Output_1

November 12, 2021

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

1 Data description

[3]:
```
# Import
df = pd.read_csv('Live.csv')
df.head()
```

[3]:

	status_id	status_type	status_published	num_reactions	num_comments	num_shares	num_likes	num_loves	num_wows	num_hahas	num_sads	num_angrys	Column1	Column2	Column3	Column4
0	246675545449582_1649696485147474	video	4/22/2018 6:00	529	512	262	432	92	3	1	1	0	NaN	NaN	NaN	NaN
1	246675545449582_1649426988507757	photo	4/21/2018 22:45	150	0	0	150	0	0	0	0	0	NaN	NaN	NaN	NaN
2	246675545449582_1648730588577397	video	4/21/2018 6:17	227	236	57	204	21	1	1	0	0	NaN	NaN	NaN	NaN
3	246675545449582_1648576705259452	photo	4/21/2018 2:29	111	0	0	111	0	0	0	0	0	NaN	NaN	NaN	NaN
4	246675545449582_1645700502213739	photo	4/18/2018 3:22	213	0	0	204	9	0	0	0	0	NaN	NaN	NaN	NaN

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7050 entries, 0 to 7049
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype
```

```
---  ------            --------------  -----
 0   status_id         7050 non-null   object
 1   status_type       7050 non-null   object
 2   status_published  7050 non-null   object
 3   num_reactions     7050 non-null   int64
 4   num_comments      7050 non-null   int64
 5   num_shares        7050 non-null   int64
 6   num_likes         7050 non-null   int64
 7   num_loves         7050 non-null   int64
 8   num_wows          7050 non-null   int64
 9   num_hahas         7050 non-null   int64
 10  num_sads          7050 non-null   int64
 11  num_angrys        7050 non-null   int64
 12  Column1           0 non-null      float64
 13  Column2           0 non-null      float64
 14  Column3           0 non-null      float64
 15  Column4           0 non-null      float64
dtypes: float64(4), int64(9), object(3)
memory usage: 881.4+ KB
```

[64]:
```
df_copy = df.copy()
df_copy.drop_duplicates(inplace=True)
df_copy.drop(['Column1', 'Column2','Column3','Column4'], axis=1, inplace=True)
df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6999 entries, 0 to 7049
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   status_id         6999 non-null   object
 1   status_type       6999 non-null   object
 2   status_published  6999 non-null   object
 3   num_reactions     6999 non-null   int64
 4   num_comments      6999 non-null   int64
 5   num_shares        6999 non-null   int64
 6   num_likes         6999 non-null   int64
 7   num_loves         6999 non-null   int64
 8   num_wows          6999 non-null   int64
 9   num_hahas         6999 non-null   int64
 10  num_sads          6999 non-null   int64
 11  num_angrys        6999 non-null   int64
dtypes: int64(9), object(3)
memory usage: 710.8+ KB
```

[65]:
```
#df_copy['status_published'] = df_copy['status_published'].str[:9]
df_copy['status_published'] = pd.to_datetime(df_copy['status_published'],␣
 ↪errors='coerce')
```

```
df_copy['status_published'] = df_copy['status_published'].dt.
 ↪strftime('%Y-%m-%d')
df_copy = df_copy.dropna()
```

[66]:
```
df_copy = df_copy.sort_values('status_published')
```

[70]:
```
df_copy.describe()
```

[70]:

	num_reactions	num_comments	num_shares	num_likes	num_loves	\
count	6999.000000	6999.000000	6999.000000	6999.000000	6999.000000	
mean	224.994571	225.552079	40.258608	209.946707	12.751536	
std	452.880746	892.743010	132.046903	439.550330	40.106872	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	17.000000	0.000000	0.000000	17.000000	0.000000	
50%	58.000000	4.000000	0.000000	57.000000	0.000000	
75%	216.000000	22.000000	4.000000	182.000000	3.000000	
max	4710.000000	20990.000000	3424.000000	4710.000000	657.000000	

	num_wows	num_hahas	num_sads	num_angrys
count	6999.000000	6999.000000	6999.000000	6999.000000
mean	1.252893	0.697957	0.232605	0.110159
std	8.725551	3.970912	1.481105	0.688582
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000
max	278.000000	157.000000	51.000000	31.000000

We have intotal 6999 non-duplicate facebook pages posts of different retail sellers records for from 9/10/2016 10:30 to 4/22/2018 6:00.

Engagement metrics consist: - type of pages posts - number of reactions - number of comments - number of shares - number of likes - number of loves - number of wows - number of hahas - number of sads - number of angrys

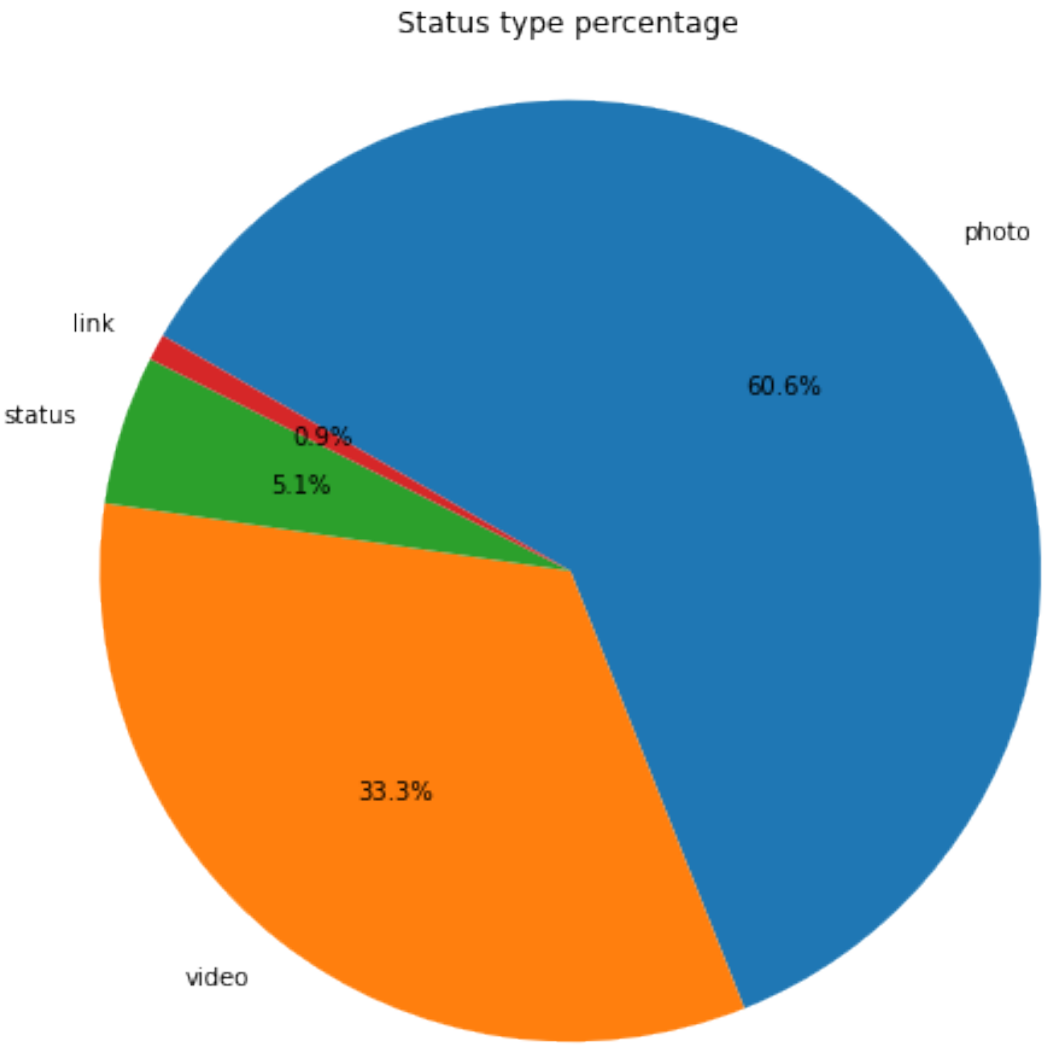
2 Data exploration and visualization

Percentage of facebook pages posts type.

Most: **Photos** Least: **Links**

[8]:
```
plt.figure(figsize = [8,8])
sorted_counts = df_copy['status_type'].value_counts()

plt.pie(sorted_counts, labels = sorted_counts.index, autopct = '%1.1f%%',␣
 ↪startangle = 150, counterclock = False);
plt.title('Status type percentage')
plt.axis('square')
```

[8]:
```
(-1.1160792851010979,
 1.1119290737673317,
 -1.1161337648246337,
 1.111874594043796)
```



2.1 links

[82]:
```
link = df_copy[df_copy['status_type']=='link']
link.describe()
```

[82]:

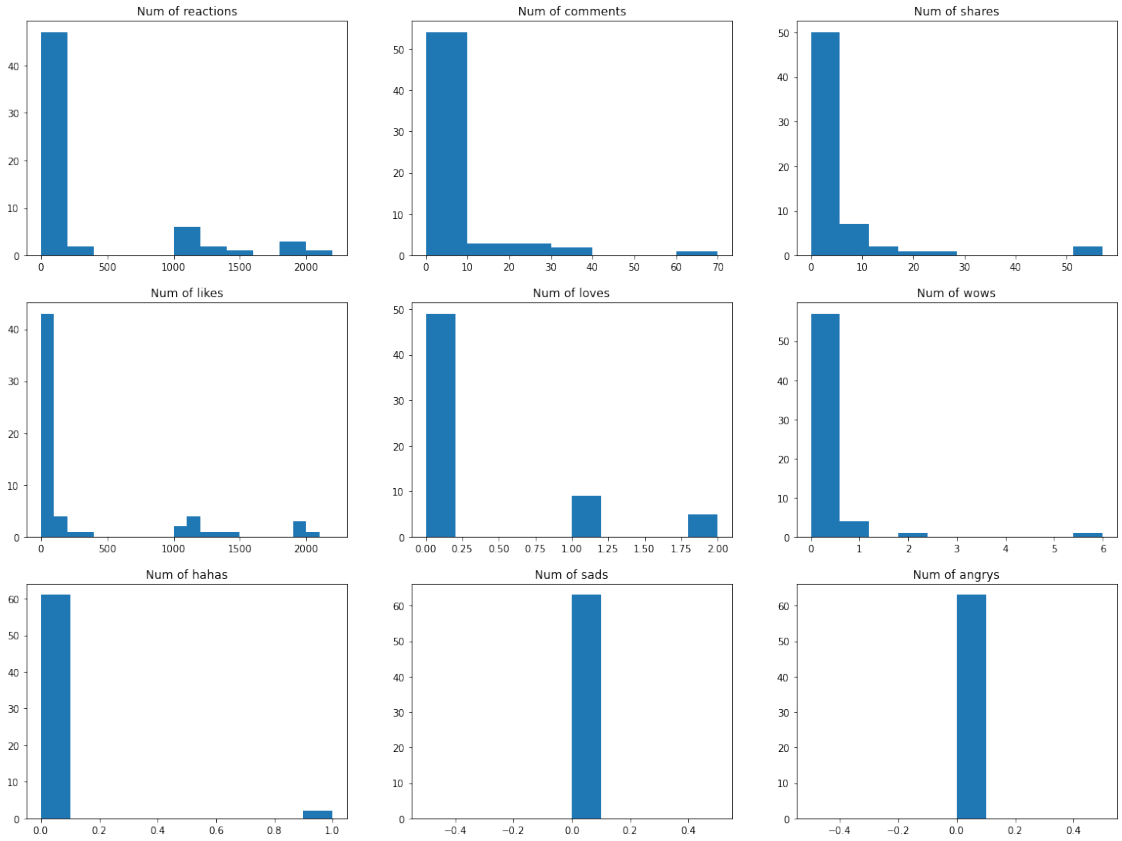
	num_reactions	num_comments	num_shares	num_likes	num_loves	\
count	63.000000	63.000000	63.000000	63.000000	63.000000	
mean	370.142857	5.698413	4.396825	369.619048	0.301587	

```
std       632.675574     11.502643   10.471990   632.878993   0.612634
min         1.000000      0.000000    0.000000     1.000000   0.000000
25%        15.500000      0.000000    0.000000    15.000000   0.000000
50%        50.000000      1.000000    0.000000    48.000000   0.000000
75%       203.000000      6.000000    4.000000   197.500000   0.000000
max      2214.000000     70.000000   57.000000  2214.000000   2.000000

        num_wows  num_hahas  num_sads  num_angrys
count  63.000000  63.000000      63.0        63.0
mean    0.190476   0.031746       0.0         0.0
std     0.820251   0.176731       0.0         0.0
min     0.000000   0.000000       0.0         0.0
25%     0.000000   0.000000       0.0         0.0
50%     0.000000   0.000000       0.0         0.0
75%     0.000000   0.000000       0.0         0.0
max     6.000000   1.000000       0.0         0.0
```

[10]:

```
plt.figure(figsize = [20, 15])


plt.subplot(3, 3, 1) # 3 row, 3 cols, subplot 1
bins = np.arange(0, link['num_reactions'].max()+4, 200)
plt.hist(data = link, x = 'num_reactions', bins=bins);
plt.title('Num of reactions');

plt.subplot(3, 3, 2)
bins = np.arange(0, link['num_comments'].max()+4, 10)
plt.hist(data = link, x = 'num_comments', bins=bins);
plt.title('Num of comments');

plt.subplot(3, 3, 3)
plt.hist(data = link, x = 'num_shares');
plt.title('Num of shares');

plt.subplot(3, 3, 4)
bins = np.arange(0, link['num_likes'].max()+4, 100)
plt.hist(data = link, x = 'num_likes', bins=bins);
plt.title('Num of likes');

plt.subplot(3, 3, 5)
plt.hist(data = link, x = 'num_loves');
plt.title('Num of loves');

plt.subplot(3, 3, 6)
plt.hist(data = link, x = 'num_wows');
plt.title('Num of wows');
```

```
plt.subplot(3, 3, 7)
plt.hist(data = link, x = 'num_hahas');
plt.title('Num of hahas');

plt.subplot(3, 3, 8)
plt.hist(data = link, x = 'num_sads');
plt.title('Num of sads');

plt.subplot(3, 3, 9)
plt.hist(data = link, x = 'num_angrys');
plt.title('Num of angrys');
```

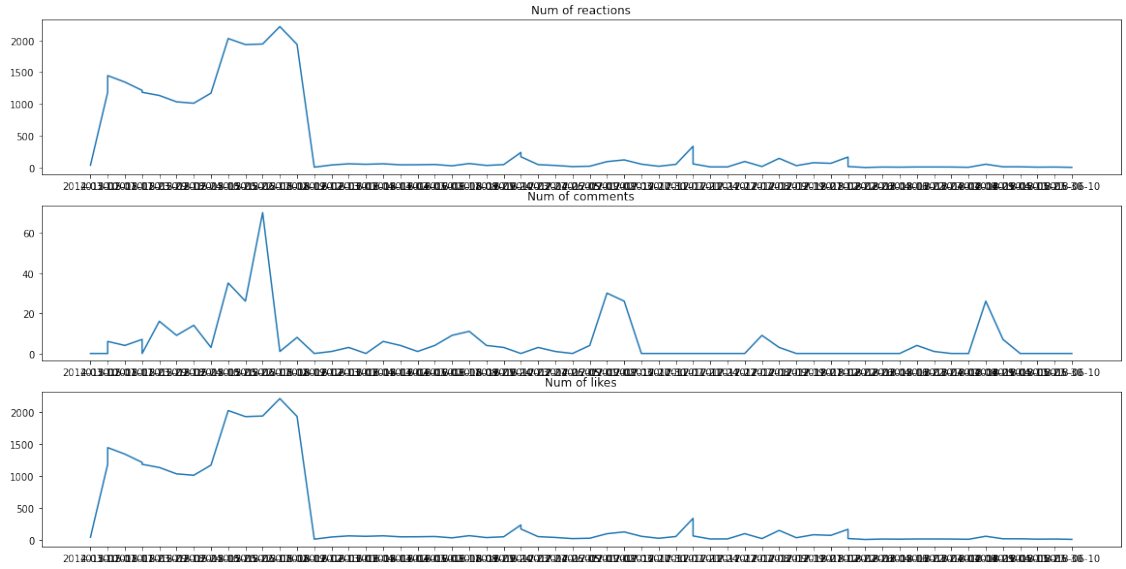


Amoung the 63 post of links, all data are highly left skewed.

we have number of reactions with a median of **50** and a mean of **370.14**. Number of comments with a median of **1** and a mean of **5.69**. Number of emotions are **nearly 0**.

[83]:
```
plt.figure(figsize = [20, 10])
plt.subplot(3, 1, 1)
plt.plot(link['status_published'],link['num_reactions'])
plt.title('Num of reactions');
```

```
plt.subplot(3, 1, 2)
plt.plot(link['status_published'],link['num_comments'])
plt.title('Num of comments');

plt.subplot(3, 1, 3)
plt.plot(link['status_published'],link['num_likes'])
plt.title('Num of likes');
```



Plot with time, we see recently video posts has number of reactions, comments and likes around 0.

2.2 photos

[80]:
```
photo = df_copy[df_copy['status_type']=='photo']
photo.describe()
```

[80]:

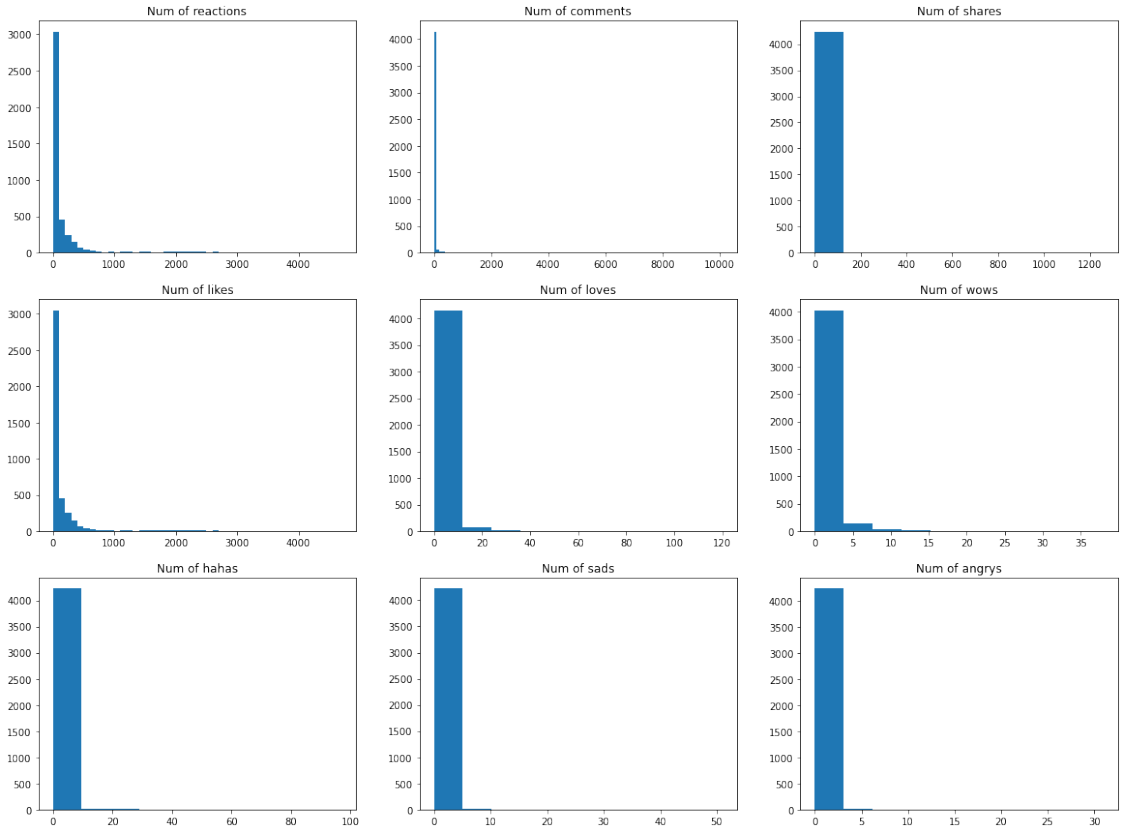
	num_reactions	num_comments	num_shares	num_likes	num_loves	\
count	4244.000000	4244.000000	4244.000000	4244.000000	4244.000000	
mean	172.330820	15.475495	2.491517	170.002356	1.354383	
std	424.942449	162.910610	22.321472	422.930092	4.228535	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	15.000000	0.000000	0.000000	15.000000	0.000000	
50%	33.000000	2.000000	0.000000	33.000000	0.000000	
75%	124.000000	9.000000	1.000000	119.000000	1.000000	
max	4710.000000	10194.000000	1260.000000	4710.000000	120.000000	

	num_wows	num_hahas	num_sads	num_angrys
count	4244.000000	4244.000000	4244.000000	4244.000000

```
mean       0.618049     0.186852     0.126296     0.039821
std        2.173039     2.234583     1.487025     0.693777
min        0.000000     0.000000     0.000000     0.000000
25%        0.000000     0.000000     0.000000     0.000000
50%        0.000000     0.000000     0.000000     0.000000
75%        0.000000     0.000000     0.000000     0.000000
max       38.000000    97.000000    51.000000    31.000000
```

[50]:
```
plt.figure(figsize = [20, 15])


plt.subplot(3, 3, 1) # 3 row, 3 cols, subplot 1
bins = np.arange(0, photo['num_reactions'].max()+4, 100)
plt.hist(data = photo, x = 'num_reactions', bins=bins);
plt.title('Num of reactions');

plt.subplot(3, 3, 2)
bins = np.arange(0, photo['num_comments'].max()+4, 100)
plt.hist(data = photo, x = 'num_comments', bins=bins);
plt.title('Num of comments');

plt.subplot(3, 3, 3)
plt.hist(data = photo, x = 'num_shares');
plt.title('Num of shares');

plt.subplot(3, 3, 4)
bins = np.arange(0, photo['num_likes'].max()+4, 100)
plt.hist(data = photo, x = 'num_likes', bins=bins);
plt.title('Num of likes');

plt.subplot(3, 3, 5)
plt.hist(data = photo, x = 'num_loves');
plt.title('Num of loves');

plt.subplot(3, 3, 6)
plt.hist(data = photo, x = 'num_wows');
plt.title('Num of wows');

plt.subplot(3, 3, 7)
plt.hist(data = photo, x = 'num_hahas');
plt.title('Num of hahas');

plt.subplot(3, 3, 8)
plt.hist(data = photo, x = 'num_sads');
plt.title('Num of sads');

plt.subplot(3, 3, 9)
```

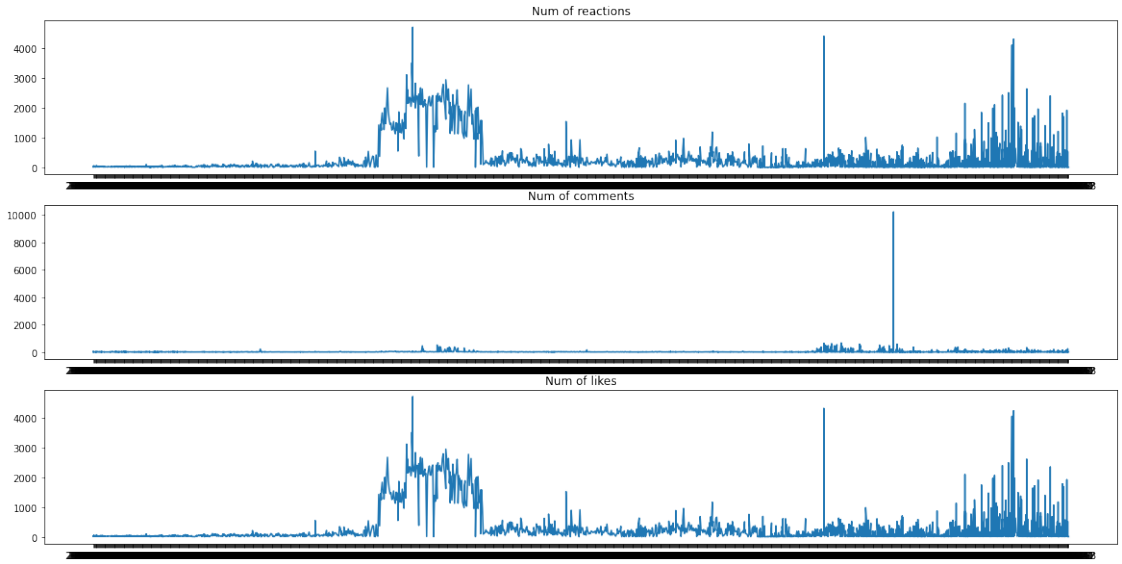
```
plt.hist(data = photo, x = 'num_angrys');
plt.title('Num of angrys');
```



Amoung the 4244 post of photos, all data are highly left skewed.

we have number of reactions with a median of **33** and a mean of **172**. Number of comments with a median of **2** and a mean of **15.48**. Number of emotions are **nearly 0**.

[81]:
```
plt.figure(figsize = [20, 10])
plt.subplot(3, 1, 1)
plt.plot(photo['status_published'],photo['num_reactions'])
plt.title('Num of reactions');

plt.subplot(3, 1, 2)
plt.plot(photo['status_published'],photo['num_comments'])
plt.title('Num of comments');

plt.subplot(3, 1, 3)
plt.plot(photo['status_published'],photo['num_likes'])
plt.title('Num of likes');
```



Plot with time, we see recently photo posts has number of reactions about 1000, comments about 0, which is the smallest and likes about 500-1000.

3 status

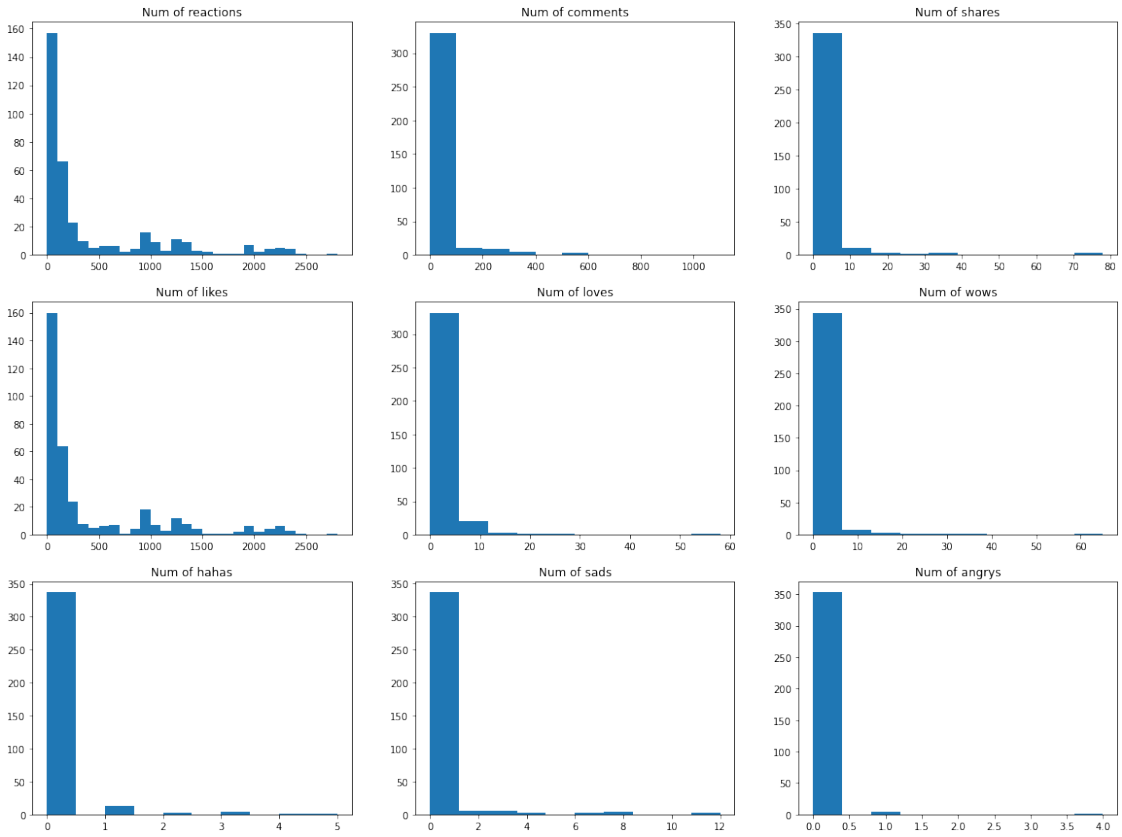
[72]:
```
status = df_copy[df_copy['status_type']=='status']
status.describe()
```

[72]:

	num_reactions	num_comments	num_shares	num_likes	num_loves	\
count	359.000000	359.000000	359.000000	359.000000	359.000000	
mean	442.740947	36.428969	2.576602	439.545961	1.529248	
std	627.361333	96.709695	9.369456	625.685222	4.254292	
min	3.000000	0.000000	0.000000	3.000000	0.000000	
25%	52.500000	2.000000	0.000000	51.500000	0.000000	
50%	118.000000	9.000000	0.000000	115.000000	0.000000	
75%	641.500000	21.500000	1.000000	631.500000	1.000000	
max	2799.000000	1186.000000	78.000000	2799.000000	58.000000	

	num_wows	num_hahas	num_sads	num_angrys
count	359.000000	359.000000	359.000000	359.000000
mean	1.178273	0.111421	0.350975	0.025070
std	4.896867	0.521984	1.437504	0.240892
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000
max	65.000000	5.000000	12.000000	4.000000

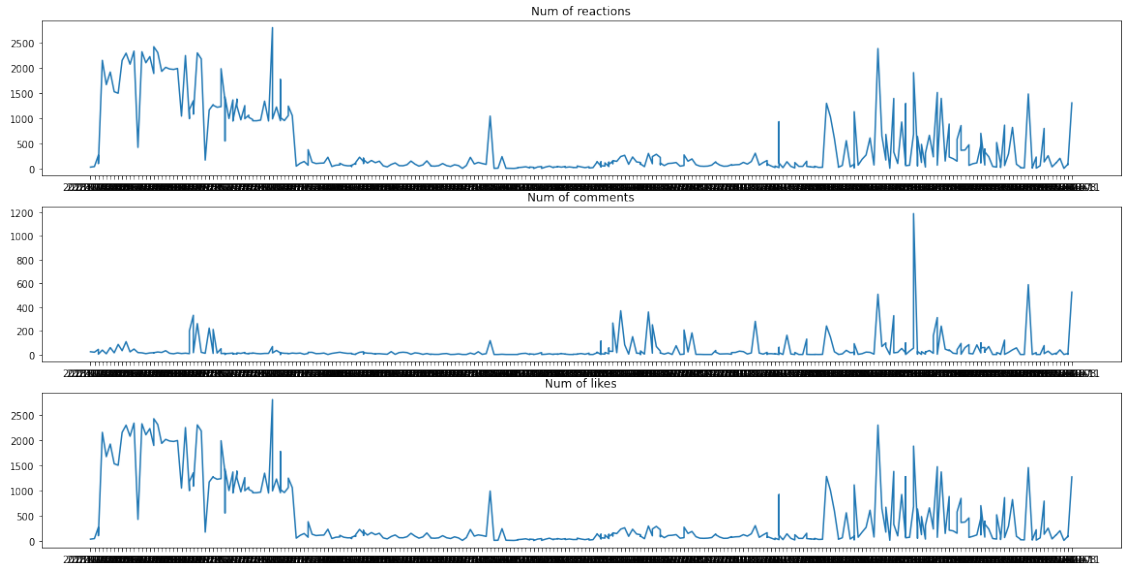
[52]:
```
plt.figure(figsize = [20, 15])


plt.subplot(3, 3, 1) # 3 row, 3 cols, subplot 1
bins = np.arange(0, status['num_reactions'].max()+4, 100)
plt.hist(data = status, x = 'num_reactions', bins=bins);
plt.title('Num of reactions');

plt.subplot(3, 3, 2)
bins = np.arange(0, status['num_comments'].max()+4, 100)
plt.hist(data = status, x = 'num_comments', bins=bins);
plt.title('Num of comments');

plt.subplot(3, 3, 3)
plt.hist(data = status, x = 'num_shares');
plt.title('Num of shares');

plt.subplot(3, 3, 4)
bins = np.arange(0, status['num_likes'].max()+4, 100)
plt.hist(data = status, x = 'num_likes', bins=bins);
plt.title('Num of likes');

plt.subplot(3, 3, 5)
plt.hist(data = status, x = 'num_loves');
plt.title('Num of loves');

plt.subplot(3, 3, 6)
plt.hist(data = status, x = 'num_wows');
plt.title('Num of wows');

plt.subplot(3, 3, 7)
plt.hist(data = status, x = 'num_hahas');
plt.title('Num of hahas');

plt.subplot(3, 3, 8)
plt.hist(data = status, x = 'num_sads');
plt.title('Num of sads');

plt.subplot(3, 3, 9)
plt.hist(data = status, x = 'num_angrys');
plt.title('Num of angrys');
```



Amoung the 359 post of status, all data are highly left skewed.

we have number of reactions with a median of **118** and a mean of **442.74**. Number of comments with a median of **9** and a mean of **36.43**. Number of emotions are **nearly 0**.

[78]:
```
plt.figure(figsize = [20, 10])
plt.subplot(3, 1, 1)
plt.plot(status['status_published'],status['num_reactions'])
plt.title('Num of reactions');

plt.subplot(3, 1, 2)
plt.plot(status['status_published'],status['num_comments'])
plt.title('Num of comments');

plt.subplot(3, 1, 3)
plt.plot(status['status_published'],status['num_likes'])
plt.title('Num of likes');
```



Plot with time, we see recently status posts has number of reactions about 500-1000, comments about 0-200 and likes about 500-1000.

4 video

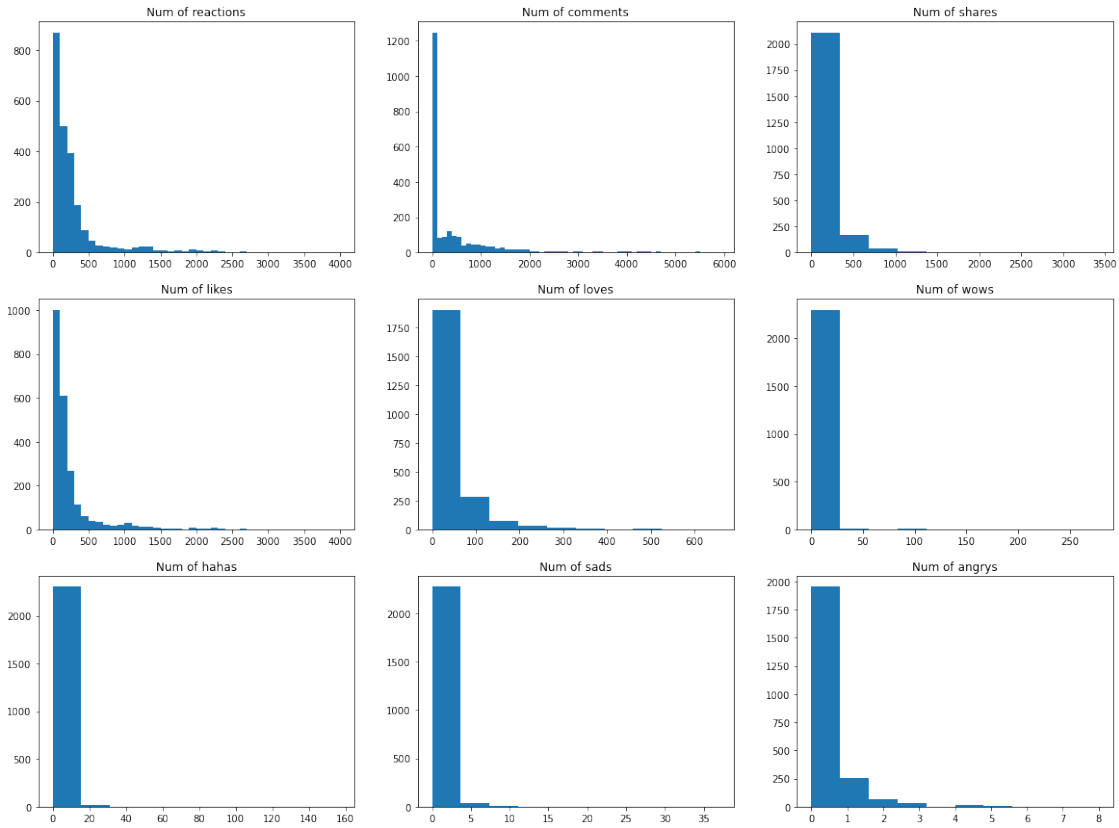
[71]:
```
video = df_copy[df_copy['status_type']=='video']
video.describe()
```

[71]:

	num_reactions	num_comments	num_shares	num_likes	num_loves	\
count	2333.000000	2333.000000	2333.000000	2333.000000	2333.000000	
mean	283.369910	642.744964	115.728247	242.967853	35.547364	
std	446.696999	1442.453954	207.011857	413.261014	63.339545	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	28.000000	0.000000	0.000000	27.000000	0.000000	
50%	160.000000	39.000000	12.000000	117.000000	3.000000	
75%	296.000000	672.000000	169.000000	238.000000	49.000000	
max	4094.000000	20990.000000	3424.000000	4094.000000	657.000000	

	num_wows	num_hahas	num_sads	num_angrys
count	2333.000000	2333.000000	2333.000000	2333.000000
mean	2.447921	1.735962	0.414059	0.254179
std	14.628283	6.047459	1.478261	0.712024
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000
75%	1.000000	2.000000	0.000000	0.000000
max	278.000000	157.000000	37.000000	8.000000

[55]:
```
plt.figure(figsize = [20, 15])


plt.subplot(3, 3, 1) # 3 row, 3 cols, subplot 1
bins = np.arange(0, video['num_reactions'].max()+4, 100)
plt.hist(data = video, x = 'num_reactions', bins=bins);
plt.title('Num of reactions');

plt.subplot(3, 3, 2)
bins = np.arange(0, 6000, 100)
plt.hist(data = video, x = 'num_comments', bins=bins);
plt.title('Num of comments');

plt.subplot(3, 3, 3)
plt.hist(data = video, x = 'num_shares');
plt.title('Num of shares');

plt.subplot(3, 3, 4)
bins = np.arange(0, video['num_likes'].max()+4, 100)
plt.hist(data = video, x = 'num_likes', bins=bins);
plt.title('Num of likes');

plt.subplot(3, 3, 5)
plt.hist(data = video, x = 'num_loves');
plt.title('Num of loves');

plt.subplot(3, 3, 6)
plt.hist(data = video, x = 'num_wows');
plt.title('Num of wows');

plt.subplot(3, 3, 7)
plt.hist(data = video, x = 'num_hahas');
plt.title('Num of hahas');

plt.subplot(3, 3, 8)
plt.hist(data = video, x = 'num_sads');
plt.title('Num of sads');

plt.subplot(3, 3, 9)
plt.hist(data = video, x = 'num_angrys');
plt.title('Num of angrys');
```

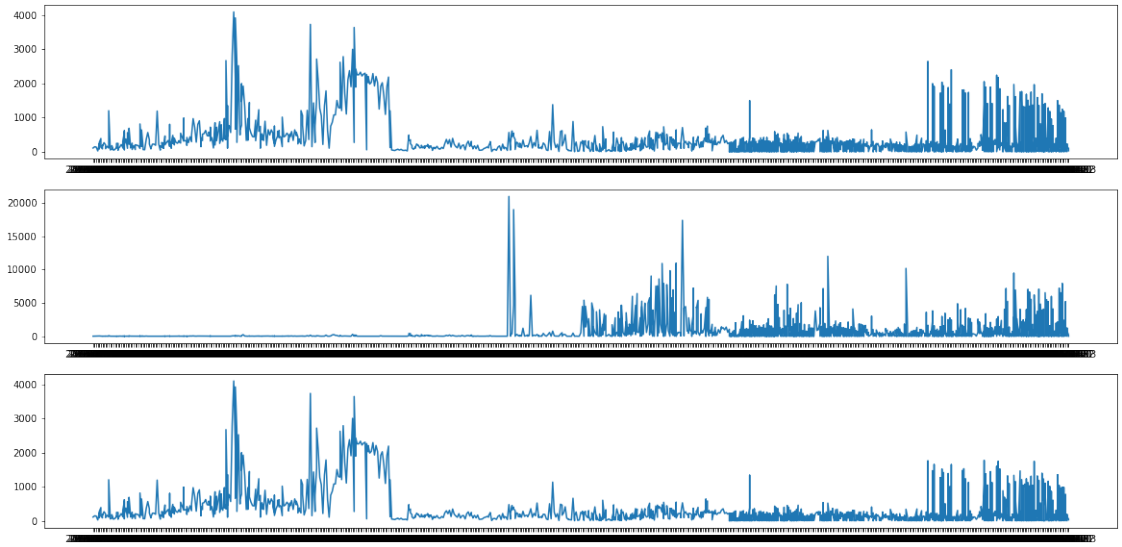


Amoung the 2333 post of video, all data are highly left skewed.

we have number of reactions with a median of **160** and a mean of **283.37**. Number of comments with a median of **39** and a mean of **642.74**. Video has a much higher comments number. Number of emotions are higher than other type of posts.

[77]:
```
plt.figure(figsize = [20, 10])
plt.subplot(3, 1, 1)
plt.plot(video['status_published'],video['num_reactions'])
plt.title('Num of reactions');

plt.subplot(3, 1, 2)
plt.plot(video['status_published'],video['num_comments'])
plt.title('Num of comments');

plt.subplot(3, 1, 3)
plt.plot(video['status_published'],video['num_likes'])
plt.title('Num of likes');
```

[77]:
```
[<matplotlib.lines.Line2D at 0x7fb3f052ff10>]
```



Plot with time, we see recently video posts has number of reactions about 1000-2000, comments about 3000-5000 and likes about 1000.

5 compare

[6]:
```
category_means_reaction = df_copy.groupby('status_type').num_reactions.mean()
category_means_reaction
```

[6]:
```
status_type
link      370.142857
photo     181.290345
status    438.783562
video     283.409597
Name: num_reactions, dtype: float64
```

Status has the highest status compare to other type of posts.

[5]:
```
category_means_comments = df_copy.groupby('status_type').num_comments.mean()
category_means_comments
```

[5]:
```
status_type
link        5.698413
photo      15.993470
status     36.238356
video     642.478149
Name: num_comments, dtype: float64
```

[12]:
```
category_means_shares = df_copy.groupby('status_type').num_shares.mean()
category_means_shares
```

[12]:
```
status_type
link        4.396825
photo       2.491517
status      2.576602
video     115.728247
Name: num_shares, dtype: float64
```

We see video has a much higher comments and shares compare to other types of post.