# Startup Success Prediction

In [6]:
```
#importing the Dependinces
import pandas as pd
import numpy as np
import matplotlib as mpl
from matplotlib import pyplot as plt
import seaborn as sns
from datetime import date
from scipy import stats
from scipy.stats import norm, skew #for some statistics

import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff
from plotly.colors import n_colors
from plotly.subplots import make_subplots

import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
pd.set_option("display.max_columns",None)
pd.set_option("display.max_rows",None)
plt.style.use('seaborn')

from collections import Counter
import datetime
import wordcloud
import json
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
import shapefile as shp
import sys
```

In [7]:
```
#Loading the dataset in pandas dataframe
data = pd.read_csv('/content/startup data.csv')
```

In [8]:
```
#check first five rows of the dataset
data.head()
```

Out[8]:

| | Unnamed: 0 | state_code | latitude | longitude | zip_code | id | city | Unnamed: 6 |
|---|---|---|---|---|---|---|---|---|
| **0** | 1005 | CA | 42.358880 | -71.056820 | 92101 | c:6669 | San Diego | NaN |
| **1** | 204 | CA | 37.238916 | -121.973718 | 95032 | c:16283 | Los Gatos | NaN |
| **2** | 1001 | CA | 32.901049 | -117.192656 | 92121 | c:65620 | San Diego | San Diego CA 92121 |
| **3** | 738 | CA | 37.320309 | -122.050040 | 95014 | c:42668 | Cupertino | Cupertino CA 95014 |
| **4** | 1002 | CA | 37.779281 | -122.419236 | 94105 | c:65806 | San Francisco | San Francisco CA 94105 |

In [9]:
```
#check last five rows of the dataset
data.tail()
```

Out[9]:

| | Unnamed: 0 | state_code | latitude | longitude | zip_code | id | city | Unname |
|---|---|---|---|---|---|---|---|---|
| **918** | 352 | CA | 37.740594 | -122.376471 | 94107 | c:21343 | San Francisco | Na |
| **919** | 721 | MA | 42.504817 | -71.195611 | 1803 | c:41747 | Burlington | Burlingto MA 18( |
| **920** | 557 | CA | 37.408261 | -122.015920 | 94089 | c:31549 | Sunnyvale | Na |
| **921** | 589 | CA | 37.556732 | -122.288378 | 94404 | c:33198 | San Francisco | Na |
| **922** | 462 | CA | 37.386778 | -121.966277 | 95054 | c:26702 | Santa Clara | San Clara C 950! |

In [10]:
```
#check shape of the dataset
data.shape
```

Out[10]: (923, 49)

In [11]:
```
#check columns of the dataset
data.columns
```

Out[11]:
```
Index(['Unnamed: 0', 'state_code', 'latitude', 'longitude', 'zip_c
ode', 'id',
       'city', 'Unnamed: 6', 'name', 'labels', 'founded_at', 'clos
ed_at',
       'first_funding_at', 'last_funding_at', 'age_first_funding_y
ear',
       'age_last_funding_year', 'age_first_milestone_year',
       'age_last_milestone_year', 'relationships', 'funding_rounds
',
       'funding_total_usd', 'milestones', 'state_code.1', 'is_CA',
'is_NY',
       'is_MA', 'is_TX', 'is_otherstate', 'category_code', 'is_sof
tware',
       'is_web', 'is_mobile', 'is_enterprise', 'is_advertising',
       'is_gamesvideo', 'is_ecommerce', 'is_biotech', 'is_consulti
ng',
       'is_othercategory', 'object_id', 'has_VC', 'has_angel', 'ha
s_roundA',
       'has_roundB', 'has_roundC', 'has_roundD', 'avg_participants
',
       'is_top500', 'status'],
      dtype='object')
```

In [12]:
```
#check more infomation of the dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 923 entries, 0 to 922
Data columns (total 49 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Unnamed: 0                923 non-null    int64
 1   state_code                923 non-null    object
 2   latitude                  923 non-null    float64
 3   longitude                 923 non-null    float64
 4   zip_code                  923 non-null    object
 5   id                        923 non-null    object
 6   city                      923 non-null    object
 7   Unnamed: 6                430 non-null    object
 8   name                      923 non-null    object
 9   labels                    923 non-null    int64
 10  founded_at                923 non-null    object
 11  closed_at                 335 non-null    object
 12  first_funding_at          923 non-null    object
 13  last_funding_at           923 non-null    object
 14  age_first_funding_year    923 non-null    float64
 15  age_last_funding_year     923 non-null    float64
 16  age_first_milestone_year  771 non-null    float64
 17  age_last_milestone_year   771 non-null    float64
 18  relationships             923 non-null    int64
```

```
 19  funding_rounds             923 non-null    int64
 20  funding_total_usd          923 non-null    int64
 21  milestones                 923 non-null    int64
 22  state_code.1               922 non-null    object
 23  is_CA                      923 non-null    int64
 24  is_NY                      923 non-null    int64
 25  is_MA                      923 non-null    int64
 26  is_TX                      923 non-null    int64
 27  is_otherstate              923 non-null    int64
 28  category_code              923 non-null    object
 29  is_software                923 non-null    int64
 30  is_web                     923 non-null    int64
 31  is_mobile                  923 non-null    int64
 32  is_enterprise              923 non-null    int64
 33  is_advertising             923 non-null    int64
 34  is_gamesvideo              923 non-null    int64
 35  is_ecommerce               923 non-null    int64
 36  is_biotech                 923 non-null    int64
 37  is_consulting              923 non-null    int64
 38  is_othercategory           923 non-null    int64
 39  object_id                  923 non-null    object
 40  has_VC                     923 non-null    int64
 41  has_angel                  923 non-null    int64
 42  has_roundA                 923 non-null    int64
 43  has_roundB                 923 non-null    int64
 44  has_roundC                 923 non-null    int64
 45  has_roundD                 923 non-null    int64
 46  avg_participants           923 non-null    float64
 47  is_top500                  923 non-null    int64
 48  status                     923 non-null    object
dtypes: float64(7), int64(28), object(14)
memory usage: 353.5+ KB
```

In [13]:

```
#check mathamtic describe
data.describe()
```

Out[13]:

| | Unnamed: 0 | latitude | longitude | labels | age_first_funding_year | age_last_fu |
|---|---|---|---|---|---|---|
| **count** | 923.000000 | 923.000000 | 923.000000 | 923.000000 | 923.000000 | |
| **mean** | 572.297941 | 38.517442 | -103.539212 | 0.646804 | 2.235630 | |
| **std** | 333.585431 | 3.741497 | 22.394167 | 0.478222 | 2.510449 | |
| **min** | 1.000000 | 25.752358 | -122.756956 | 0.000000 | -9.046600 | |
| **25%** | 283.500000 | 37.388869 | -122.198732 | 0.000000 | 0.576700 | |
| **50%** | 577.000000 | 37.779281 | -118.374037 | 1.000000 | 1.446600 | |
| **75%** | 866.500000 | 40.730646 | -77.214731 | 1.000000 | 3.575350 | |
| **max** | 1153.000000 | 59.335232 | 18.057121 | 1.000000 | 21.895900 | |

In [14]:

```
#check corr relation of the dataset
data.corr()
```

Out[14]:

| | Unnamed: 0 | latitude | longitude | labels | age_first_funding_year |
|---|---|---|---|---|---|
| **Unnamed: 0** | 1.000000 | 0.054726 | 0.023292 | -0.068721 | -0.004507 |
| **latitude** | 0.054726 | 1.000000 | 0.368475 | 0.046560 | -0.046868 |
| **longitude** | 0.023292 | 0.368475 | 1.000000 | -0.036092 | -0.014158 |
| **labels** | -0.068721 | 0.046560 | -0.036092 | 1.000000 | -0.075637 |
| **age_first_funding_year** | -0.004507 | -0.046868 | -0.014158 | -0.075637 | 1.000000 |
| **age_last_funding_year** | -0.116533 | -0.041692 | -0.000148 | 0.073731 | 0.762382 |
| **age_first_milestone_year** | -0.135614 | -0.072000 | -0.051674 | 0.162279 | 0.593526 |
| **age_last_milestone_year** | -0.131698 | -0.054275 | -0.087701 | 0.265871 | 0.472029 |
| **relationships** | -0.079950 | -0.039198 | -0.073197 | 0.360434 | -0.187817 |
| **funding_rounds** | -0.118456 | -0.000659 | 0.022447 | 0.206049 | -0.155478 |
| **funding_total_usd** | -0.064169 | -0.072941 | 0.017970 | 0.040176 | 0.046350 |
| **milestones** | -0.000338 | 0.017708 | -0.016420 | 0.328260 | -0.295894 |
| **is_CA** | -0.042446 | -0.417471 | -0.780122 | 0.077217 | -0.010800 |
| **is_NY** | 0.033485 | 0.205747 | 0.449871 | 0.059996 | -0.128102 |
| **is_MA** | 0.043021 | 0.318015 | 0.441031 | 0.081735 | 0.020279 |
| **is_TX** | -0.021463 | -0.423888 | 0.066199 | -0.045309 | 0.032838 |
| **is_otherstate** | 0.002249 | 0.338590 | 0.257801 | -0.169067 | 0.081031 |
| **is_software** | 0.001367 | -0.001656 | 0.024857 | 0.012429 | 0.116797 |
| **is_web** | 0.007076 | -0.009799 | -0.022024 | -0.000873 | -0.166601 |
| **is_mobile** | -0.028279 | 0.035917 | 0.013527 | 0.007312 | -0.054658 |
| **is_enterprise** | 0.042640 | -0.002291 | -0.003244 | 0.073772 | -0.047326 |
| **is_advertising** | -0.075131 | 0.054575 | 0.039998 | 0.044355 | -0.071336 |
| **is_gamesvideo** | 0.065020 | -0.033160 | -0.025569 | -0.025893 | -0.063787 |
| **is_ecommerce** | -0.026132 | 0.041628 | 0.043092 | -0.072193 | -0.071580 |
| **is_biotech** | 0.004224 | 0.012956 | 0.028075 | 0.000104 | 0.190653 |
| **is_consulting** | -0.040929 | -0.033905 | 0.021244 | 0.002373 | -0.012596 |
| **is_othercategory** | 0.006243 | -0.039656 | -0.046560 | -0.042408 | 0.115649 |
| **has_VC** | -0.040057 | 0.031045 | 0.024852 | -0.056515 | 0.168140 |
| **has_angel** | 0.134044 | 0.028891 | 0.102001 | -0.072840 | -0.345985 |
| **has_roundA** | -0.076568 | -0.033072 | -0.066288 | 0.184307 | -0.293081 |

| | | | | | |
|---|---|---|---|---|---|
| **has_roundB** | -0.135289 | -0.011801 | -0.067017 | 0.208257 | -0.060532 |
| **has_roundC** | -0.090922 | -0.057762 | -0.042309 | 0.165902 | 0.033388 |
| **has_roundD** | -0.081123 | -0.018825 | -0.042854 | 0.139940 | 0.121338 |
| **avg_participants** | 0.026713 | -0.018176 | -0.045191 | 0.185992 | 0.114363 |
| **is_top500** | 0.026019 | 0.032675 | -0.091913 | 0.310652 | 0.050638 |

In [15]:
```
#check missing value of the dataset
data.isnull().sum()
```

Out[15]:
```
Unnamed: 0                    0
state_code                    0
latitude                      0
longitude                     0
zip_code                      0
id                            0
city                          0
Unnamed: 6                  493
name                          0
labels                        0
founded_at                    0
closed_at                   588
first_funding_at              0
last_funding_at               0
age_first_funding_year        0
age_last_funding_year         0
age_first_milestone_year    152
age_last_milestone_year     152
relationships                 0
funding_rounds                0
funding_total_usd             0
milestones                    0
state_code.1                  1
is_CA                         0
is_NY                         0
is_MA                         0
is_TX                         0
is_otherstate                 0
category_code                 0
is_software                   0
is_web                        0
is_mobile                     0
is_enterprise                 0
is_advertising                0
is_gamesvideo                 0
is_ecommerce                  0
is_biotech                    0
is_consulting                 0
is_othercategory              0
object_id                     0
```

```
has_VC                      0
has_angel                   0
has_roundA                  0
has_roundB                  0
has_roundC                  0
has_roundD                  0
avg_participants            0
is_top500                   0
status                      0
dtype: int64
```

In [16]:
```
# Total Missing Values kolom "Unnamed: 6"
totalNull = data['Unnamed: 6'].isnull().sum()

print('Total Missing Values Kolom "Unnamed: 6": ', totalNull)
```

```
Total Missing Values Kolom "Unnamed: 6":  493
```

In [17]:
```
#Handling Missing Value closed_at
data['closed_at'] = data['closed_at'].fillna(value="31/12/2013")
```

In [18]:
```
totalNull = data['closed_at'].isnull().sum()

print('Total Missing Values Kolom "closed_at": ', totalNull)
```

```
Total Missing Values Kolom "closed_at":  0
```

In [19]:
```
#Handling Missing Value age_first_milestone_year and age_last_miles
data[['age_first_milestone_year','age_last_milestone_year','milesto
```

Out[19]:

| | age_first_milestone_year | age_last_milestone_year | milestones |
|---|---|---|---|
| **0** | 4.6685 | 6.7041 | 3 |
| **1** | 7.0055 | 7.0055 | 1 |
| **2** | 1.4575 | 2.2055 | 2 |
| **3** | 6.0027 | 6.0027 | 1 |
| **4** | 0.0384 | 0.0384 | 1 |

In [20]:
```
data['age_first_milestone_year'] = data['age_first_milestone_year']
data['age_last_milestone_year'] = data['age_last_milestone_year'].f
```

In [21]:
```
#Handling Missing Value state_code.1
for index, row in data.iterrows():
    if row['state_code']!=row['state_code.1']:
        print(index, row['state_code'], row['state_code.1'])
```

```
515 CA nan
```

In [22]:
```
data.drop(["state_code.1"], axis=1, inplace=True)
```

In [23]:
```
null=pd.DataFrame(data.isnull().sum(),columns=["Null Values"])
null["% Missing Values"]=(data.isna().sum()/len(data)*100)
null = null[null["% Missing Values"] > 0]
null.style.background_gradient(cmap='viridis',low =0.2,high=0.1)
```

Out[23]:

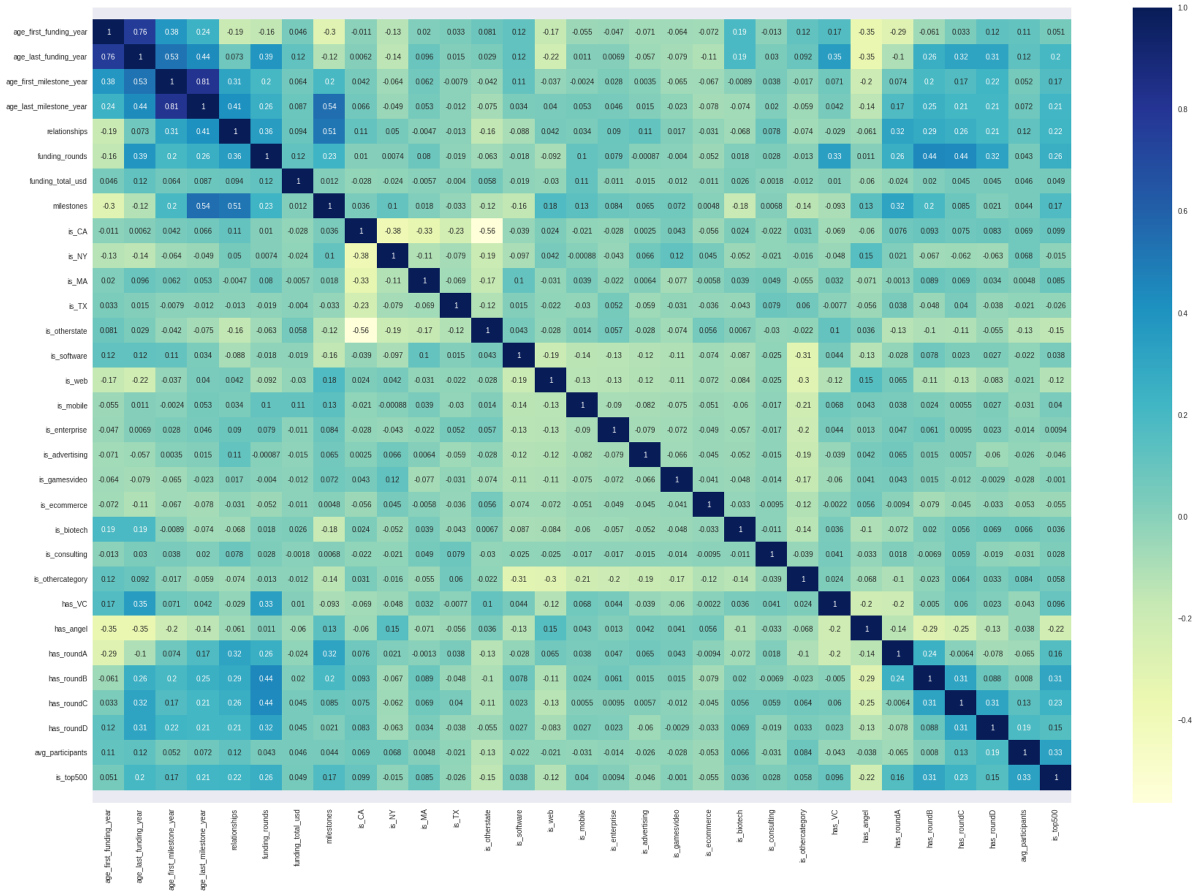
| | Null Values | % Missing Values |
|---|---|---|
| **Unnamed: 6** | 493 | 53.412784 |

# Graphic Approach

In [24]:
```
#Correlation heatmap
data['age_first_milestone_year'] = data.age_first_milestone_year.as
data['age_last_milestone_year'] = data.age_last_milestone_year.asty
```

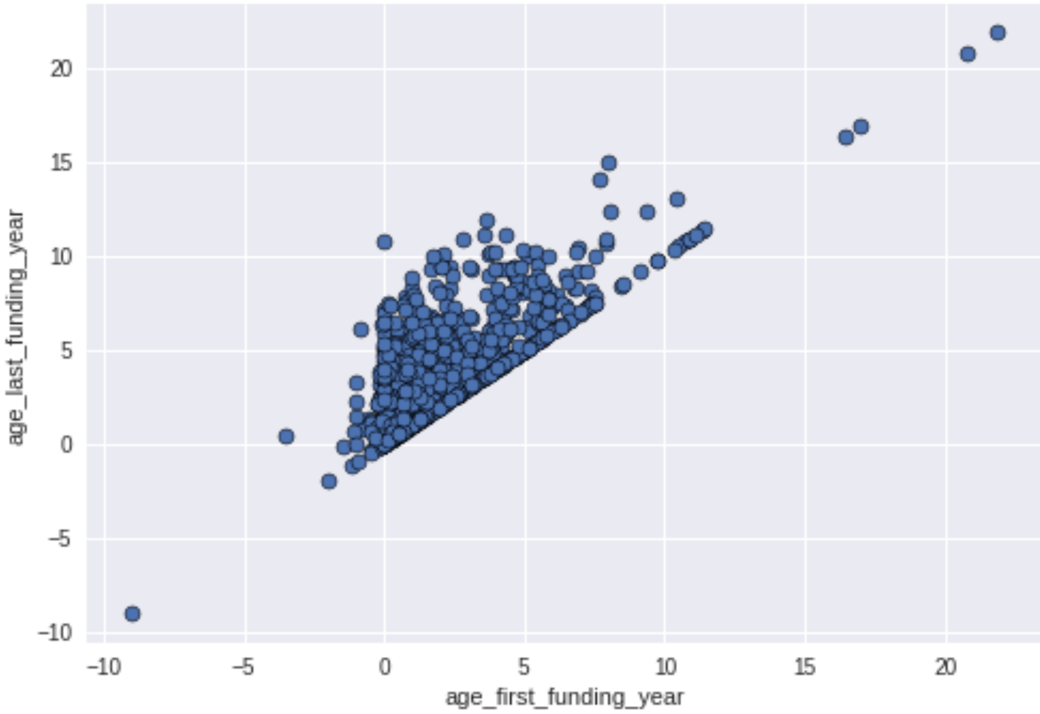
In [25]:
```
features = ['age_first_funding_year','age_last_funding_year','age_f

plt.figure(figsize=(30,20))
ax = sns.heatmap(data = data[features].corr(),cmap='YlGnBu',annot=T

bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5,top - 0.5)
```

Out[25]: (31.5, -0.5)

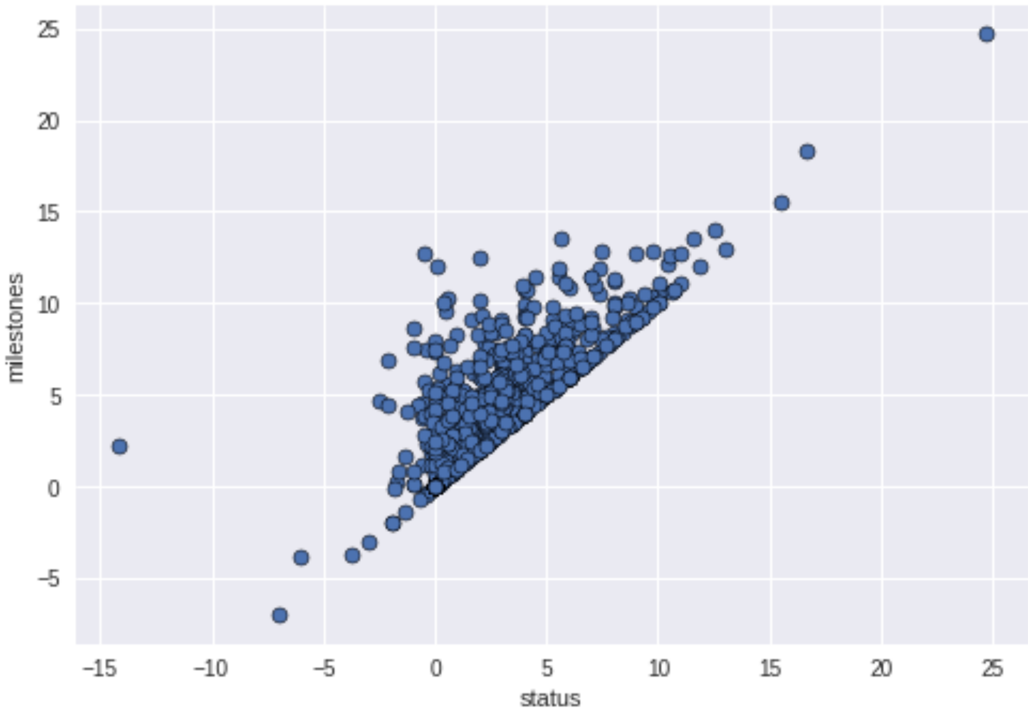



# Scatter plot

In [26]:
```
fig, ax = plt.subplots()
_ = plt.scatter(x=data['age_first_funding_year'], y=data['age_last_
_ = ax.set(xlabel="age_first_funding_year", ylabel="age_last_fundin
```



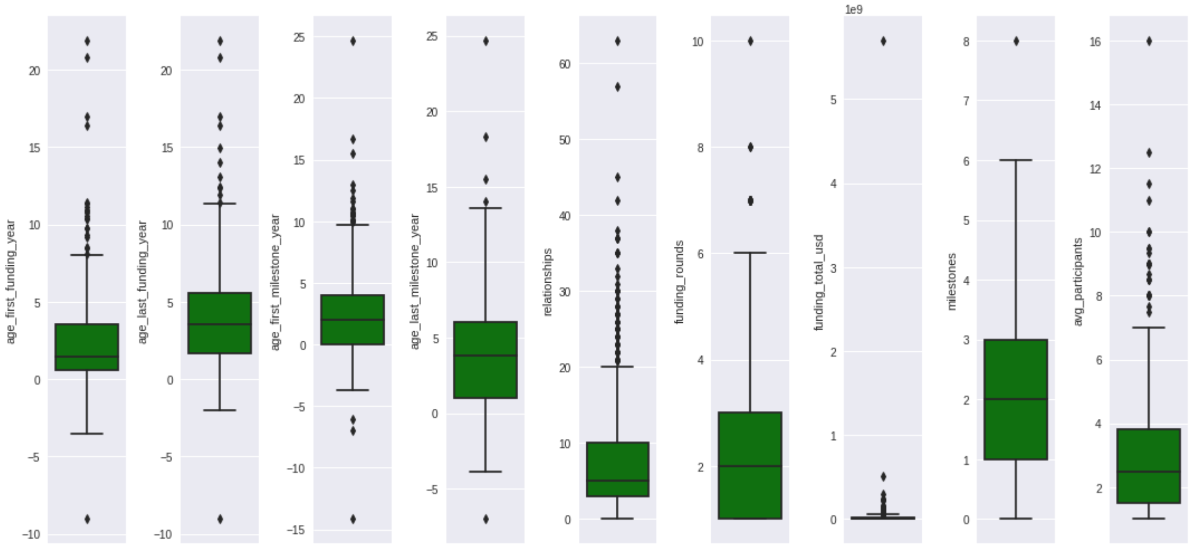

In [27]:
```
fig, ax = plt.subplots()
_ = plt.scatter(x=data['age_first_milestone_year'], y=data['age_las
_ = ax.set(xlabel="status", ylabel="milestones")
```



# Box plots

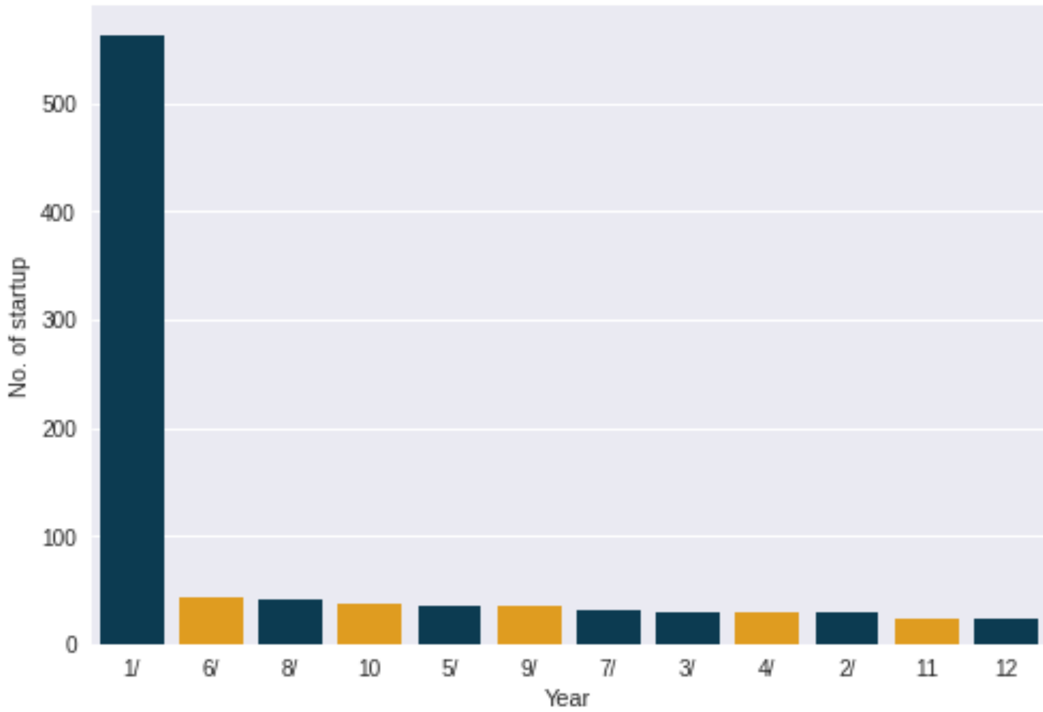
In [28]:

```
featuresNum = ['age_first_funding_year','age_last_funding_year','ag

plt.figure(figsize=(15, 7))
for i in range(0, len(featuresNum)):
    plt.subplot(1, len(featuresNum), i+1)
    sns.boxplot(y=data[featuresNum[i]], color='green', orient='v')
    plt.tight_layout()
```




# Dataset collection founded years

In [29]:
```
cdf = data["founded_at"].apply(lambda x: '' + x[:2]).value_counts()
            .to_frame().reset_index() \
            .rename(columns={"index": "year", "founded_at": "No_of_

fig, ax = plt.subplots()
_ = sns.barplot(x="year", y="No_of_startup", data=cdf,
                palette=sns.color_palette(['#003f5c', '#ffa600'], n
_ = ax.set(xlabel="Year", ylabel="No. of startup")
```




In [30]:
```
data["founded_at"].apply(lambda x: '20:' + x[:2]).value_counts(norm
```

Out[30]:
```
20:1/    563
20:6/     43
20:8/     42
20:10     38
20:5/     36
20:9/     35
20:7/     31
20:3/     30
20:4/     30
20:2/     29
20:11     23
20:12     23
Name: founded_at, dtype: int64
```

In [31]:
```
data["founded_at"].apply(lambda x: '20:' + x[:2]).value_counts(norm
```

Out[31]:
```
20:1/    0.609967
20:6/    0.046587
20:8/    0.045504
20:10    0.041170
20:5/    0.039003
20:9/    0.037920
20:7/    0.033586
20:3/    0.032503
20:4/    0.032503
20:2/    0.031419
20:11    0.024919
20:12    0.024919
Name: founded_at, dtype: float64
```

In [32]:
```
data["closed_at"].apply(lambda x: '20:' + x[:2]).value_counts(norma
```

Out[32]:
```
20:31    0.637053
20:1/    0.069339
20:6/    0.041170
20:7/    0.037920
20:2/    0.035753
20:5/    0.033586
20:8/    0.027086
20:10    0.020585
20:3/    0.020585
20:11    0.020585
20:4/    0.019502
20:12    0.018418
20:9/    0.018418
Name: closed_at, dtype: float64
```

# How many Startup are acquired or closed have?

In [33]:
```
df_acquired = data[(data["status"] == True)]
df_acquired.shape
```

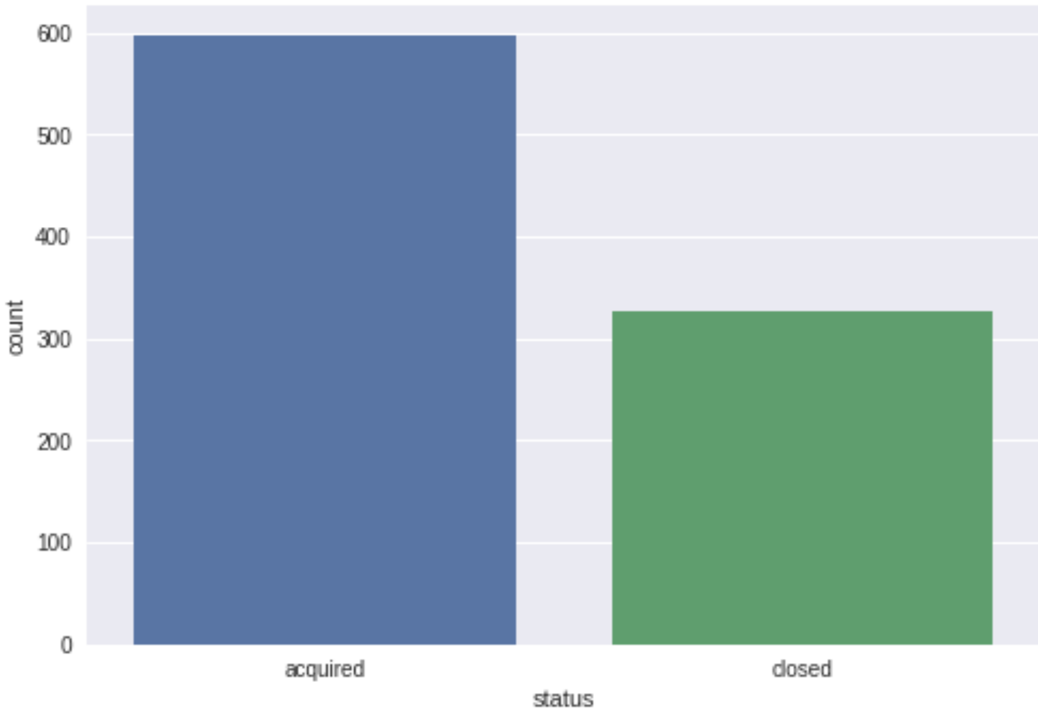
Out[33]:
```
(0, 48)
```

In [34]:
```
df_closed = data[(data["status"] == False)]
df_closed.shape
```

Out[34]:
```
(0, 48)
```

In [35]:
```
sns.countplot(data['status'])
```

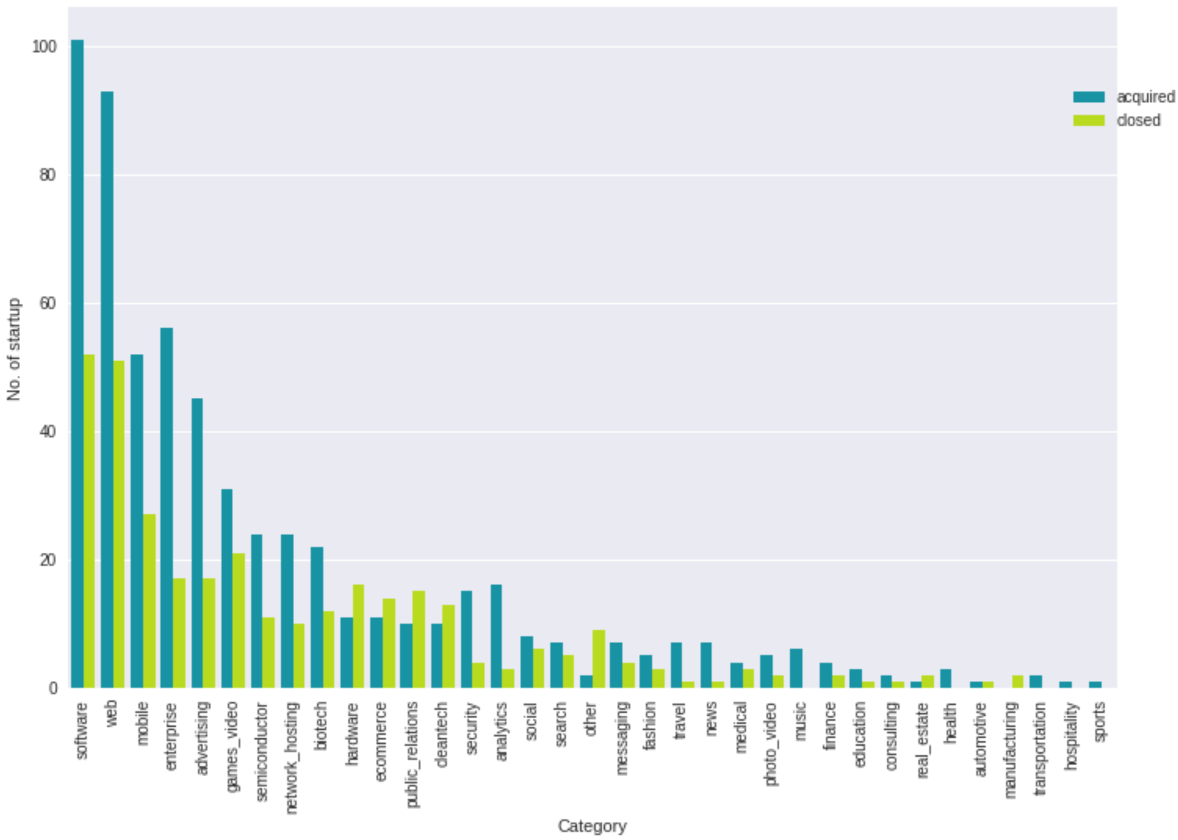
Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x7f88cc80cb50>




# Which category has the largest number of startup

In [36]:
```
fig, ax = plt.subplots(figsize=(12,8))

_ = sns.countplot(x="category_code", hue="status", data=data, palet
                  order=data.category_code.value_counts().index)

_ = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
_ = ax.set(xlabel="Category", ylabel="No. of startup")
plt.legend(bbox_to_anchor=(0.945, 0.90))
```

Out[36]: <matplotlib.legend.Legend at 0x7f88cc7480d0>



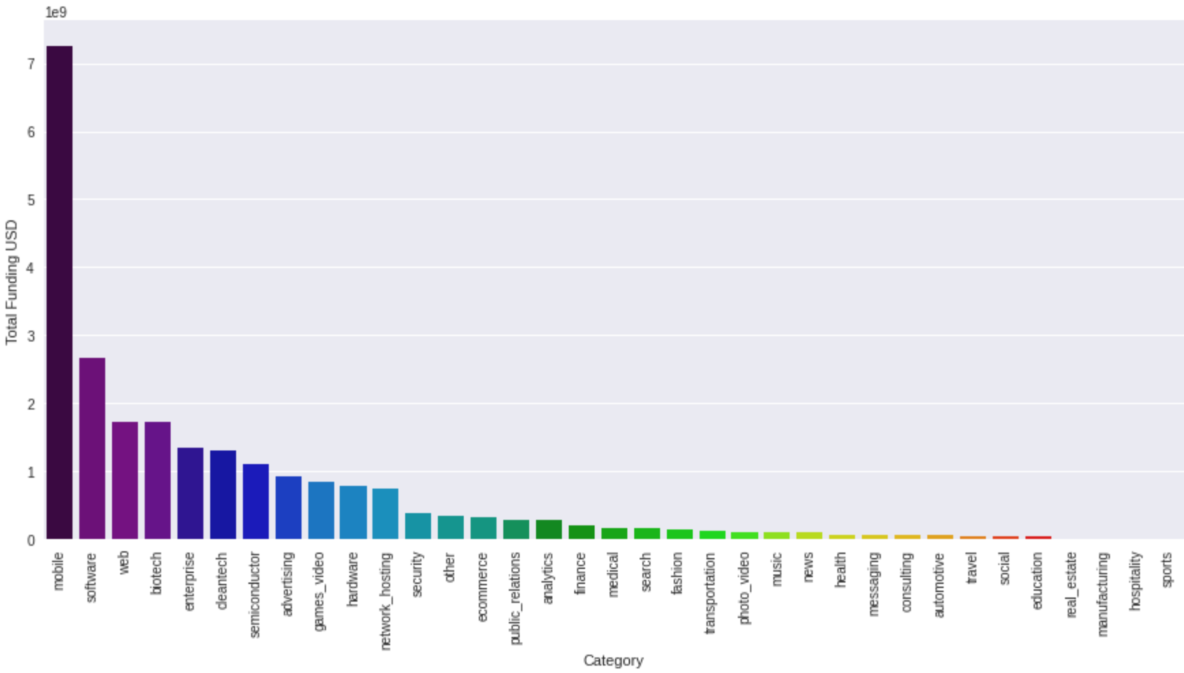

# Which category having most number of total funding

In [37]:
```
funding_sorted_category = pd.pivot_table(data,
              index=['category_code'],
              values=['funding_total_usd'],
              aggfunc=['sum']
              ).reset_index()
funding_sorted_category.columns = ['category_code', 'funding_total_
funding_sorted_category = funding_sorted_category.sort_values(['fun
funding_sorted_category.head(10)
```

Out[37]:

| | category_code | funding_total_usd |
|---|---|---|
| **18** | mobile | 7263750881 |
| **30** | software | 2657598865 |
| **34** | web | 1729035436 |
| **3** | biotech | 1723699484 |
| **8** | enterprise | 1338882096 |
| **4** | cleantech | 1300284730 |
| **28** | semiconductor | 1105156970 |
| **0** | advertising | 918619012 |
| **11** | games_video | 844643530 |
| **12** | hardware | 773938873 |

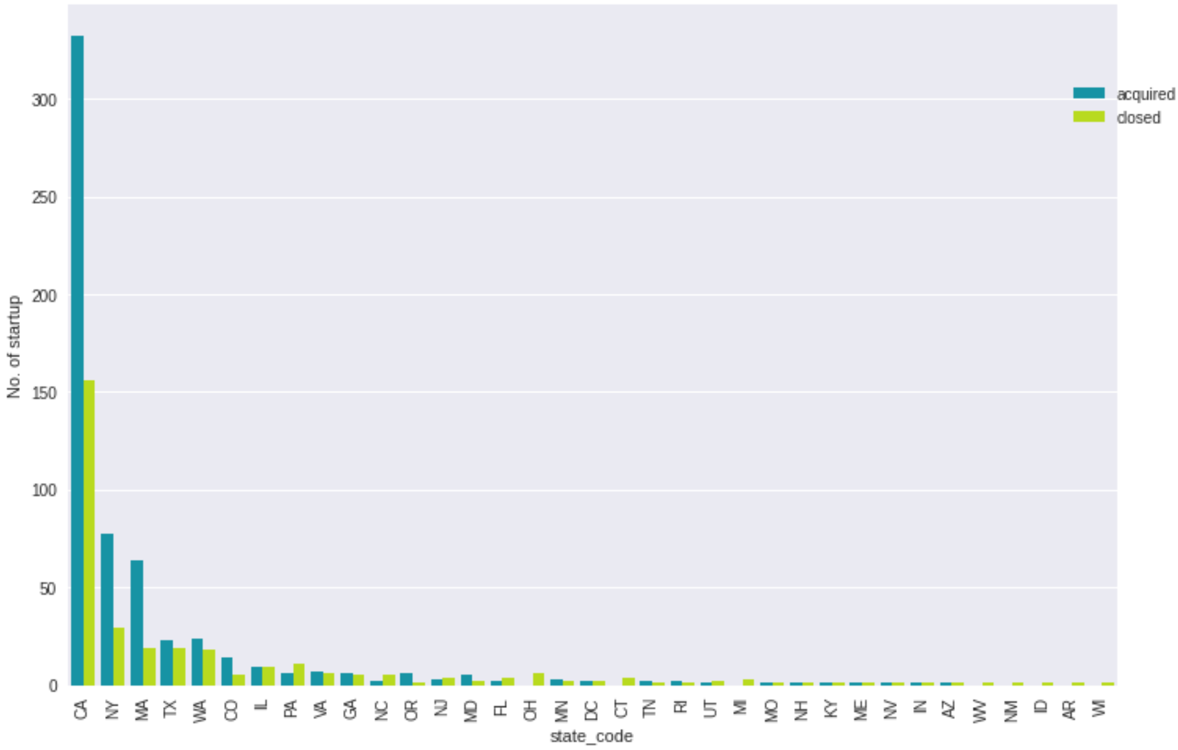
In [38]:
```
fig, ax = plt.subplots(figsize=(15,7))
_ = sns.barplot(x="category_code", y="funding_total_usd", data=fund
                palette="nipy_spectral", ax=ax)
_ = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
_ = ax.set(xlabel="Category", ylabel="Total Funding USD")
```




# Which State having most number of Startup

In [39]:
```
fig, ax = plt.subplots(figsize=(12,8))

_ = sns.countplot(x="state_code", hue="status", data=data, palette=
                  order=data.state_code.value_counts().index)

_ = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
_ = ax.set(xlabel="state_code", ylabel="No. of startup")
plt.legend(bbox_to_anchor=(0.945, 0.90))
```

Out[39]: <matplotlib.legend.Legend at 0x7f88cc3f10d0>




In [40]:
```
trending_statea = data.groupby(['state_code']).size().rename('num_s

most_trending_statea = trending_statea[trending_statea.groupby('sta
most_trending_statea = most_trending_statea.sort_values('num_startu
most_trending_statea
```

Out[40]:

| | state_code | num_startup |
|---|---|---|
| **2** | CA | 488 |
| **23** | NY | 106 |
| **12** | MA | 83 |
| **32** | WA | 42 |
| **29** | TX | 42 |
| **3** | CO | 19 |
| **9** | IL | 18 |
| **26** | PA | 17 |

| | | |
|---|---|---|
| **31** | VA | 13 |
| **7** | GA | 11 |
| **20** | NJ | 7 |
| **13** | MD | 7 |
| **25** | OR | 7 |
| **18** | NC | 7 |
| **6** | FL | 6 |
| **24** | OH | 6 |
| **16** | MN | 5 |
| **5** | DC | 4 |
| **4** | CT | 4 |
| **15** | MI | 3 |
| **28** | TN | 3 |
| **27** | RI | 3 |
| **30** | UT | 3 |
| **22** | NV | 2 |
| **19** | NH | 2 |
| **1** | AZ | 2 |
| **14** | ME | 2 |
| **11** | KY | 2 |
| **10** | IN | 2 |
| **17** | MO | 2 |
| **33** | WI | 1 |
| **0** | AR | 1 |
| **21** | NM | 1 |
| **8** | ID | 1 |
| **34** | WV | 1 |

# Which State having most number of acquired Startup per category

In [41]:
```
trending_statea = df_acquired.groupby(['state_code','category_code'

most_trending_statea = trending_statea[trending_statea.groupby('sta
most_trending_statea = most_trending_statea.sort_values('num_startu
most_trending_statea.head(10)
```

Out[41]:

| state_code | category_code | num_startup |
|---|---|---|

# Which State having most number of closed Startup per category

In [42]:
```
trending_statec = df_closed.groupby(['state_code','category_code'])

most_trending_statec = trending_statec[trending_statec.groupby('sta
most_trending_statec = most_trending_statec.sort_values('num_startu
most_trending_statec
```

Out[42]:

| state_code | category_code | num_startup |
|---|---|---|

# Which city having most number of acquired Startup per category

In [43]:
```
trending_categorya = df_acquired.groupby(['city','category_code']).

most_trending_categorya = trending_categorya[trending_categorya.gro
most_trending_categorya = most_trending_categorya.sort_values('num_
most_trending_categorya
```

Out[43]:

| city | category_code | num_startup |
|---|---|---|

# Which city having most number of closed Startup per category

In [44]:
```
trending_categoryc = df_closed.groupby(['city','category_code']).si

most_trending_categoryc = trending_categoryc[trending_categoryc.gro
most_trending_categoryc = most_trending_categoryc.sort_values('num_
most_trending_categoryc
```

Out[44]:

| index | city | category_code | num_startup |
|---|---|---|---|

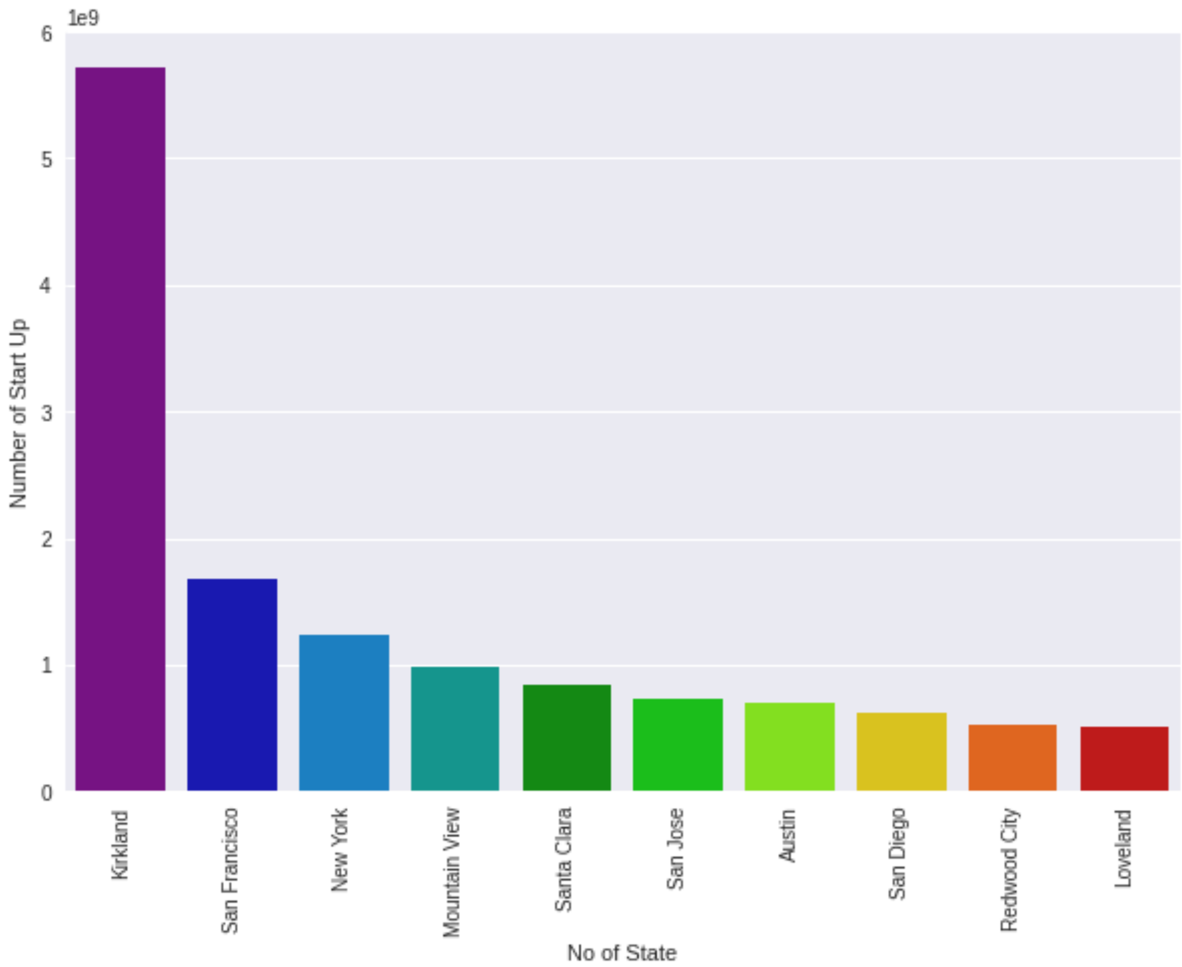
# Which city having most number of total funding

In [45]:
```
funding_sorted_city = pd.pivot_table(data,
              index=['city'],
              values=['funding_total_usd'],
              aggfunc=['sum']
              ).reset_index()
funding_sorted_city.columns = ['city', 'funding_total_usd']
funding_sorted_city = funding_sorted_city.sort_values(['funding_tot
funding_sorted_city = funding_sorted_city.head(10)
funding_sorted_city
```

Out[45]:

| | city | funding_total_usd |
|---|---|---|
| **91** | Kirkland | 5718914576 |
| **174** | San Francisco | 1673487129 |
| **135** | New York | 1231405734 |
| **125** | Mountain View | 985553322 |
| **181** | Santa Clara | 839050274 |
| **176** | San Jose | 733181780 |
| **13** | Austin | 706317317 |
| **173** | San Diego | 614475001 |
| **163** | Redwood City | 521330100 |
| **110** | Loveland | 510000000 |

In [46]:
```
fig, ax = plt.subplots(figsize=(10,7))
_ = sns.barplot(x="city", y="funding_total_usd", data=funding_sorte
                palette="nipy_spectral", ax=ax)
_ = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
_ = ax.set(xlabel="No of State", ylabel="Number of Start Up")
```




In [47]:
```
df_what_in_kirkland = data[(data["city"] == 'Kirkland')]
df_what_in_kirkland.shape
```

Out[47]: (2, 48)

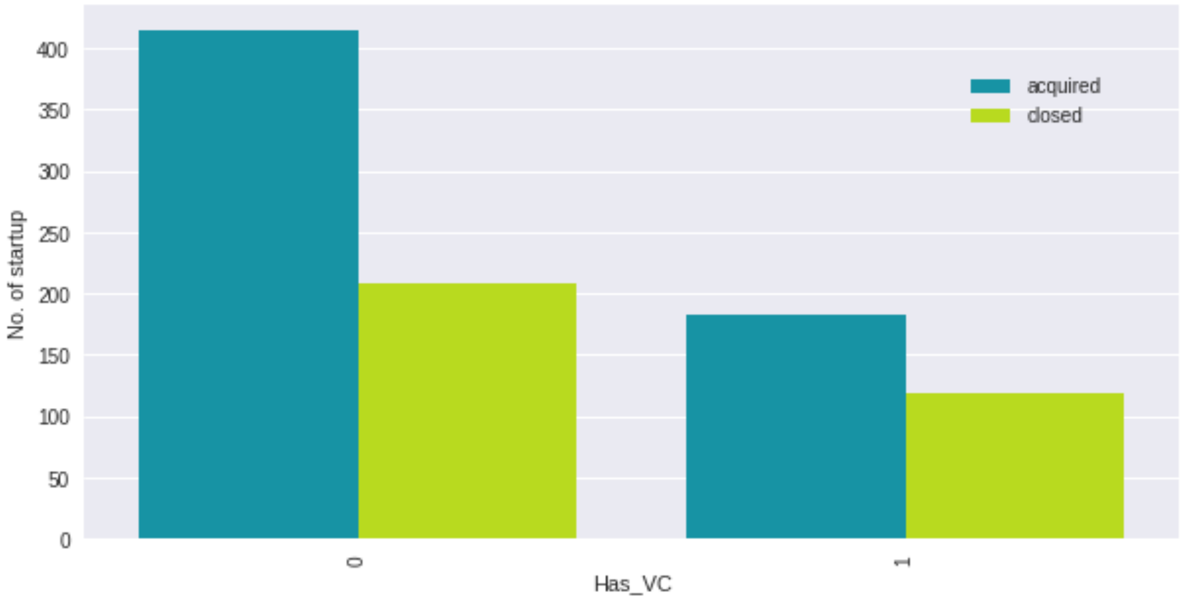
In [48]:
```
df_what_in_kirkland.head()
```

Out[48]:

| | Unnamed: 0 | state_code | latitude | longitude | zip_code | id | city | Unnamed: 6 |
|---|---|---|---|---|---|---|---|---|
| **62** | 332 | WA | 47.675489 | -122.191667 | 98033-6314 | c:19861 | Kirkland | Kirkland WA 98033-6314 |
| **364** | 86 | WA | 30.632480 | -86.984345 | 98033 | c:13219 | Kirkland | NaN |

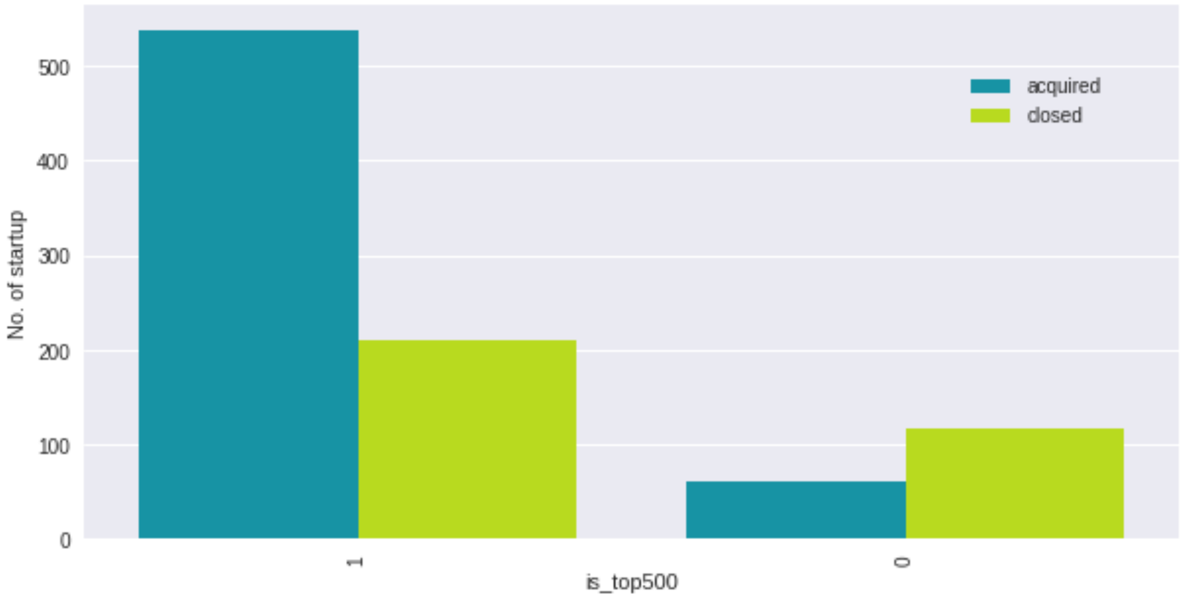
In [49]:
```
#How many Startup have has_VC?
fig, ax = plt.subplots(figsize=(10,5))

_ = sns.countplot(x="has_VC", hue="status", data=data, palette="nip
                 order=data.has_VC.value_counts().index)

_ = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
_ = ax.set(xlabel="Has_VC", ylabel="No. of startup")
plt.legend(bbox_to_anchor=(0.945, 0.90))
```

Out[49]: <matplotlib.legend.Legend at 0x7f88cc2e5c50>



In [50]:
```
#How many Startup have is_top500?
fig, ax = plt.subplots(figsize=(10,5))

_ = sns.countplot(x="is_top500", hue="status", data=data, palette="
                  order=data.is_top500.value_counts().index)

_ = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
_ = ax.set(xlabel="is_top500", ylabel="No. of startup")
plt.legend(bbox_to_anchor=(0.945, 0.90))
```

Out[50]:
```
<matplotlib.legend.Legend at 0x7f88cc4d1490>
```




In [51]:
```
#How many Startup have both 'acquired' status and is_top500?
len(data[(data["status"] == True) & (data["is_top500"] == True)].in
```

Out[51]:
```
0
```

In [52]:
```
#How many Startup have both 'closed' status and is_top500?
len(data[(data["status"] == False) & (data["is_top500"] == False)].
```

Out[52]:
```
0
```

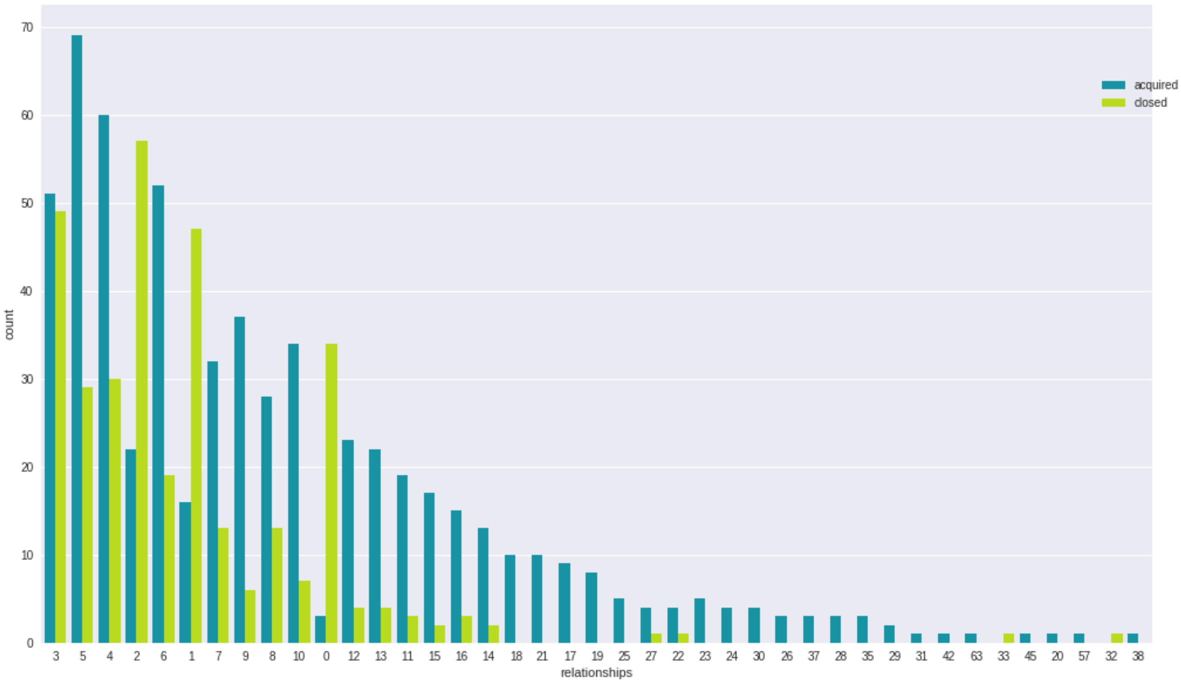
In [53]:
```
df_acquired["is_top500"].value_counts(normalize=True)
```

Out[53]:
```
Series([], Name: is_top500, dtype: float64)
```

In [54]:
```
#How many years on average the company closes
df_closed.founded_at=pd.to_datetime(df_closed.founded_at)
df_closed.closed_at=pd.to_datetime(df_closed.closed_at)
```

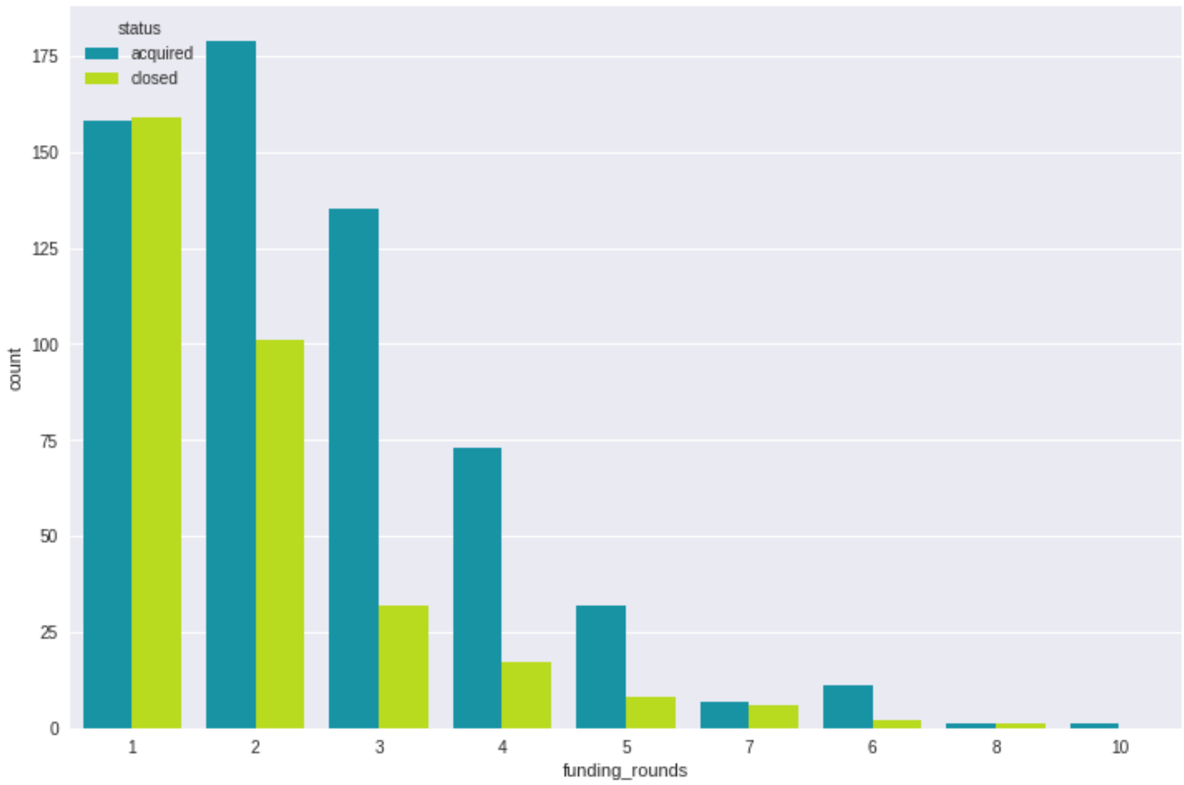
In [55]:
```
#which relationship related to acquired or closed startup?
fig, ax = plt.subplots(figsize=(17,10))

sns.countplot(x="relationships", hue="status", data=data, palette="
                order=data.relationships.value_counts().index)
plt.legend(bbox_to_anchor=(0.945, 0.90))
```

Out[55]: <matplotlib.legend.Legend at 0x7f88cc86f3d0>



In [56]:
```
#which funding_rounds related to acquired or closed startup?
fig, ax = plt.subplots(figsize=(12,8))

sns.countplot(x="funding_rounds", hue="status", data=data, palette=
              order=data.funding_rounds.value_counts().index)
```

Out[56]: <matplotlib.axes._subplots.AxesSubplot at 0x7f88ccbf37d0>




# Mapping area startup

In [57]:
```
'geopandas' in sys.modules
```

Out[57]: True

In [58]:
```
new_data = gpd.GeoDataFrame(data, geometry=gpd.points_from_xy(data.
```

In [59]:
```
new_data.head()
```

Out[59]:

| | Unnamed: 0 | state_code | latitude | longitude | zip_code | id | city | Unnamed: 6 |
|---|---|---|---|---|---|---|---|---|
| **0** | 1005 | CA | 42.358880 | -71.056820 | 92101 | c:6669 | San Diego | NaN |
| **1** | 204 | CA | 37.238916 | -121.973718 | 95032 | c:16283 | Los Gatos | NaN |
| **2** | 1001 | CA | 32.901049 | -117.192656 | 92121 | c:65620 | San Diego | San Diego CA 92121 |
| **3** | 738 | CA | 37.320309 | -122.050040 | 95014 | c:42668 | Cupertino | Cupertino CA 95014 |
| **4** | 1002 | CA | 37.779281 | -122.419236 | 94105 | c:65806 | San Francisco | San Francisco CA 94105 |

In [60]:
```
age=["age_first_funding_year","age_last_funding_year","age_first_mi

for a in range(len(age)):
    print("Is there any negative value in '{}' column  : {} ".forma
```

```
Is there any negative value in 'age_first_funding_year' column  :
True
Is there any negative value in 'age_last_funding_year' column  : T
rue
Is there any negative value in 'age_first_milestone_year' column
: True
Is there any negative value in 'age_last_milestone_year' column  :
True
```

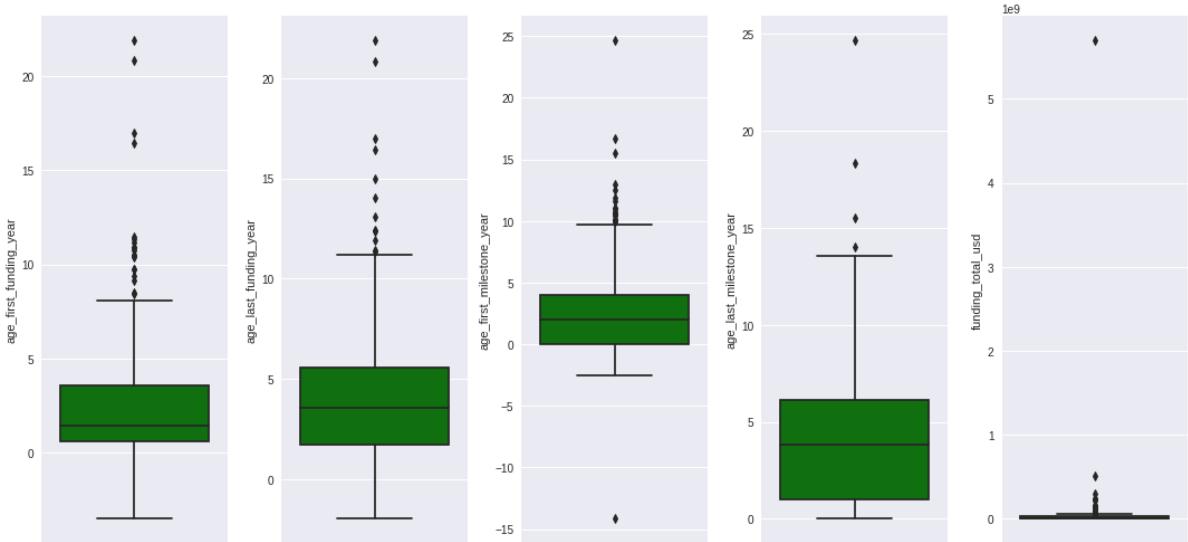
In [61]:
```
df=data.drop(data[data.age_first_funding_year<0].index)
df=data.drop(data[data.age_last_funding_year<0].index)
df=data.drop(df[data.age_first_milestone_year<0].index)
df=data.drop(data[data.age_last_milestone_year<0].index)
```

In [62]:
```
for a in range(len(age)):
    print("Is there any negative value in '{}' column  : {} ".forma
```

```
Is there any negative value in 'age_first_funding_year' column  :
True
Is there any negative value in 'age_last_funding_year' column  : T
rue
Is there any negative value in 'age_first_milestone_year' column
: True
Is there any negative value in 'age_last_milestone_year' column  :
False
```

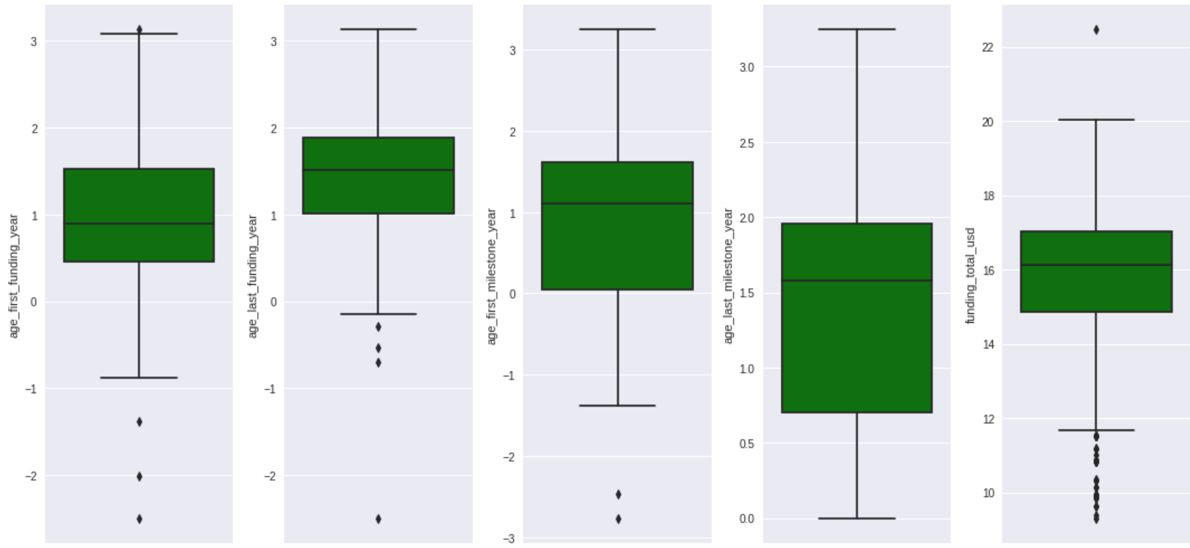
# Outliers

In [63]:
```
featuresNumfinal = ['age_first_funding_year','age_last_funding_year

plt.figure(figsize=(15, 7))
for i in range(0, len(featuresNumfinal)):
    plt.subplot(1, len(featuresNumfinal), i+1)
    sns.boxplot(y=df[featuresNumfinal[i]], color='green', orient='v
    plt.tight_layout()
```




In [64]:
```
#Log-transformation of the funding and milestone year variable
df["age_first_funding_year"] = np.log1p(df["age_first_funding_year"
df["age_last_funding_year"] = np.log1p(df["age_last_funding_year"])
df["age_first_milestone_year"] = np.log1p(df["age_first_milestone_y
df["age_last_milestone_year"] = np.log1p(df["age_last_milestone_yea
df["funding_total_usd"] = np.log1p(df["funding_total_usd"])
```

In [65]:
```
featuresNumfinal = ['age_first_funding_year','age_last_funding_year

plt.figure(figsize=(15, 7))
for i in range(0, len(featuresNumfinal)):
    plt.subplot(1, len(featuresNumfinal), i+1)
    sns.boxplot(y=df[featuresNumfinal[i]], color='green', orient='v
    plt.tight_layout()
```




# Feature Engineering

In [66]:
```
#New Column has_RoundABCD
df['has_RoundABCD'] = np.where((df['has_roundA'] == 1) | (df['has_r
df.head()
```

Out[66]:

| | Unnamed: 0 | state_code | latitude | longitude | zip_code | id | city | Unnamed: 6 |
|---|---|---|---|---|---|---|---|---|
| 0 | 1005 | CA | 42.358880 | -71.056820 | 92101 | c:6669 | San Diego | NaN |
| 1 | 204 | CA | 37.238916 | -121.973718 | 95032 | c:16283 | Los Gatos | NaN |
| 2 | 1001 | CA | 32.901049 | -117.192656 | 92121 | c:65620 | San Diego | San Diego CA 92121 |
| 3 | 738 | CA | 37.320309 | -122.050040 | 95014 | c:42668 | Cupertino | Cupertino CA 95014 |
| 4 | 1002 | CA | 37.779281 | -122.419236 | 94105 | c:65806 | San Francisco | San Francisco CA 94105 |

In [67]:
```
#New Column "has_Investor
df['has_Investor'] = np.where((df['has_VC'] == 1) | (df['has_angel'
df.head()
```

Out[67]:

| | Unnamed: 0 | state_code | latitude | longitude | zip_code | id | city | Unnamed: 6 |
|---|---|---|---|---|---|---|---|---|
| **0** | 1005 | CA | 42.358880 | -71.056820 | 92101 | c:6669 | San Diego | NaN |
| **1** | 204 | CA | 37.238916 | -121.973718 | 95032 | c:16283 | Los Gatos | NaN |
| **2** | 1001 | CA | 32.901049 | -117.192656 | 92121 | c:65620 | San Diego | San Diego CA 92121 |
| **3** | 738 | CA | 37.320309 | -122.050040 | 95014 | c:42668 | Cupertino | Cupertino CA 95014 |
| **4** | 1002 | CA | 37.779281 | -122.419236 | 94105 | c:65806 | San Francisco | San Francisco CA 94105 |

In [68]:
```
len(df[(df["has_RoundABCD"] == 1)].index)
```

Out[68]: 674

In [69]:
```
len(df[ (df['has_RoundABCD']  == 1) & (df['status']  == 1) ].index)
len(df)
```

Out[69]: 911

In [70]:
```
#New Column "has_Seed"
df['has_Seed'] = np.where((df['has_RoundABCD'] == 0) & (df['has_Inv
df.head()
```

Out[70]:

| | Unnamed: 0 | state_code | latitude | longitude | zip_code | id | city | Unnamed: 6 |
|---|---|---|---|---|---|---|---|---|
| **0** | 1005 | CA | 42.358880 | -71.056820 | 92101 | c:6669 | San Diego | NaN |
| **1** | 204 | CA | 37.238916 | -121.973718 | 95032 | c:16283 | Los Gatos | NaN |
| **2** | 1001 | CA | 32.901049 | -117.192656 | 92121 | c:65620 | San Diego | San Diego CA 92121 |
| **3** | 738 | CA | 37.320309 | -122.050040 | 95014 | c:42668 | Cupertino | Cupertino CA 95014 |
| **4** | 1002 | CA | 37.779281 | -122.419236 | 94105 | c:65806 | San Francisco | San Francisco CA 94105 |

In [71]:
```
df['has_Seed'] == 1
```

Out[71]:
```
0      True
1     False
2     False
3     False
4      True
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12     True
13    False
14    False
15     True
16    False
17    False
18    False
19    False
```

# Model belding

In [72]:
```
#Cek categorical
cat_feature = df.select_dtypes(include='object')
cat_feature.head()
```

Out[72]:

| | state_code | zip_code | id | city | Unnamed: 6 | name | founded_at | closed_a |
|---|---|---|---|---|---|---|---|---|
| **0** | CA | 92101 | c:6669 | San Diego | NaN | Bandsintown | 1/1/2007 | 31/12/201 |
| **1** | CA | 95032 | c:16283 | Los Gatos | NaN | TriCipher | 1/1/2000 | 31/12/201 |
| **2** | CA | 92121 | c:65620 | San Diego | San Diego CA 92121 | Plixi | 3/18/2009 | 31/12/201 |
| **3** | CA | 95014 | c:42668 | Cupertino | Cupertino CA 95014 | Solidcore Systems | 1/1/2002 | 31/12/201 |
| **4** | CA | 94105 | c:65806 | San Francisco | San Francisco CA 94105 | Inhale Digital | 8/1/2010 | 10/1/201 |

In [73]:
```
df = data.drop(['state_code'],axis=1)
df = df.drop(['id'],axis=1)
df = df.drop(['Unnamed: 6'],axis=1)
df = df.drop(['category_code'],axis=1)
df = df.drop(['object_id'],axis=1)
df = df.drop(['zip_code'],axis=1)
df = df.drop(['founded_at'],axis=1)
df = df.drop(['closed_at'],axis=1)
df = df.drop(['first_funding_at'],axis=1)
df = df.drop(['last_funding_at'],axis=1)
df = df.drop(['city'],axis=1)
df = df.drop(['name'],axis=1)
df = df.drop(['Unnamed: 0'],axis=1)
df = df.drop(['latitude','longitude'],axis=1)
df = df.drop(['geometry'],axis=1)
#df = df.drop(['age_closed_startup'],axis=1)
df = df.drop(['relationships'],axis=1)
```

In [74]:
```
df.columns
```

Out[74]:
```
Index(['labels', 'age_first_funding_year', 'age_last_funding_year'
,
       'age_first_milestone_year', 'age_last_milestone_year', 'fun
ding_rounds',
       'funding_total_usd', 'milestones', 'is_CA', 'is_NY', 'is_MA
', 'is_TX',
       'is_otherstate', 'is_software', 'is_web', 'is_mobile', 'is_
enterprise',
       'is_advertising', 'is_gamesvideo', 'is_ecommerce', 'is_biot
ech',
       'is_consulting', 'is_othercategory', 'has_VC', 'has_angel',
       'has_roundA', 'has_roundB', 'has_roundC', 'has_roundD',
       'avg_participants', 'is_top500', 'status'],
      dtype='object')
```

In [75]:
```
#del data['Unnamed: 6']
del data['Unnamed: 0']
```

In [76]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 923 entries, 0 to 922
Data columns (total 32 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   labels                    923 non-null    int64  
 1   age_first_funding_year    923 non-null    float64
 2   age_last_funding_year     923 non-null    float64
 3   age_first_milestone_year  923 non-null    float64
 4   age_last_milestone_year   923 non-null    float64
 5   funding_rounds            923 non-null    int64  
 6   funding_total_usd         923 non-null    int64  
 7   milestones                923 non-null    int64  
 8   is_CA                     923 non-null    int64  
 9   is_NY                     923 non-null    int64  
 10  is_MA                     923 non-null    int64  
 11  is_TX                     923 non-null    int64  
 12  is_otherstate             923 non-null    int64  
 13  is_software               923 non-null    int64  
 14  is_web                    923 non-null    int64  
 15  is_mobile                 923 non-null    int64  
 16  is_enterprise             923 non-null    int64  
 17  is_advertising            923 non-null    int64  
 18  is_gamesvideo             923 non-null    int64  
 19  is_ecommerce              923 non-null    int64  
 20  is_biotech                923 non-null    int64  
 21  is_consulting             923 non-null    int64  
 22  is_othercategory          923 non-null    int64  
 23  has_VC                    923 non-null    int64  
 24  has_angel                 923 non-null    int64  
 25  has_roundA                923 non-null    int64  
 26  has_roundB                923 non-null    int64  
 27  has_roundC                923 non-null    int64  
 28  has_roundD                923 non-null    int64  
 29  avg_participants          923 non-null    float64
 30  is_top500                 923 non-null    int64  
 31  status                    923 non-null    object 
dtypes: float64(5), int64(26), object(1)
memory usage: 230.9+ KB
```

In [77]:
```
X = df.drop('status', axis=1)
Y = df['status']
```

In [78]:
```
print(X)
print(Y)
```

```
     labels  age_first_funding_year  age_last_funding_year  \
0         1                  2.2493                 3.0027
1         1                  5.1260                 9.9973
2         1                  1.0329                 1.0329
3         1                  3.1315                 5.3151
4         0                  0.0000                 1.6685
5         0                  4.5452                 4.5452
6         1                  1.7205                 5.2110
7         1                  1.6466                 6.7616
8         1                  3.5863                11.1123
9         1                  1.6712                 4.6849
10        1                  4.6274                 9.4493
11        0                  1.0849                 5.3370
12        0                  4.9041                 4.9041
13        1                  0.0192                 2.4356
14        1                  4.6658                 8.9973
15        0                  6.6082                 6.6082
16        0                  2.5863                 6.7644
17        1                  4.5918                 7.1726
```

Type *Markdown* and LaTeX: $\alpha^2$

In [79]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size
```

In [81]:
```
print(X_train.shape, X_test.shape, Y_train.shape, Y_test.shape)
```

```
(738, 31) (185, 31) (738,) (185,)
```

In [82]:
```
from sklearn.preprocessing import StandardScaler
scler = StandardScaler()
X_train = scler.fit_transform(X_train)
X_test = scler.transform(X_test)
```

In [84]:
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

In [85]:
```
model.fit(X_train, Y_train)
```

Out[85]:
```
LogisticRegression()
```

In [87]:

```
#accuracy on training data
from sklearn.metrics import accuracy_score
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train
```

In [88]:

```
print('Accuracy on training data : ', training_data_accuracy)
```

```
Accuracy on training data :  1.0
```

In [89]:

```
#accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

In [90]:

```
print('Accuracy on test data : ', test_data_accuracy)
```

```
Accuracy on test data :  1.0
```