```
2      None      ...             NaN     NaN               NaN       NaN
3      None      ...        0.787424    True  miniature_poodle  0.202225
4      None      ...        0.412893    True          Pembroke  0.312958

  p2_dog         p3   p3_conf p3_dog favorite_count  retweet_count
0   True     collie  0.000069   True            0.0           35.0
1    NaN        NaN       NaN    NaN            0.0           33.0
2    NaN        NaN       NaN    NaN         6028.0         2419.0
3   True      teddy  0.004047  False          345.0          134.0
4   True  Chihuahua  0.071960   True          903.0          555.0

[5 rows x 23 columns]
```

```
In [226]: twEnArch.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2356 entries, 2259 to 7236
Data columns (total 10 columns):
tweet_id                 2356 non-null int64
in_reply_to_status_id    2356 non-null object
in_reply_to_user_id      2356 non-null object
timestamp                2356 non-null datetime64[ns]
source                   2356 non-null object
text                     2356 non-null object
expanded_urls            2297 non-null object
rating_scale_10          2356 non-null int64
name                     2356 non-null object
dog_stage                2356 non-null object
dtypes: datetime64[ns](1), int64(2), object(7)
memory usage: 202.5+ KB
```

5 analyzing and visualization

```
In [227]: df.columns
```

```
Out[227]: Index(['tweet_id', 'in_reply_to_status_id', 'in_reply_to_user_id', 'timestamp',
                 'source', 'text', 'expanded_urls', 'rating_scale_10', 'name',
                 'dog_stage', 'jpg_url', 'img_num', 'p1', 'p1_conf', 'p1_dog', 'p2',
                 'p2_conf', 'p2_dog', 'p3', 'p3_conf', 'p3_dog', 'favorite_count',
                 'retweet_count'],
                dtype='object')
```

this are the most columns will be important for visualizations
timestamp
dog_stage
P's
counts

6 what is the most loved stage

In [ ]:

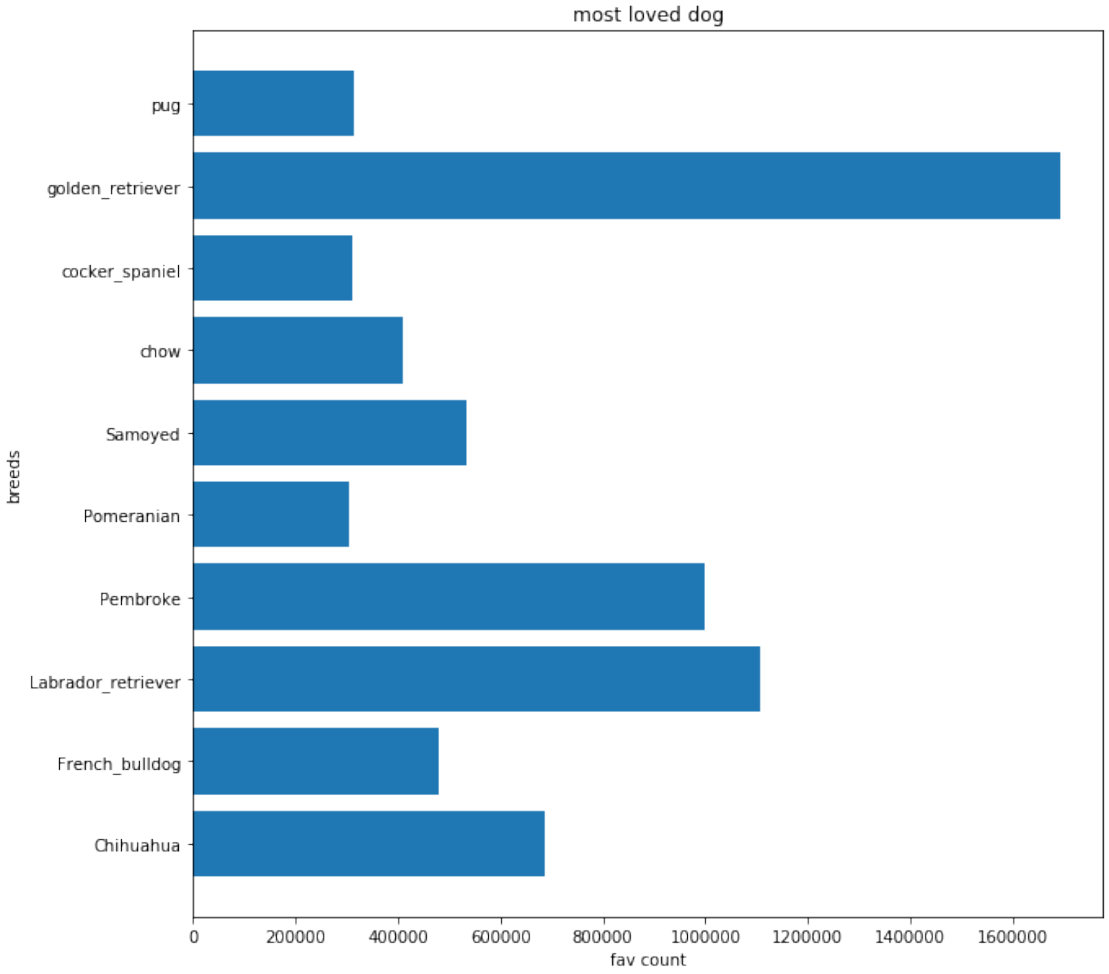
In [228]:
```
fav = df.groupby('p1')['favorite_count'].sum().reset_index()
fav= fav.sort_values('favorite_count',ascending=False).head(10)
fav
```

Out[228]:

	p1	favorite_count
124	golden_retriever	1691197.0
40	Labrador_retriever	1107532.0
53	Pembroke	1000119.0
19	Chihuahua	685212.0
59	Samoyed	535489.0
27	French_bulldog	478876.0
103	chow	409855.0
164	pug	315622.0
105	cocker_spaniel	311347.0
54	Pomeranian	304466.0

In [229]:
```
plt.figure(figsize=(10,10))
plt.barh(fav.p1, fav.favorite_count)
plt.title('most loved dog')
plt.xlabel('fav count')
plt.ylabel('breeds')
```

Out[229]:
```
Text(0,0.5,'breeds')
```



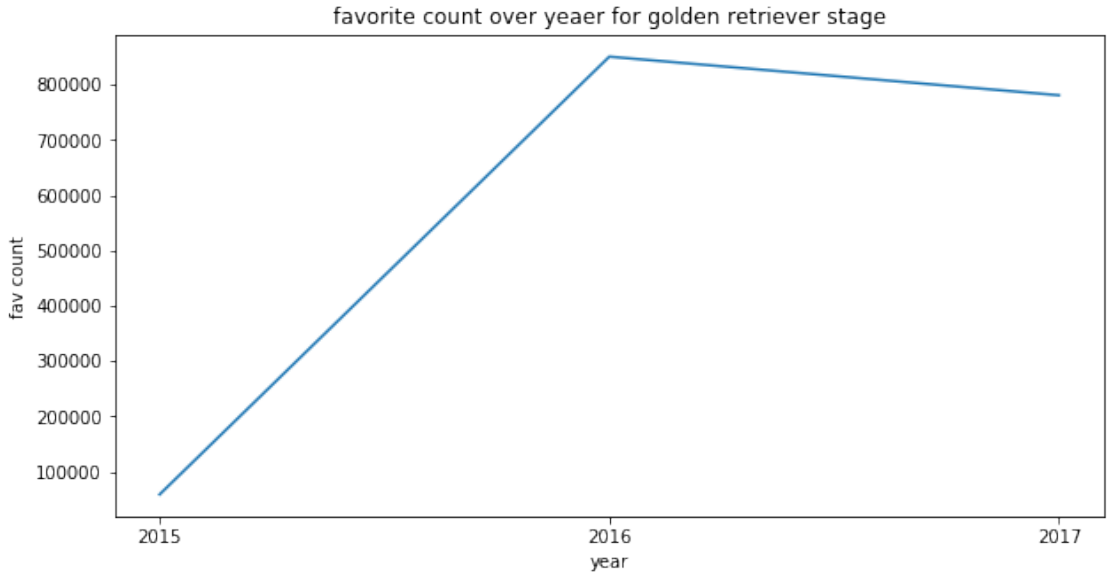
the golden_retriever stage is the most loved one and then the labrador_retriever then pembroke

7 what is the change of the golden_retriever fav count over time (year)

```
In [230]: df.timestamp = df.timestamp.astype('str')

          df['year'] = df.timestamp.apply(lambda x : x.split('-')[0])
          df['month'] = df.timestamp.apply(lambda x : x.split('-')[1])

          df.head()

Out[230]:             tweet_id in_reply_to_status_id in_reply_to_user_id  \
          0  667550904950915073                   nan                  nan
          1  667550882905632768                   nan                  nan
          2  667549055577362432                   nan                  nan
          3  667546741521195010                   nan                  nan
```

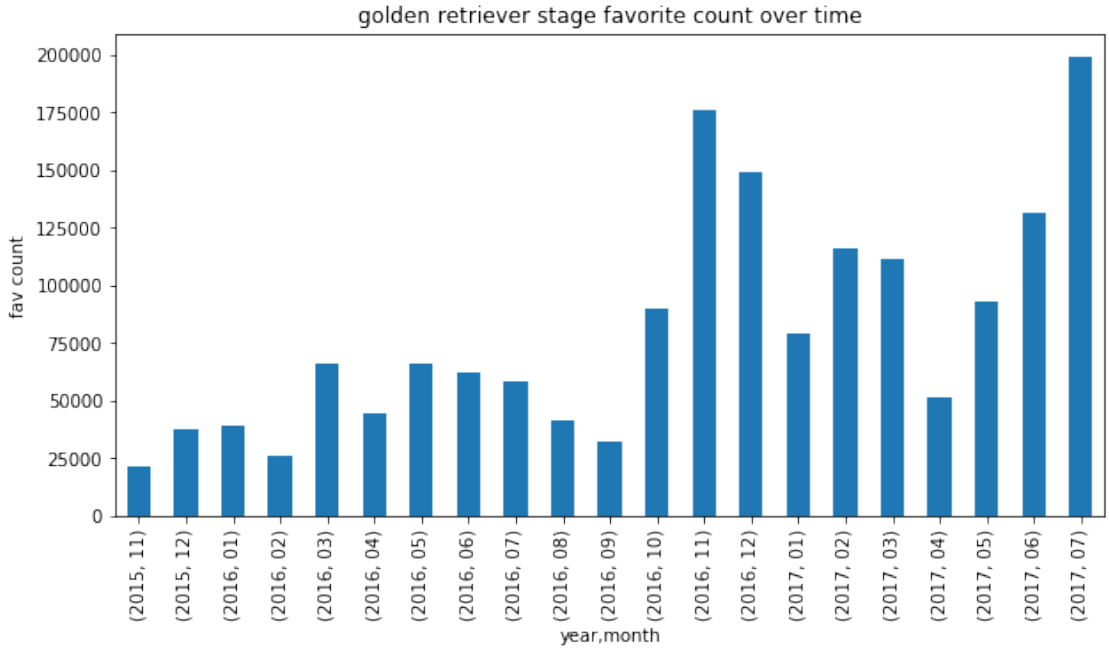
```
4  667544320556335104                   nan                  nan

             timestamp              source  \
0  2015-11-20 03:51:52  Twitter Web Client
1  2015-11-20 03:51:47  Twitter Web Client
2  2015-11-20 03:44:31  Twitter Web Client
3  2015-11-20 03:35:20  Twitter Web Client
4  2015-11-20 03:25:43  Twitter Web Client

                                                text  \
0  RT @dogratingrating: Exceptional talent. Origi...
1  RT @dogratingrating: Unoriginal idea. Blatant ...
2  Never seen dog like this. Breathes heavy. Tilt...
3  Here is George. George took a selfie of his ne...
4  This is Kial. Kial is either wearing a cape, w...

                                       expanded_urls  rating_scale_10  name  \
0  https://twitter.com/dogratingrating/status/667...               12   NaN
1  https://twitter.com/dogratingrating/status/667...                5   NaN
2  https://twitter.com/dog_rates/status/667549055...                1   NaN
3  https://twitter.com/dog_rates/status/667546741...                9   NaN
4  https://twitter.com/dog_rates/status/667544320...               10  Kial

  dog_stage  ...                  p2   p2_conf p2_dog         p3   p3_conf  \
0      None  ...              vizsla  0.000081   True     collie  0.000069
1      None  ...                 NaN       NaN    NaN        NaN       NaN
2      None  ...                 NaN       NaN    NaN        NaN       NaN
3      None  ...    miniature_poodle  0.202225   True      teddy  0.004047
4      None  ...           Pembroke  0.312958   True  Chihuahua  0.071960

  p3_dog  favorite_count retweet_count  year  month
0   True             0.0          35.0  2015     11
1    NaN             0.0          33.0  2015     11
2    NaN          6028.0        2419.0  2015     11
3  False           345.0         134.0  2015     11
4   True           903.0         555.0  2015     11

[5 rows x 25 columns]
```

In [231]:
```
temp = df.groupby(['year','p1']).sum().favorite_count.reset_index()
temp = temp.query('p1 == "golden_retriever"')

plt.figure(figsize=(10,5))
plt.plot(temp.year, temp.favorite_count);
plt.title('favorite count over yeaer for golden retriever stage');
plt.xlabel('year');
plt.ylabel('fav count');
```



fav count increased by 800000 from 2015 to 2016 then decreased a littel bit

```
In [232]: temp = df.query('p1 == "golden_retriever"')
          temp = temp.groupby(['year','month']).favorite_count.sum()
          temp
          ax = temp.plot(kind='bar',title='golden retriever stage favorite count over time',figs

          ax.set_ylabel('fav count')

Out[232]: Text(0,0.5,'fav count')
```



the fav count tend to be increase over time
drop year and month columns

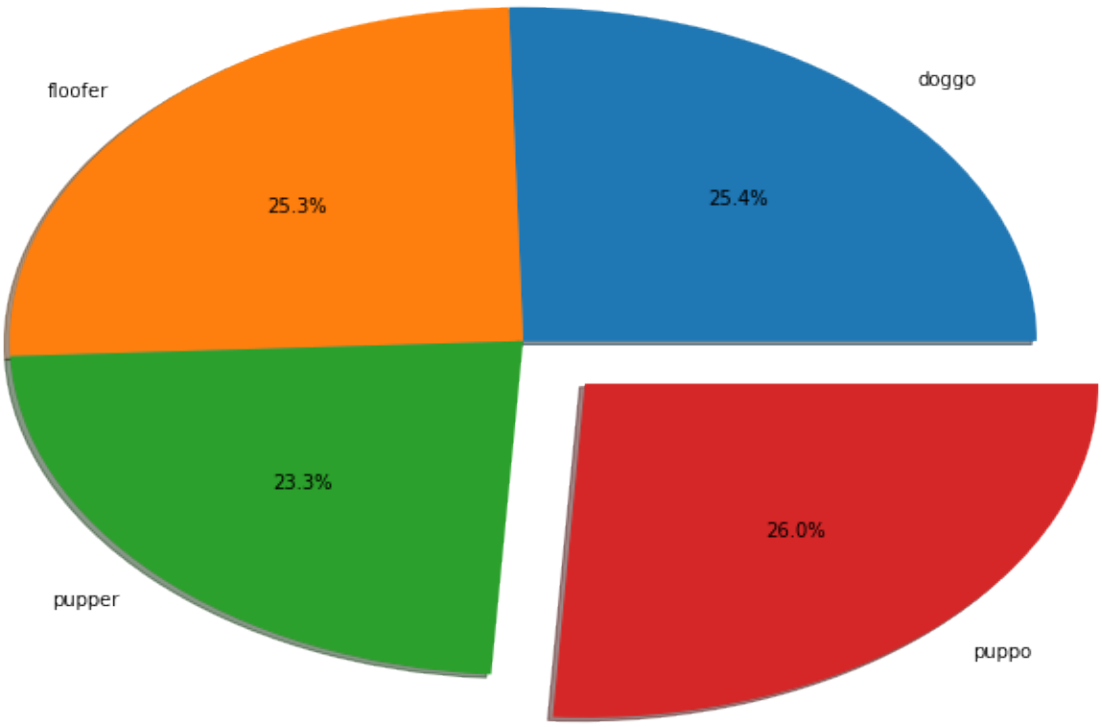
```
In [233]: df = df.drop(['year','month'],axis=1)
```

8 dog stages and its rating

```
In [234]: temp = df.query('dog_stage != "None"')
          temp = temp.groupby('dog_stage').rating_scale_10.mean().reset_index()
          temp
```

```
Out[234]:   dog_stage  rating_scale_10
          0     doggo        11.879518
          1   floofer        11.800000
          2    pupper        10.871595
          3     puppo        12.133333
```

```
In [235]: plt.pie(temp.rating_scale_10,labels=temp.dog_stage,radius=2,autopct='%0.1f%%',explode=
```



puppo has the largest rating by 26.0