	28.0	0
	12	7065	1401.639456	28.0	0
Airlines/Aviation	**01**	50558	9178.440617	2568.0	52
	02	50980	8895.937884	2366.5	54
	03	51381	8388.170149	2069.0	53
	04	51797	9303.010221	2669.0	54
	05	52343	9578.555724	2746.0	55
	06	52784	9714.781150	2792.5	55
	07	53055	9690.290284	3369.0	55
	08	48404	9622.855565	3381.0	58
	09	48877	9813.135774	3402.0	61
	10	49336	10469.502894	3993.0	62
	11	49919	10912.679134	3725.5	32
	12	50092	9483.760361	2542.0	52
Apparel & Fashion	**01**	15308	2900.005495	1793.5	56
	02	15405	3245.376597	2069.0	56
	03	19963	3457.734007	1941.0	62
	04	20263	3578.659076	2090.0	63
	05	20576	3653.960664	2169.0	125
	06	20857	3668.531726	2230.0	124
...	**...**	...	...	...	...

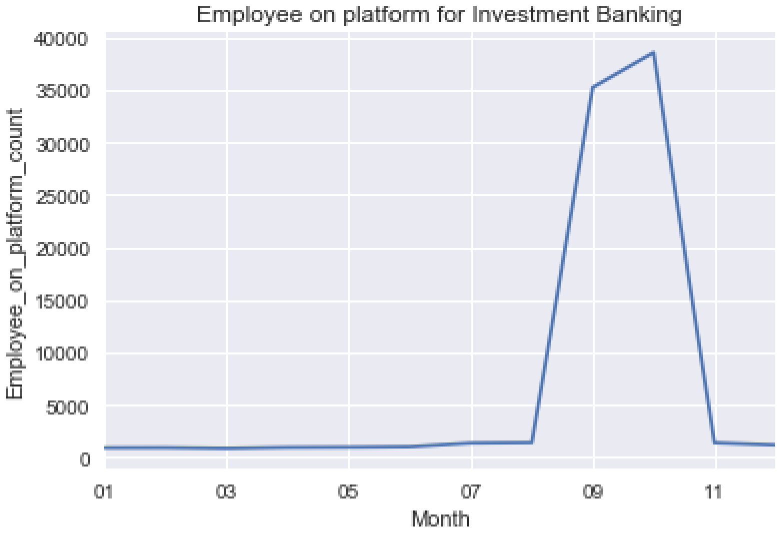
		max	mean	median	min
industry	**Month_l**				
Wine and Spirits	**07**	26866	8962.058577	3332.0	87
	08	20480	7540.840796	3350.0	87
	09	20572	7800.233871	3350.5	88
	10	21522	7894.495017	3163.0	89
	11	25430	7817.465798	3099.0	13
	12	25528	7834.829851	3129.0	13
Wireless	**01**	33119	4271.256724	425.0	11
	02	33329	4054.480959	423.0	11
	03	33522	3668.365101	424.0	14
	04	33726	3945.579762	433.0	14
	05	34039	4086.067929	443.0	14
	06	34070	4123.887719	463.0	14
	07	34042	4146.782192	449.0	14
	08	30967	4201.642447	398.0	14
	09	31094	5135.162879	411.0	14
	10	31364	5357.600324	415.0	14
	11	32676	5363.748792	420.0	11
	12	32880	4778.982289	421.0	11
Writing and Editing	**01**	18	7.148148	8.0	1
	02	18	8.186813	7.0	1
	03	18	8.623656	7.5	1
	04	18	8.666667	7.5	1
	05	18	8.698925	7.5	1
	06	18	8.836158	8.0	1
	07	18	8.881944	8.0	1
	08	18	9.000000	8.0	1
	09	18	8.849462	8.0	1
	10	18	7.897849	8.0	1
	11	18	8.680412	8.0	1
	12	18	7.558140	8.0	1

1466 rows × 4 columns

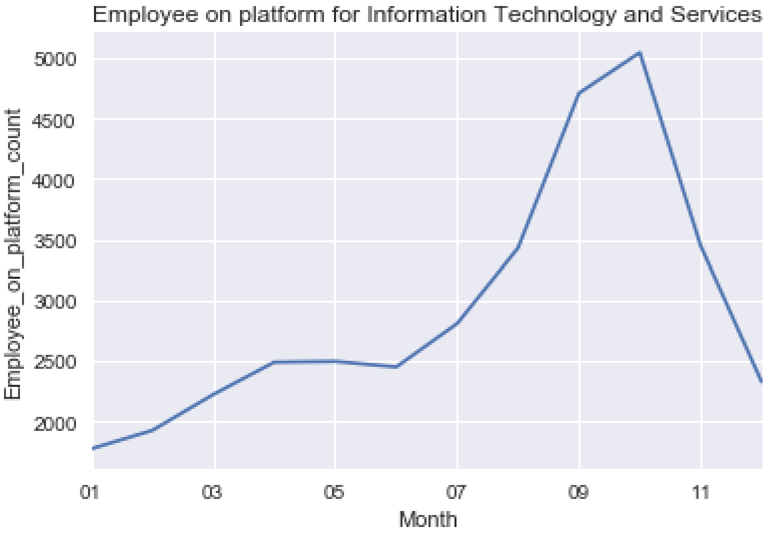
```
In [122]: employee.loc["Banking"]["median"].plot()  #
          plt.xlabel("Month")
          plt.ylabel("Employee_on_platform_count")
          plt.title("Employee on platform for bank")

Out[122]: <matplotlib.text.Text at 0x291b32f7710>
```



```
In [125]: employee.loc["Investment Banking"]["median"].plot()  #
          plt.xlabel("Month")
          plt.ylabel("Employee_on_platform_count")
          plt.title("Employee on platform for Investment Banking")

Out[125]: <matplotlib.text.Text at 0x291b32e96d8>
```



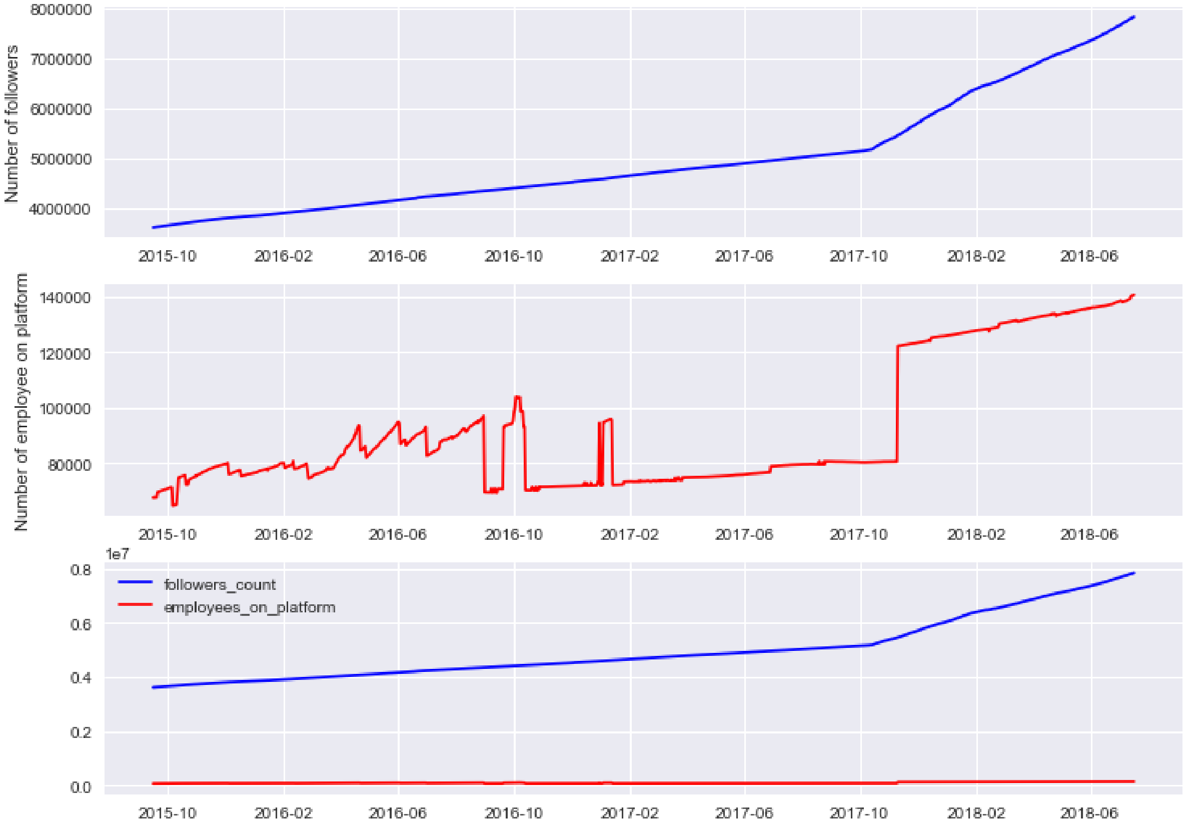
```
In [126]: employee.loc["Information Technology and Services"]["median"].plot()  #
          plt.xlabel("Month")
          plt.ylabel("Employee_on_platform_count")
          plt.title("Employee on platform for Information Technology and Services")

Out[126]: <matplotlib.text.Text at 0x291aaa1acc0>
```



In [133]:

```
def ln_statistics(name):
    '''The function that provides the date when a company ,its username is "na
me",
    had most followers count and employee on platform" ; and displays the tren
d in a single graph '''
    name=str(name)

    if name in ln["company_name"].values: # check if the name is in the datafr
ame
        ln_data=ln[ln["company_name"]==name]

       # plot the number of followers and imployee on platform  in a single gr
aph
        fig, axes=plt.subplots(nrows=3, ncols=1, figsize=(12,9))
        axes[0].plot_date(ln_data["as_of_date"].values ,ln_data["followers_cou
nt"].values ,"b-")
        axes[0].set_ylabel("Number of followers")


        axes[1].plot_date(ln_data["as_of_date"].values ,ln_data["employees_on_
platform"].values, "r-" )
        axes[1].set_ylabel("Number of employee on platform")

        axes[2].plot_date(ln_data["as_of_date"].values ,ln_data["followers_cou
nt"].values ,"b-")
        axes[2].plot_date(ln_data["as_of_date"].values ,ln_data["employees_on_
platform"].values, "r-" )
        axes[2].legend(["followers_count", "employees_on_platform"])

       # fig.show()

    else:
        print("Company name not found! Please try a different name")
```

In [136]:
```
ln_statistics("google")
```



who has the most likes on facebook

In [128]:
```
fb[fb.likes==fb.likes.max()]
```

Out[128]:

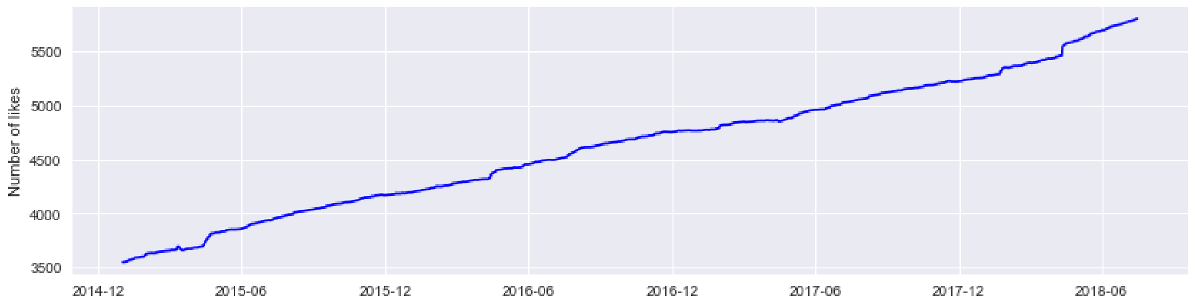
	dataset_id	time	username	checkins	has_added_app	were_here_cour
3617487	56196	2018-07-16 04:00:00+00	facebook	12	f	146272

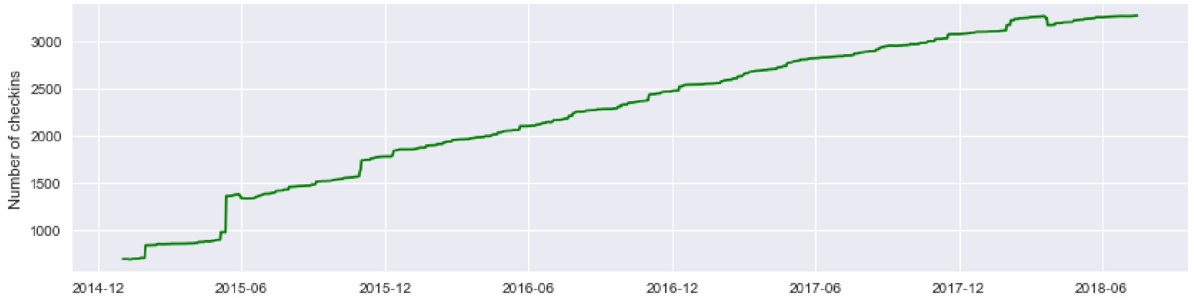
```
In [129]: fb_statistics("2u")
```

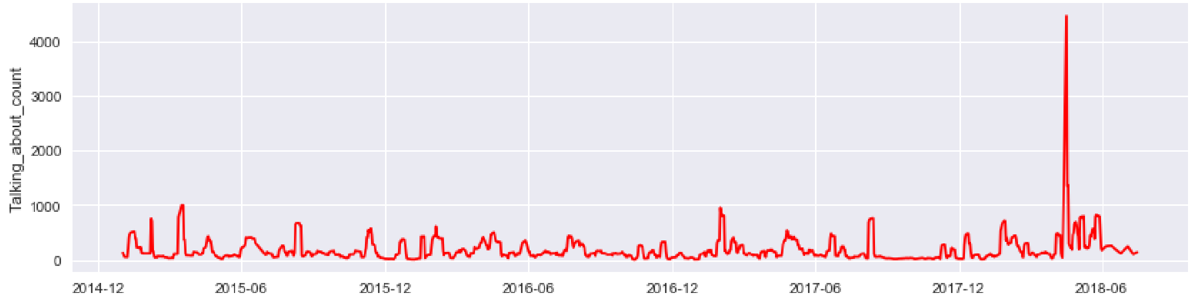
```
 The following table shows the date when the highest likes, checkins and talk
ing_about_counts were observed for 2u
```

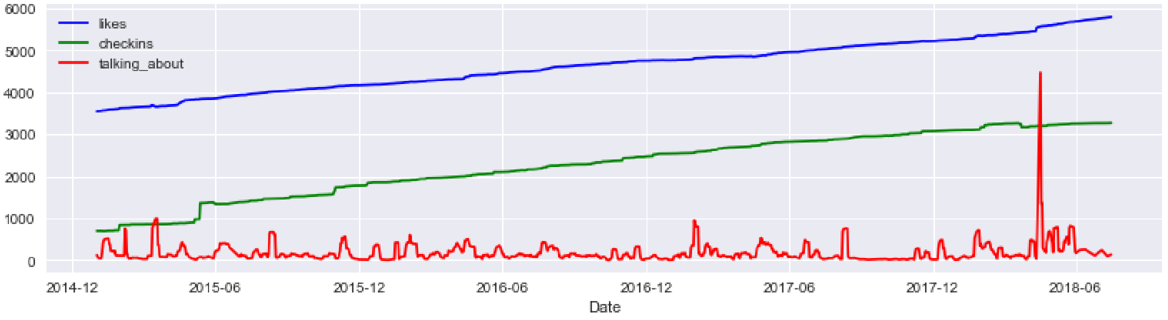
Out[129]:

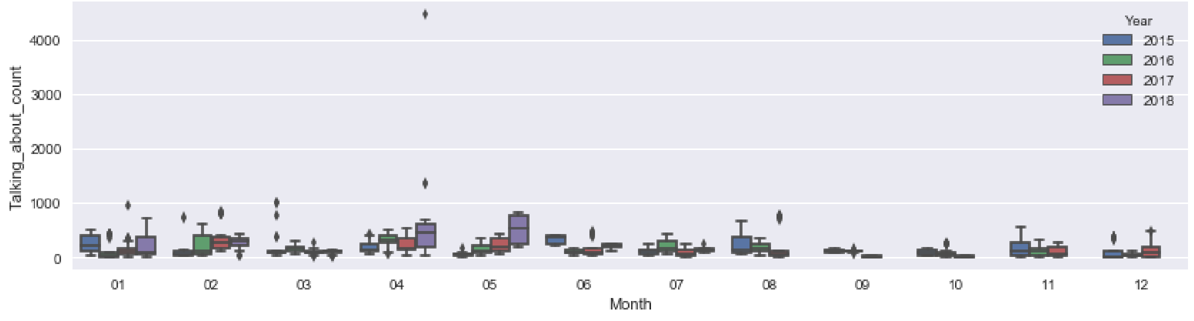
	DATE	checkins	likes	talking_about_count
3617658	2018-07-16	NaN	5799.0	NaN
3614347	2018-07-14	3271.0	NaN	NaN
3617658	2018-07-16	3271.0	NaN	NaN
3396455	2018-04-17	NaN	NaN	4469.0


Number of likes
5500
5000
4500
4000
3500
2014-12
2015-06
2015-12
2016-06
2016-12
2017-06
2017-12
2018-06

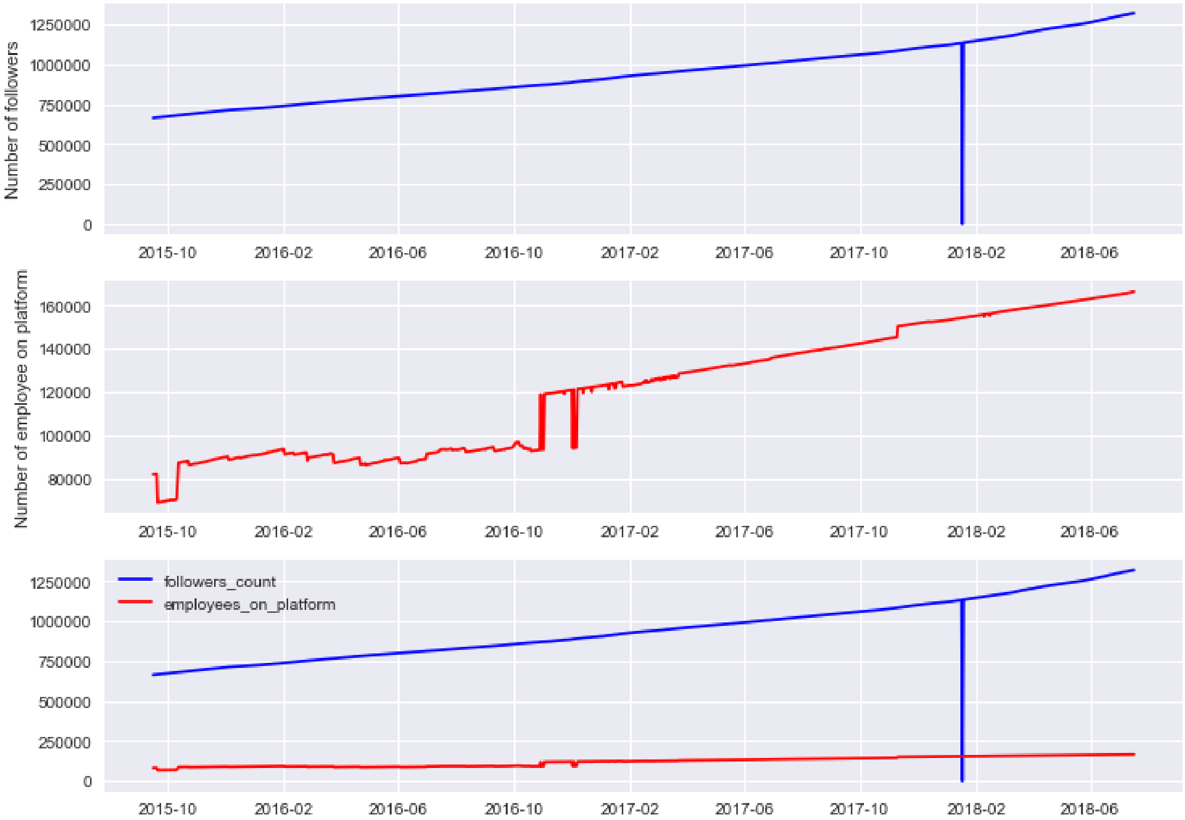

Number of checkins
3000
2500
2000
1500
1000
2014-12
2015-06
2015-12
2016-06
2016-12
2017-06
2017-12
2018-06


Talking_about_count
4000
3000
2000
1000
0
2014-12
2015-06
2015-12
2016-06
2016-12
2017-06
2017-12
2018-06


likes
checkins
talking_about
6000
5000
4000
3000
2000
1000
0
2014-12
2015-06
2015-12
2016-06
2016-12
2017-06
2017-12
2018-06
Date


Talking_about_count
Year
2015
2016
2017
2018
4000
3000
2000
1000
0
01
02
03
04
05
06
07
08
09
10
11
12
Month

```
In [166]: ln_statistics("fordmotorcompany")
```



```
In [ ]:
```

```
In [ ]:
```