act_report

EDA

首先读取twitter_archive_master.csv文件，将文件中的信息存入一个新的dataframe中。然后执行常规数据评估和探索步骤：

1. 查看数据属性
2. 查看数据维度
3. 视觉评估数据
4. 综合参看数据
   A. 数据完整性
   B. 数据类型使用恰当
5. 常规统计数据
6. 查看数据缺失程度

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
df_weratedog_complete = pd.read_csv('twitter_archive_master.csv')
```

In [2]:
```
df_weratedog_complete.columns
```

Out[2]:
```
Index(['tweet_id', 'timestamp', 'source', 'text', 'retweeted_status_id',
       'retweeted_status_user_id', 'retweeted_status_timestamp',
       'expanded_urls', 'rating_numerator', 'rating_denominator', 'name',
       'stage', 'retweet_count', 'favorite_count', 'rating', 'jpg_url',
       'img_num', 'p1', 'p1_conf', 'p1_dog', 'p2', 'p2_conf', 'p2_dog', 'p3',
       'p3_conf', 'p3_dog'],
      dtype='object')
```

In [3]:
```
df_weratedog_complete.shape
```

Out[3]:
```
(1852, 26)
```

In [4]:
```
df_weratedog_complete.head()
```

Out[4]:

	tweet_id	timestamp	source	text	retweeted_status_id	retweeted_status_user_id	retweeted_status_timestamp	
0	892420643555336193	2017-08-01 16:23:56	Twitter for iPhone	This is Phineas. He's a mystical boy. Only eve...	NaN	NaN	NaN	https://twitter.cc
1	892177421306343426	2017-08-01 00:17:27	Twitter for iPhone	This is Tilly. She's just checking pup on you....	NaN	NaN	NaN	https://twitter.cc
2	891815181378084864	2017-07-31 00:18:03	Twitter for iPhone	This is Archie. He is a rare Norwegian Pouncin...	NaN	NaN	NaN	https://twitter.cc
3	891689557279858688	2017-07-30 15:58:51	Twitter for iPhone	This is Darla. She commenced a snooze mid meal...	NaN	NaN	NaN	https://twitter.cc
4	891327558926688256	2017-07-29 16:00:24	Twitter for iPhone	This is Franklin. He would like you to stop ca...	NaN	NaN	NaN	https://twitter.cc

5 rows × 26 columns

In [5]:
```
df_weratedog_complete.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1852 entries, 0 to 1851
Data columns (total 26 columns):
tweet_id                      1852 non-null int64
timestamp                     1852 non-null object
source                        1852 non-null object
text                          1852 non-null object
retweeted_status_id           0 non-null float64
retweeted_status_user_id      0 non-null float64
retweeted_status_timestamp    0 non-null float64
expanded_urls                 1852 non-null object
rating_numerator              1839 non-null float64
rating_denominator            1852 non-null int64
name                          1280 non-null object
stage                         312 non-null object
retweet_count                 1852 non-null int64
favorite_count                1852 non-null int64
rating                        1839 non-null float64
jpg_url                       1852 non-null object
img_num                       1852 non-null int64
p1                            1852 non-null object
p1_conf                       1852 non-null float64
p1_dog                        1852 non-null bool
p2                            1852 non-null object
p2_conf                       1852 non-null float64
p2_dog                        1852 non-null bool
p3                            1852 non-null object
p3_conf                       1852 non-null float64
p3_dog                        1852 non-null bool
dtypes: bool(3), float64(8), int64(5), object(10)
memory usage: 338.3+ KB
```

In [6]:
```
df_weratedog_complete.describe()
```

Out[6]:

	tweet_id	retweeted_status_id	retweeted_status_user_id	retweeted_status_timestamp	rating_numerator	rating_denominator	retweet_cou
count	1.852000e+03	0.0	0.0	0.0	1839.000000	1852.0	1852.0000
mean	7.334711e+17	NaN	NaN	NaN	20.357803	10.0	2294.7861
std	6.739552e+16	NaN	NaN	NaN	256.972715	0.0	3818.9515
min	6.660209e+17	NaN	NaN	NaN	0.000000	10.0	15.0000
25%	6.752923e+17	NaN	NaN	NaN	10.000000	10.0	593.7500
50%	7.059727e+17	NaN	NaN	NaN	11.000000	10.0	1234.0000
75%	7.826358e+17	NaN	NaN	NaN	12.000000	10.0	2660.5000
max	8.924206e+17	NaN	NaN	NaN	10712.000000	10.0	79116.0000

根据项目说明，除了rating_denominator以外，其他数值型数据都需要有极端值现象存在，而且这些极端值现象在真实的社交媒体中是真实存在的，所有不考清除极端值。

In [7]:
```
df_weratedog_complete.isnull().sum()
```

Out[7]:
```
tweet_id                      0
timestamp                     0
source                        0
text                          0
retweeted_status_id        1852
retweeted_status_user_id   1852
retweeted_status_timestamp 1852
expanded_urls                 0
rating_numerator             13
rating_denominator            0
name                        572
stage                      1540
retweet_count                 0
favorite_count                0
rating                       13
jpg_url                       0
img_num                       0
p1                            0
p1_conf                       0
p1_dog                        0
p2                            0
p2_conf                       0
p2_dog                        0
p3                            0
p3_conf                       0
p3_dog                        0
dtype: int64
```

retweeted_status_id ， retweeted_status_user_id和retweeted_status_timestamp存在大量null数据。 介于在分析过程中根本用不到任何跟retweet相关的数据，直接删除掉数据也不会对整个分析过程产生影响，所以直接删除这三列。

In [8]:
```
df_weratedog_complete.drop(['retweeted_status_id','retweeted_status_user_id','retweeted_status_timestamp'],ax
is=1,inplace=True)
```

In [9]:
```
df_weratedog_complete.dropna(subset=['rating'],inplace=True)
```

In [10]:
```
df_weratedog_complete.isnull().sum()
```

Out[10]:
```
tweet_id                 0
timestamp                0
source                   0
text                     0
expanded_urls            0
rating_numerator         0
rating_denominator       0
name                   561
stage                 1535
retweet_count            0
favorite_count           0
rating                   0
jpg_url                  0
img_num                  0
p1                       0
p1_conf                  0
p1_dog                   0
p2                       0
p2_conf                  0
p2_dog                   0
p3                       0
p3_conf                  0
p3_dog                   0
dtype: int64
```

In [11]:
```
df_weratedog_complete.info()
```

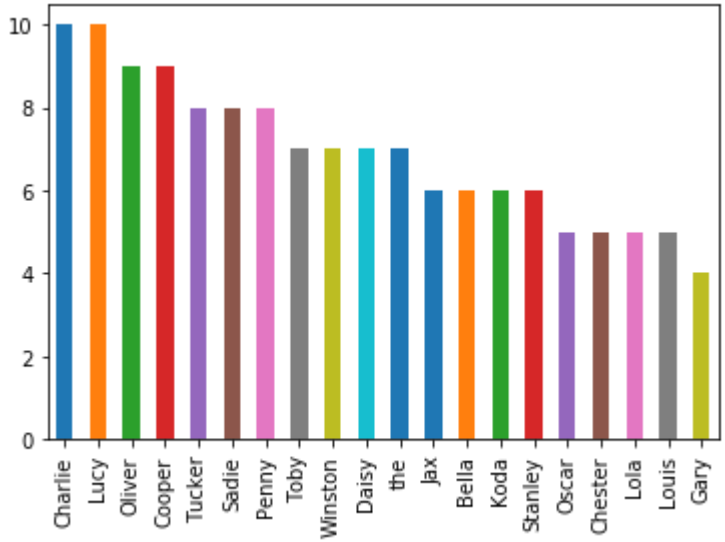
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1839 entries, 0 to 1851
Data columns (total 23 columns):
tweet_id              1839 non-null int64
timestamp             1839 non-null object
source                1839 non-null object
text                  1839 non-null object
expanded_urls         1839 non-null object
rating_numerator      1839 non-null float64
rating_denominator    1839 non-null int64
name                  1278 non-null object
stage                 304 non-null object
retweet_count         1839 non-null int64
favorite_count        1839 non-null int64
rating                1839 non-null float64
jpg_url               1839 non-null object
img_num               1839 non-null int64
p1                    1839 non-null object
p1_conf               1839 non-null float64
p1_dog                1839 non-null bool
p2                    1839 non-null object
p2_conf               1839 non-null float64
p2_dog                1839 non-null bool
p3                    1839 non-null object
p3_conf               1839 non-null float64
p3_dog                1839 non-null bool
dtypes: bool(3), float64(5), int64(5), object(10)
memory usage: 307.1+ KB
```

数据分析

由于狗的名字太多，我们不方便一一列举，只选择前20名最受欢迎的名字。

In [12]:
```
# 哪种名字比较受欢迎?
df_weratedog_complete[df_weratedog_complete['name'].notna()].name.value_counts().head(20).plot(kind='bar')
```

Out[12]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7feed21defd0>
```



受欢迎的名字数目大于5个，其中charile和Lucy最多，数目为10。

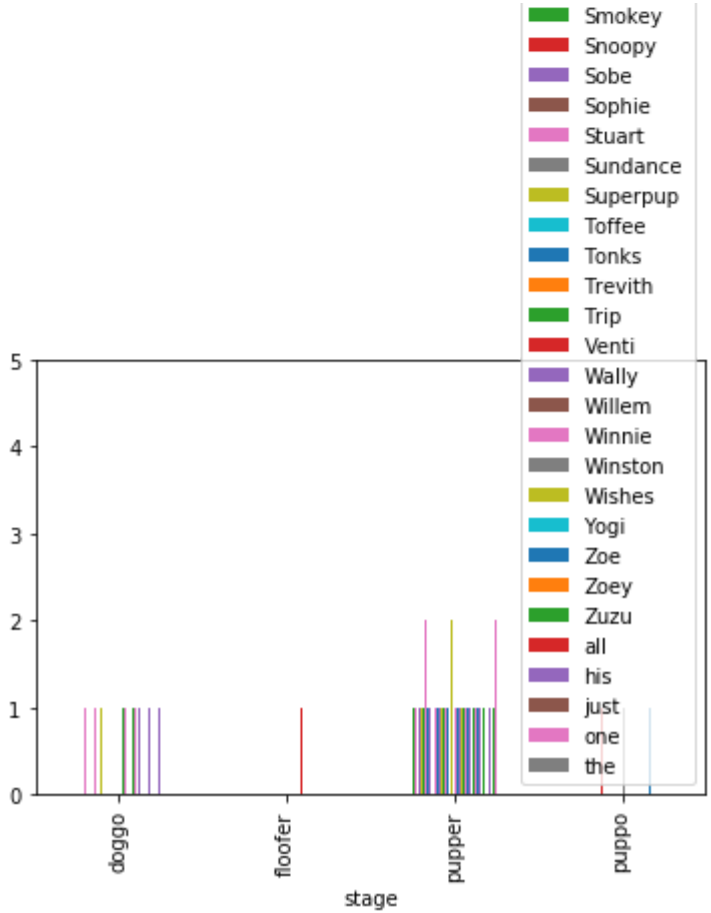
In [13]:
```
#不同种类狗起名字的偏好是什么?
temp = df_weratedog_complete[(df_weratedog_complete['name'].notna()) & (df_weratedog_complete['stage'].notna
())]

temp.groupby(["stage"]).name.value_counts().unstack().plot(ylim=(0,5),kind = 'bar')
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7feed010c978>
```


name
Abby
Adele
Albert
Ambrose
Aqua
Arlen
Ashleigh
Ava
Banjo
Barney
Bayley
Bella
Benji
Birf
Blakely
Blu
Bones
Boomer
Brandy
Bubba
Buckley
Cassie
Charlie
Cheryl
Chet
Chubbs
Chuckles
Clark
Clyde
Cooper
Craig
Cupid
Curtis
Deacon
Derek
Dido
Dietrich
Diogi
Divine
Django
Doc
Doobert
Duchess
Duke
Dwight
Edmund


Ellie
Emanuel
Finley
Finn
Finnegus
Fletcher
Gabe
Gary
Gerbald
Ginger
Gizmo
Godzilla
Grady
Grizzwald
Gus
Hamrick
Hector
Hubertson
Huck
Jax
Jazz
Jed
Jeffri
Kaiya
Kawhi
Kellogg
Kilo
Kona
Kyle
Laika
Larry
Lassie
Lennon
Lenox
Lili
Lillie
Lily
Lizzie
Loki
Lorenzo
Louie
Lucy
Luther
Maggie
Malcolm
Marty
Meera
Miguel

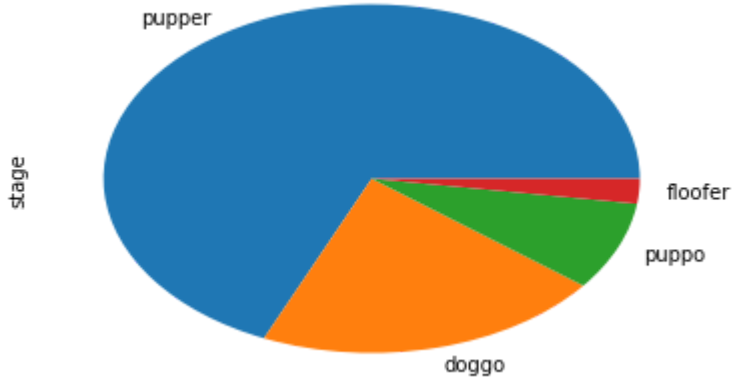

Millie
Milo
Mimosa
Misty
Mollie
Mona
Moose
Napolean
Norman
Obie
Oliver
Olivia
Ollie
Opal
Otis
Patrick
Penny
Pepper
Pete
Petrick
Phil
Pickles
Pilot
Pinot
Piper
Ralphson
Raymond
Rhino
Rinna
Rizzo
Rocky
Rodman
Rolf
Rooney
Rory
Roscoe
Rover
Rueben
Rufio
Sadie
Sansa
Schnozz
Scooter
Scott
Scout
Sebastian
Shikha
Smiley



鉴于狗的名字太多，无法通过条形图查看具体细节信息。打算换一种方法，可以采用宏观可视化展示结合程序化分析的方法展示具体信息。 先用饼状图展示狗的种类分类信息，通过饼状图我们发现，pupper的种类最多,其次是doggo，puppo和floofer.然后再通过程序化评估的方式分别查看不同种类狗的具体，命名情况。

In [14]:
```
df_weratedog_complete.stage.value_counts().plot(kind='pie')
```

Out[14]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7feecf819fd0>
```



In [15]:
```
df_weratedog_complete[df_weratedog_complete.stage=='pupper'].name.value_counts()
```

Out[15]:
```
Lennon      2
Gus         2
one         2
Pickles     2
Milo        2
Cooper      2
Chuckles    2
the         2
Lorenzo     1
Rodman      1
Ellie       1
Edmund      1
Superpup    1
Malcolm     1
Cheryl      1
Pepper      1
Rueben      1
Sophie      1
just        1
Opal        1
Hamrick     1
Ava         1
Sadie       1
Otis        1
Zuzu        1
Millie      1
Obie        1
Aqua        1
Buckley     1
Ginger      1
           ..
Ambrose     1
Patrick     1
Lillie      1
Gabe        1
Clark       1
Huck        1
Mona        1
Oliver      1
Kawhi       1
Brandy      1
```

```
Chet        1
Misty       1
Rory        1
Zoe         1
Norman      1
Gary        1
Luther      1
Ollie       1
Winnie      1
Banjo       1
Rooney      1
Gizmo       1
Dwight      1
Scott       1
Phil        1
Clyde       1
Kona        1
Zoey        1
Rufio       1
Winston     1
Name: name, Length: 114, dtype: int64
```

In [16]:
```
df_weratedog_complete[df_weratedog_complete.stage=='doggo'].name.value_counts()
```

Out[16]:
```
Finley      1
one         1
Dietrich    1
Divine      1
Miguel      1
Lenox       1
Bones       1
Meera       1
Piper       1
his         1
Duchess     1
Barney      1
Pinot       1
Emanuel     1
Chubbs      1
Cassie      1
Cupid       1
Pilot       1
Mimosa      1
Scout       1
Sundance    1
Rizzo       1
Pete        1
Rhino       1
Sobe        1
Yogi        1
Wishes      1
Deacon      1
Rocky       1
Maggie      1
Doobert     1
Kellogg     1
Smiley      1
Kyle        1
Napolean    1
Name: name, dtype: int64
```

In [17]:
```
df_weratedog_complete[df_weratedog_complete.stage=='puppo'].name.value_counts()
```

Out[17]:
```
Lili         1
Stuart       1
Lassie       1
Duke         1
Bayley       1
Snoopy       1
Lily         1
Tonks        1
Loki         1
Abby         1
Cooper       1
Sebastian    1
Kilo         1
Venti        1
Diogi        1
Lucy         1
Shikha       1
Name: name, dtype: int64
```

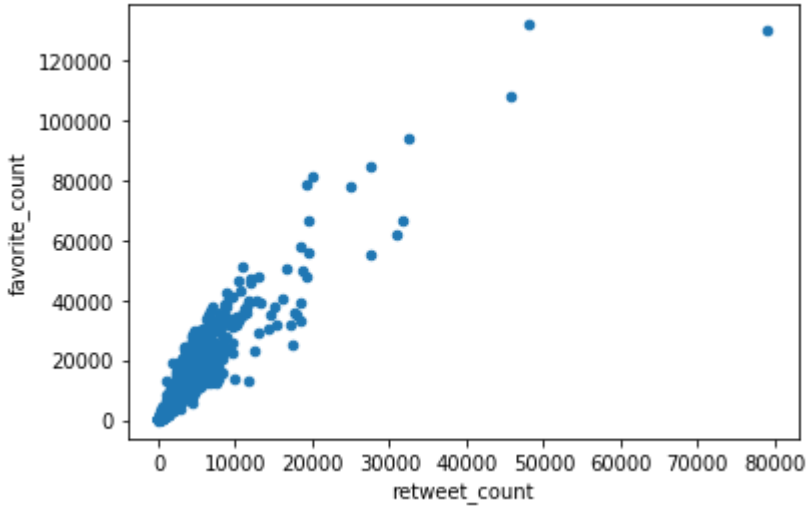
In [18]:
```
df_weratedog_complete[df_weratedog_complete.stage=='floofer'].name.value_counts()
```

Out[18]:
```
Blu          1
Petrick      1
Grizzwald    1
Doc          1
Moose        1
Name: name, dtype: int64
```

通过程序分析结果可以看出，没有没有哪个特定的名字集中于哪个种类中。

In [21]:
```
#高点赞代表高转发吗?
df_weratedog_complete.plot.scatter('retweet_count', 'favorite_count')
```

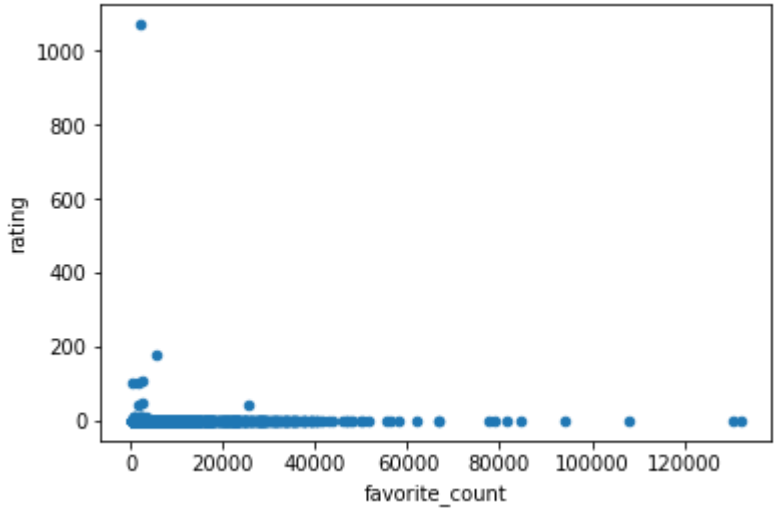
Out[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7feecff395c0>
```



retweet_count和favorite_count成很强的线性关系，可以同过肉眼直接判断。

In [22]:
```
# 受大众喜欢的狗评价越高?
df_weratedog_complete.plot.scatter('favorite_count','rating')
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x7feecf3bd2e8>



通过观察散点图，我们发现favorite_count和rating变量之间的相关性不大。从图中可以看出，斜率趋近于0.

得出答案

问题

1. 哪种名字比较受欢迎?
2. 不同种类狗起名字的偏好是什么?
3. 高点赞代表高转发吗?
4. 高点赞代表了专业好评吗?

答案

美国狗狗最常见的名为charile,Lucy,oliver,coopper等。名字大多情况下代表一个人性格和特质，但是通过分析狗种类与名字的关系，未发现特定种类的狗狗，出现一些使用频繁的名字。给狗命名无偏好体现。互联网时代点赞的成本非常低，远远低于转发的成本。研究和可视化发现点赞和转发的数据之前存在很强的线性关系，既点赞越多又转发越多。在宠物领域内，点赞和转发是成正比的。通过可视化研究发现，点赞和评分无实际关系，评分不会随着点赞数的增加而增加。

In [ ]: