Explore preprocessed Kickstarter data

- plot basic relations, distributions, etc.

In [1]:
```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
%matplotlib inline
```

In [2]:
```
target_path = '../data/interim/kickstarter_csvs'
filename = 'kick_id.csv'
```

In [3]:
```
datecols = ['created_at', 'deadline', 'state_changed_at', 'launched_at']
```

In [4]:
```
fdatpars = lambda x: datetime.datetime.fromtimestamp(int(x)).strftime('%Y-%m-%d %H:%M:%S')
```

In [5]:
```
df = pd.read_csv(os.path.join(target_path, filename), index_col='id', parse_dates=datecols, dat
e_parser=fdatpars)
```

In [6]:
```
df.head(3)
```

Out[6]:

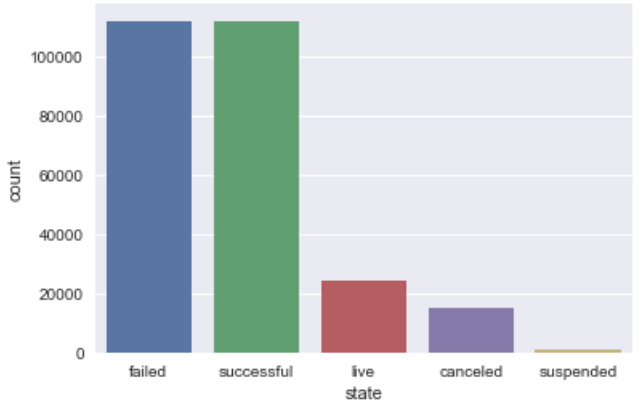
	name	goal	pledged	usd_pledged	state	slug	disable_communication	country	currency
id									
18520	Grandma's are Life	15000.0	62.0	62.000000	failed	grandmas-are-life	False	US	USD
21109	Meta	150.0	173.0	258.036032	successful	meta	False	GB	GBP
24380	Puss N' Books: A relaxing cat cafe and bookstore.	20000.0	776.0	776.000000	failed	puss-n-books-a-relaxing-cat-cafe-and-bookstore	False	US	USD

In [7]:
```
df.dtypes
```

Out[7]:
```
name                             object
goal                            float64
pledged                         float64
usd_pledged                     float64
state                            object
slug                             object
disable_communication              bool
country                          object
currency                         object
deadline                 datetime64[ns]
state_changed_at         datetime64[ns]
created_at               datetime64[ns]
launched_at              datetime64[ns]
staff_pick                         bool
backers_count                     int64
blurb                            object
spotlight                          bool
category                         object
dtype: object
```

In [8]:
```
sns.countplot(x='state', data=df)
```

Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x120acb908>
```



In [9]:
```
from collections import Counter
Counter(df['state'])
```

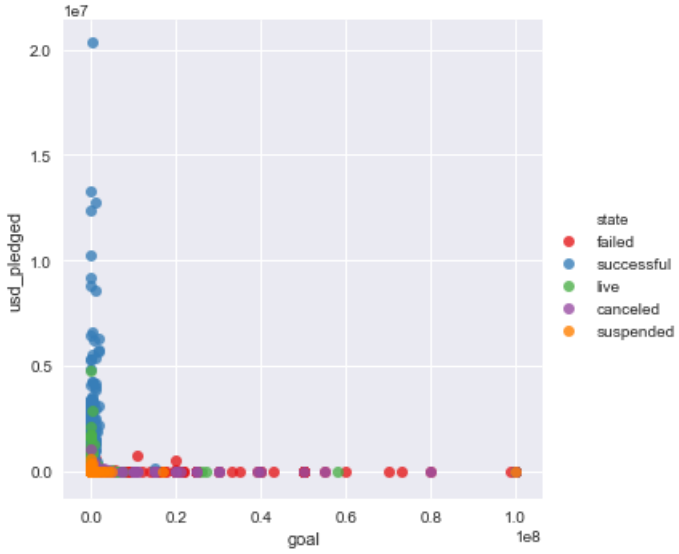
Out[9]:
```
Counter({'canceled': 15021,
         'failed': 111621,
         'live': 24357,
         'successful': 111814,
         'suspended': 952})
```

- About half the projects were successful (the funding process)
- This does not seem to match the info I got before (~34%) - why?
- I may have deleted resubmittals during preprocessing ??? - not likely, but look into it

Goals vs. pledged

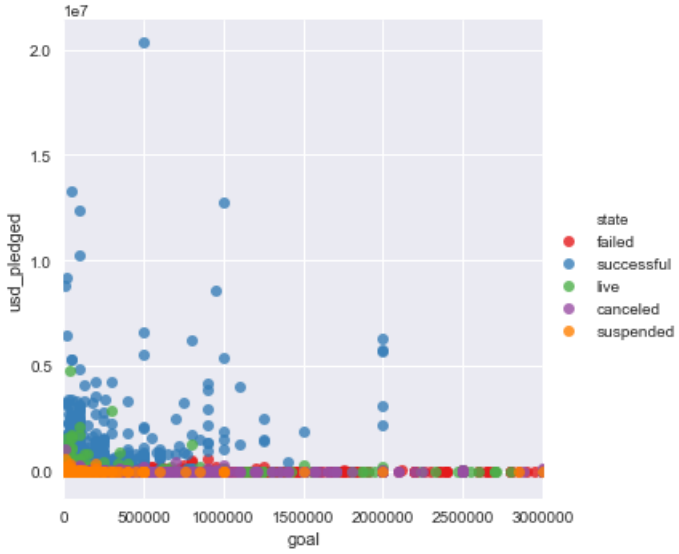
In [10]:
```
sns.lmplot(x='goal', y='usd_pledged', hue='state', data=df, fit_reg=False, palette='Set1')
```

Out[10]:
```
<seaborn.axisgrid.FacetGrid at 0x10fec9e80>
```



In [11]:
```
sns.lmplot(x='goal', y='usd_pledged', hue='state', data=df, fit_reg=False, palette='Set1')
plt.xlim(0,0.3e7)
```

Out[11]:
```
(0, 3000000.0)
```



- There seems to be inverse relationship between the goal and amount pledged
- Successful funding campain should be easier with lower target, but asking less also helps to get more money
- Lowballing a good strategy?
  - May get staff endorsment
  - Better perception of achievability
  - Large funding goals scare people away
  - Large funding goals may be correlate with poor planning or crackpot ideas?

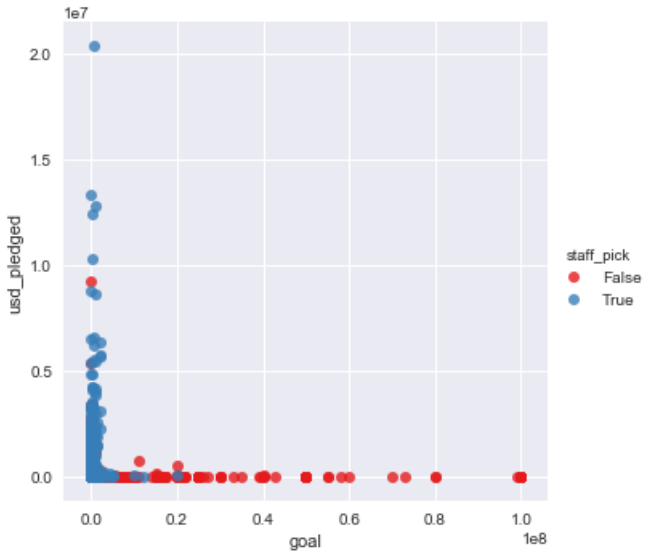
Staff picks - are they any good?

In [12]:
```
Counter(df['staff_pick'])
```

Out[12]:
```
Counter({False: 234750, True: 29015})
```

In [13]:
```
sns.lmplot(x='goal', y='usd_pledged', hue='staff_pick', data=df, fit_reg=False, palette='Set1')
```

Out[13]:
```
<seaborn.axisgrid.FacetGrid at 0x120ad1550>
```



- Staff picks are usually a good predictor of success
- A few outliers, when they missed
- Any time the staff picked higher goal (above 0.1e8) they missed.
- Better stick to small goals
- Is the staff_pick info redundant (use just the goal)?

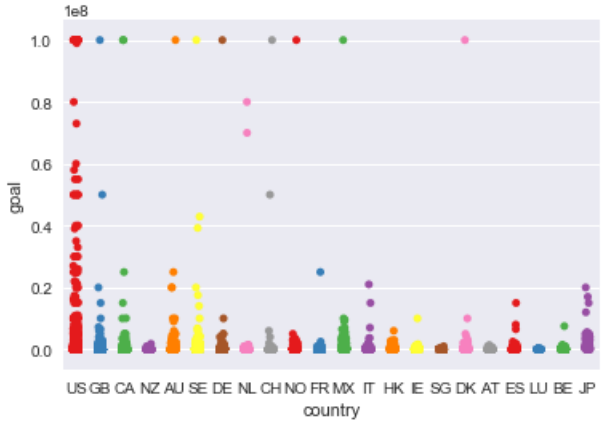
Countries comparison - how many projects, how much they ask and get

In [14]:
```
import operator
sorted(Counter(df['country']).items(), key=operator.itemgetter(1), reverse=True)
```

Out[14]:
```
[('US', 203618),
 ('GB', 23662),
 ('CA', 10583),
 ('AU', 5637),
 ('DE', 3140),
 ('FR', 2332),
 ('IT', 2248),
 ('NL', 2081),
 ('ES', 1837),
 ('MX', 1518),
 ('SE', 1340),
 ('NZ', 1047),
 ('DK', 866),
 ('IE', 616),
 ('CH', 602),
 ('HK', 578),
 ('NO', 542),
 ('BE', 497),
 ('SG', 455),
 ('AT', 445),
 ('JP', 77),
 ('LU', 44)]
```

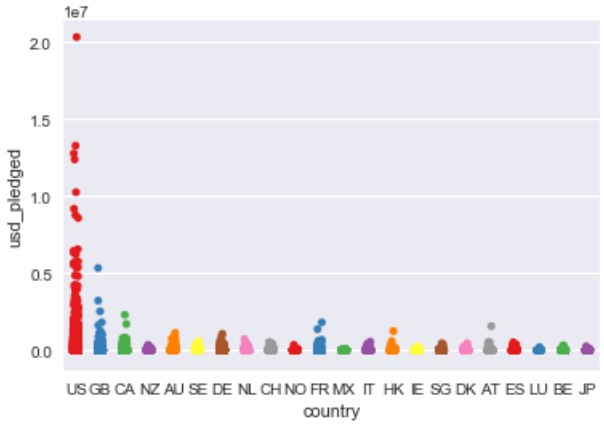
In [15]:
```
sns.stripplot(x='country', y='goal', data=df, palette='Set1', jitter=True)
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x11b4c92b0>



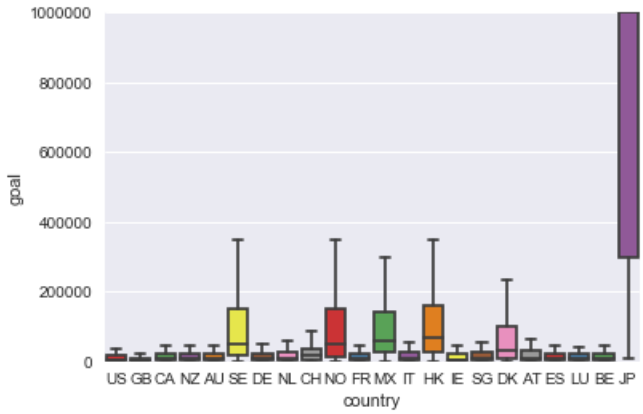
In [16]:
```
sns.stripplot(x='country', y='usd_pledged', data=df, palette='Set1', jitter=True)
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x1145f0160>



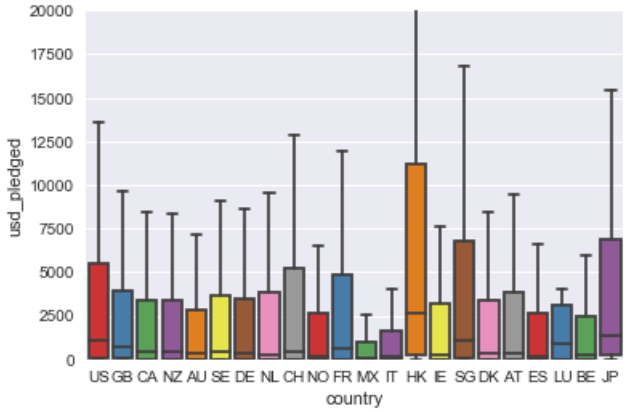
In [17]:
```
sns.boxplot(x='country', y='goal', data=df, palette='Set1', fliersize=0)
plt.ylim(0,1e6)
```

Out[17]: (0, 1000000.0)



In [18]:
```
sns.boxplot(x='country', y='usd_pledged', data=df, palette='Set1', fliersize=0)
plt.ylim(0,2e4)
```

Out[18]: (0, 20000.0)



- 22 countries
- Hong Kong asks a lot and gets a lot
- Japanese ask the most, get good funding (could there be some currency mess-up with Japanese Yen?)
- Mexicans ask a lot, get the least
- Americans don't ask much, but still are funded well

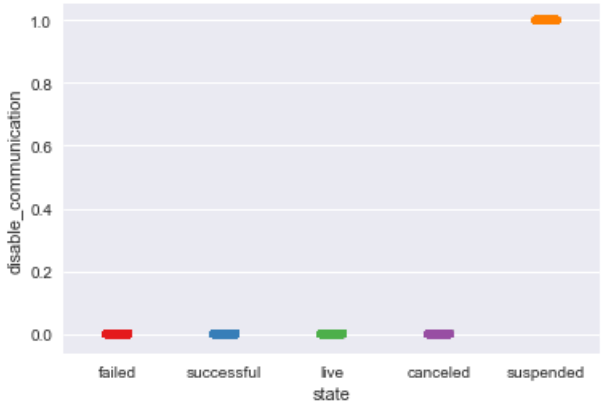
What is 'disable_communication'?

In [19]:
```
Counter(df['disable_communication'])
```

Out[19]: Counter({False: 262813, True: 952})

In [20]:
```
sns.stripplot(y='disable_communication', x='state', data=df, palette='Set1', jitter=True)
```

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x1170fbb38>



- It just happens when a project is suspended