

Kickstarter Dashboard

Kickstarter is a crowdfunding platform that aims to "help bring creative projects to life". The data was originally from <https://www.kaggle.com/datasets/kemical/kickstarter-projects?resource=download>.

```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import holoviews as hv
import hvplot.pandas
import panel as pn
pn.extension()

import warnings
warnings.filterwarnings('ignore')
```

```
In [3]: data_df = pd.read_csv('kickstarter.csv')
```

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

Out [4]:

	ID	name	category	main_category	country	currency	launched	deadline	goal	pledged	state	backers	usd_pledged_real	usd_goal_real
0	1000002330	The Songs of Adelaide & Abullah	Poetry	Publishing	GB	GBP	2015-08-11 12:12:28	2015-10-09	1000.0	0.0	failed	0	0.0	0.0
1	1000003930	Greeting From Earth: ZGAC Arts Capsule For ET	Narrative Film	Film & Video	US	USD	2017-09-02 04:43:57	2017-11-01	30000.0	2421.0	failed	15	2421.0	2421.0
2	1000004038	Where is Hank?	Narrative Film	Film & Video	US	USD	2013-01-12 00:20:50	2013-02-26	45000.0	220.0	failed	3	220.0	220.0
3	1000007540	ToshiCapital Rekordz Needs Help to Complete Album	Music	Music	US	USD	2012-03-17 03:24:11	2012-04-16	5000.0	1.0	failed	1	1.0	1.0
4	1000011046	Community Film Project: The Art of Neighborhoo...	Film & Video	Film & Video	US	USD	2015-07-04 08:35:03	2015-08-29	19500.0	1283.0	canceled	14	1283.0	1283.0

In [5]: data_df.columns

Out[5]: Index(['ID', 'name', 'category', 'main_category', 'country', 'currency', 'launched', 'deadline', 'goal', 'pledged', 'state', 'backers', 'usd_pledged_real', 'usd_goal_real'], dtype='object')

Turn Data into Specific Format

Change the format of columns, `launched` and `deadline`, to datetime and generate new columns to represent the `year` and `month` during which the project was `launched`

In [6]: *# check the data type of each column*
data_df['launched'].dtype
data_df.dtypes

```
Out [6]: ID                int64
          name              object
          category          object
          main_category      object
          country            object
          currency           object
          launched           object
          deadline           object
          goal               float64
          pledged            float64
          state              object
          backers            int64
          usd_pledged_real   float64
          usd_goal_real      float64
          dtype: object
```

```
In [7]: # pd.to_datetime()
data_df['launched'] = pd.to_datetime(data_df['launched'])
data_df['deadline'] = pd.to_datetime(data_df['deadline'])
data_df.dtypes
```

```
Out[7]: ID                int64
          name              object
          category          object
          main_category      object
          country            object
          currency           object
          launched           datetime64[ns]
          deadline           datetime64[ns]
          goal               float64
          pledged            float64
          state              object
          backers            int64
          usd_pledged_real   float64
          usd_goal_real      float64
          dtype: object
```

```
In [8]: # year, month, day, hour, minute, second
data_df['year_launched'] = data_df['launched'].dt.year
data_df['month_launched'] = data_df['launched'].dt.month
data_df.head()
```

Out [8]:

	ID	name	category	main_category	country	currency	launched	deadline	goal	pledged	state	backers	usd_pledged_real	t
0	1000002330	The Songs of Adelaide & Abullah	Poetry	Publishing	GB	GBP	2015-08-11 12:12:28	2015-10-09	1000.0	0.0	failed	0	0.0	
1	1000003930	Greeting From Earth: ZGAC Arts Capsule For ET	Narrative Film	Film & Video	US	USD	2017-09-02 04:43:57	2017-11-01	30000.0	2421.0	failed	15	2421.0	
2	1000004038	Where is Hank?	Narrative Film	Film & Video	US	USD	2013-01-12 00:20:50	2013-02-26	45000.0	220.0	failed	3	220.0	
3	1000007540	ToshiCapital Rekordz Needs Help to Complete Album	Music	Music	US	USD	2012-03-17 03:24:11	2012-04-16	5000.0	1.0	failed	1	1.0	
4	1000011046	Community Film Project: The Art of Neighborhoo...	Film & Video	Film & Video	US	USD	2015-07-04 08:35:03	2015-08-29	19500.0	1283.0	canceled	14	1283.0	



Deal with Noisy data

Find out data with noisy launched time and decide whether to fix the noise or remove the noise

```
In [9]: data_df['year_launched'].value_counts()
```

```
Out [9]: 2015    77300
2014    67745
2016    57184
2017    52200
2013    44851
2012    41165
2011    26237
2010    10519
2009     1329
2018     124
1970       7
Name: year_launched, dtype: int64
```

```
In [10]: data_df = data_df[data_df['year_launched'] != 1970]
```

```
In [11]: # check whether there is missing data in column 'year_launched'
data_df.isna().any()
```

Out[11]:

ID	False
name	True
category	False
main_category	False
country	False
currency	False
launched	False
deadline	False
goal	False
pledged	False
state	False
backers	False
usd_pledged_real	False
usd_goal_real	False
year_launched	False
month_launched	False
dtype: bool	

```
In [12]: data_df[data_df['name'].isna()]
```

Out[12]:

	ID	name	category	main_category	country	currency	launched	deadline	goal	pledged	state	backers	usd_pledged_real
166851	1848699072	NaN	Narrative Film	Film & Video	US	USD	2012-01-01 12:35:31	2012-02-29	200000.0	100.0	failed	1	100.00
307234	634871725	NaN	Video Games	Games	GB	GBP	2012-12-19 23:57:48	2013-01-06	2000.0	196.0	failed	12	316.05
309991	648853978	NaN	Product Design	Design	US	USD	2016-06-18 05:01:47	2016-07-18	2500.0	0.0	suspended	0	0.00
338931	796533179	NaN	Painting	Art	US	USD	2011-11-06 23:55:55	2011-12-05	35000.0	220.0	failed	5	220.00



Calculate Success rate

Create a figure with two subplots (one column two rows) sharing

- Subplot 1: Use line chart to visualize the average success rate of projects (i.e., annual success rate) across years until 2017 (in other words, x-axis: year, y-axis: success rate)
- Subplot 2: Use a bar plot to display the number of projects per year (until 2017)

```
In [13]: # duration that is until 2017
data_df = data_df[data_df['year_launched'] < 2018]
```

```
In [14]: # success case
success = data_df[data_df['state'] == 'successful']
```

```
In [15]: # success case every year
all_data = data_df.groupby('year_launched')['ID'].count()
success_data = success.groupby('year_launched')['ID'].count()
```

```
In [16]: success_rate = success_data / all_data
success_data / all_data
```

```
Out[16]: year_launched
2009      0.435666
2010      0.436638
2011      0.463887
2012      0.434641
2013      0.432878
2014      0.311565
2015      0.271294
2016      0.328169
2017      0.353678
Name: ID, dtype: float64
```

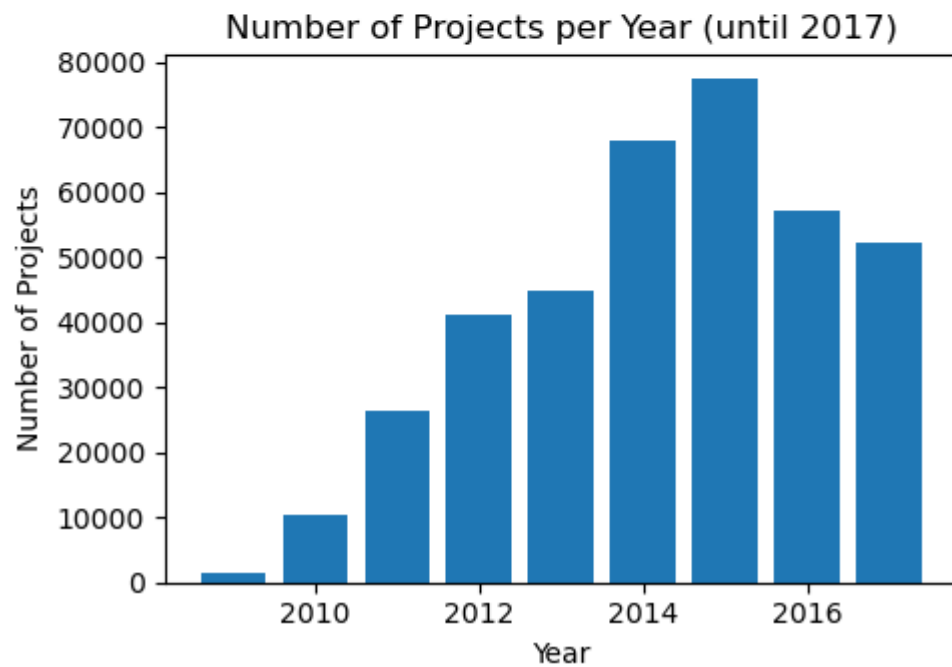
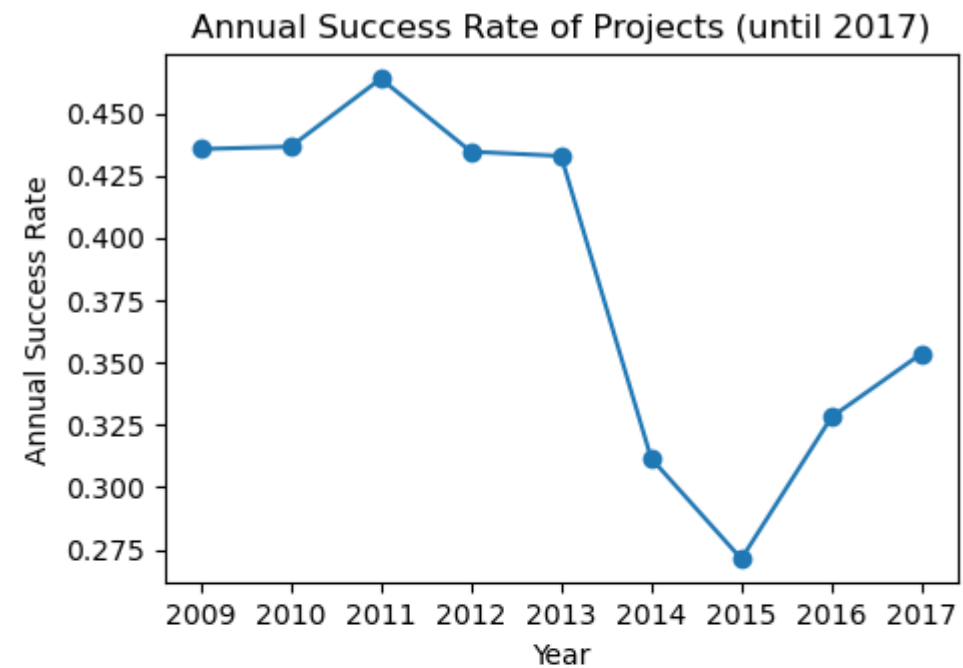
```
In [17]: # creates a figure (fig) and a set of subplots (axes)
fig, axes = plt.subplots(2, 1, figsize=(5, 7))

# Subplot 1: Line chart for annual success rate
axes[0].plot(success_rate.index, success_rate.values, marker='o', linestyle='-')
axes[0].set_xlabel('Year')
axes[0].set_ylabel('Annual Success Rate')
axes[0].set_title('Annual Success Rate of Projects (until 2017)')

# Subplot 2: the number of projects per year
axes[1].bar(all_data.index, all_data.values)
axes[1].set_xlabel('Year')
axes[1].set_ylabel('Number of Projects')
axes[1].set_title('Number of Projects per Year (until 2017)')

# Adjust layout for subplots
```

```
plt.tight_layout()  
plt.show()
```



Hypothesis and Exploration

Use a visualization to show preliminary results examining the hypothesis.

The success rate of different 'main_category' differs. From 2009 to 2017, year 2015 has the lowest annual success rate. I hypothesize that in the year 2015, the category with the lowest success rate takes a larger proportion of projects compared to other years.

```
In [21]: # success rate of different categories
success_by_category = success.groupby('main_category')['ID'].count()
by_category = data_df.groupby('main_category')['ID'].count()
success_rate_by_category = (success_by_category / by_category).sort_values(ascending=False)

# Technology has the lowest success rate.
# Therefore, the next step is to check whether technology takes higher project proportion in 2015.
technology_df = data_df[data_df['main_category'] == 'Technology']
technology_rate = technology_df.groupby('year_launched')['ID'].count() / all_data

fig, axes = plt.subplots(2, 1, figsize=(15, 12))

# Subplot 1: Different main categories success rate
axes[0].plot(success_rate_by_category.index, success_rate_by_category.values, marker='o', linestyle='-')
axes[0].set_xlabel('Year')
axes[0].set_ylabel('Success Rate by Categories')
axes[0].set_title('Different main categories success rate')

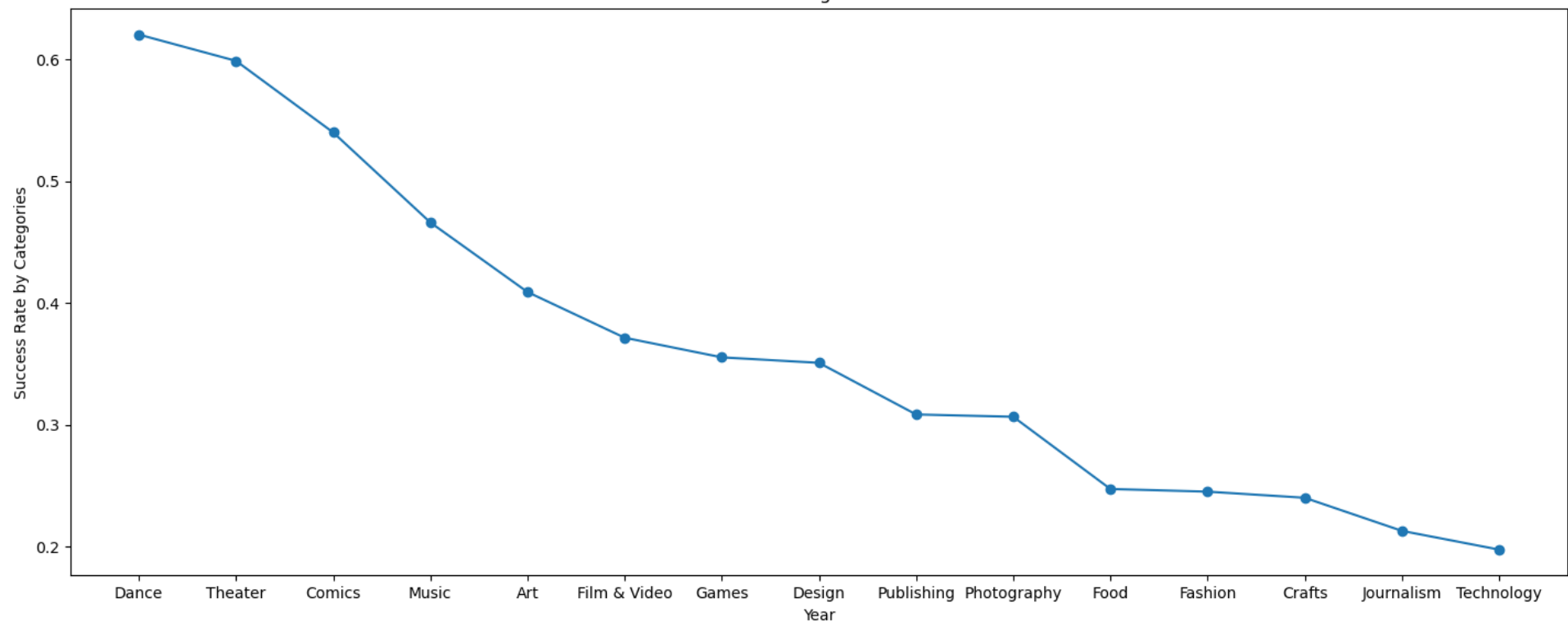
# Subplot 2: the number of projects per year
axes[1].bar(success_rate.index, success_rate.values, color='g')
axes[1].set_xlabel('Year')
axes[1].set_ylabel('Annual Success Rate', color='g')
axes[1].set_title('Annual Success Rate of Projects & Technology Proportion per Year')

# Create a right axis that shares the same x-axis with the left axis
ax2 = axes[1].twinx()

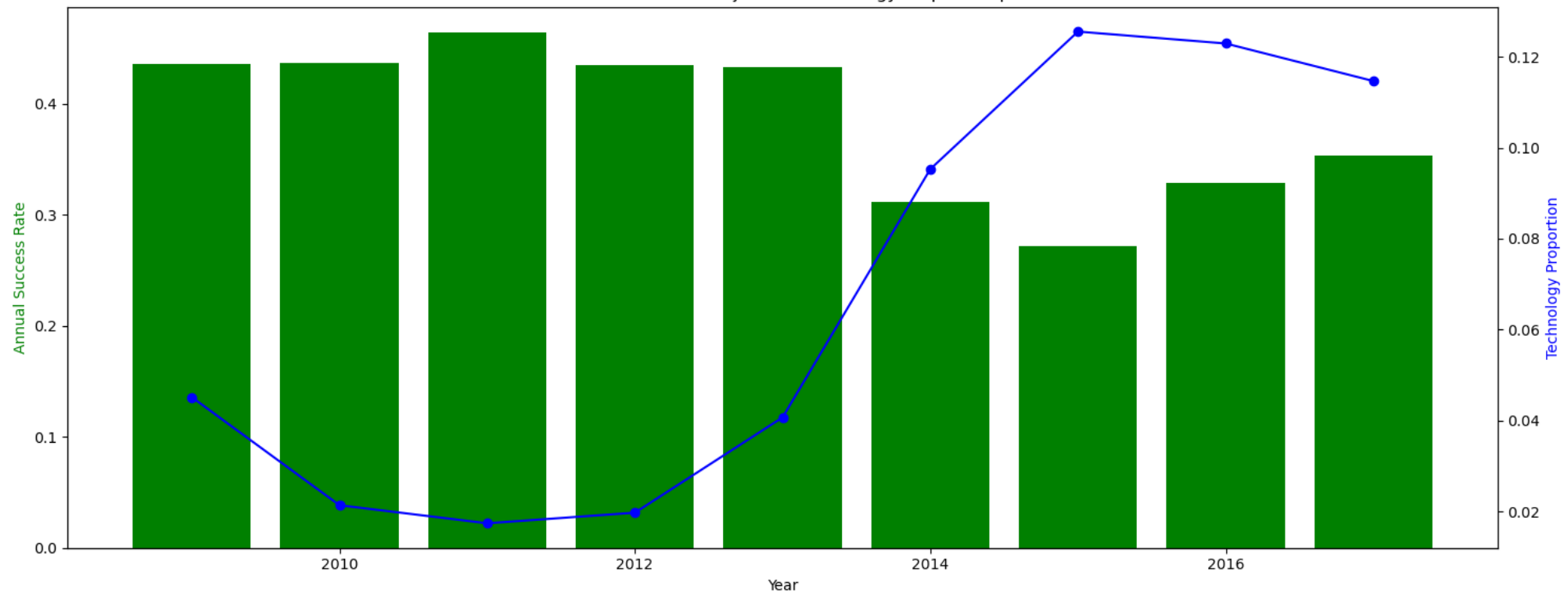
ax2.plot(technology_rate.index, technology_rate.values, marker='o', linestyle='-', color='b', label='Technology')
ax2.set_ylabel('Technology Proportion', color='b')

plt.tight_layout()
plt.show()
```


Different main categories success rate



Annual Success Rate of Projects & Technology Proportion per Year



The first graph shows that 'Technology' category has the lowest success rate. The second graph shows that the year 2015 has the lowest annual success rate and highest proportion of 'Technology' category project.

Implement Dashboard

Build an interactive dashboard following the requirements below:

1. Could choose from one of the three countries: US, France (FR), United Kingdom of Great Britain and Northern Ireland (GB)
2. Has a slider to choose the year of the visualization
3. Could choose to visualize either the number of projects or the number of success projects (as y-axis)
4. Has a line chart to visualize the y-axis across months of the year

```
In [36]: # filter and add the column we need in this exercise
columns_to_keep = ['ID', 'country', 'state', 'year_launched', 'month_launched']
idf = data_df[columns_to_keep]
idf = idf.groupby(['country', 'year_launched', 'month_launched', 'state']).count().reset_index()
idf['project_count'] = idf['ID']
```

```
In [37]: # Define a custom function to conditionally create the new column
def new_column(row):
    if row['state'] == 'successful':
        return row['project_count']
    else:
        return 0

# Apply the custom function to create the new column
idf['success_project_count'] = idf.apply(new_column, axis=1)
idf = idf.drop('ID', axis=1)
idf = idf.drop('state', axis=1)
idf
```

Out [37]:

	country	year_launched	month_launched	project_count	success_project_count
0	AT	2015	6	11	0
1	AT	2015	7	8	0
2	AT	2015	7	18	0
3	AT	2015	7	2	2
4	AT	2015	7	1	0
...
2837	US	2017	12	120	0
2838	US	2017	12	138	0
2839	US	2017	12	1322	0
2840	US	2017	12	106	106
2841	US	2017	12	4	0

2842 rows × 5 columns

```
In [38]: # Step 1: get interactive dataframe
idf = idf.interactive()
```

```
In [39]: # can choose one value
country = pn.widgets.RadioButtonGroup(
    name='Country',
    options=['US', 'FR', 'GB'],
    value=['US', 'FR', 'GB'],
    button_type='success',
    width_policy='max',
    sizing_mode='stretch_width' # same width
)
# country_list = ['US', 'FR', 'GB']
country
```

Out[39]:

US	FR	GB
----	----	----

```
In [26]: # Step 2: choose the year of visualization
year = pn.widgets.IntSlider(name='Year', start=2009, end=2017, step=1, sizing_mode='stretch_width')
year
```

```
Traceback (most recent call last):
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/pyviz_comms/__init__.py", line 340, in _handle_msg
    self._on_msg(msg)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/viewable.py", line 472, in _on_msg
    doc.unhold()
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/bokeh/document/document.py", line 776, in unhold
    self.callbacks.unhold()
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/bokeh/document/callbacks.py", line 431, in unhold
    self.trigger_on_change(event)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/bokeh/document/callbacks.py", line 408, in trigger_on_change
    invoke_with_curdoc(doc, event.callback_invoker)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/bokeh/document/callbacks.py", line 443, in invoke_with_curdoc
    return f()
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/bokeh/util/callback_manager.py", line 185, in invoke
    callback(attr, old, new)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/reactive.py", line 468, in _comm_change
    state._handle_exception(e)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/io/state.py", line 436, in _handle_exception
    raise exception
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/reactive.py", line 466, in _comm_change
    self._schedule_change(doc, comm)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/reactive.py", line 448, in _schedule_change
    self._change_event(doc)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/reactive.py", line 444, in _change_event
    self._process_events(events)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/reactive.py", line 383, in _process_events
    self.param.update(**self_events)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/param/parameterized.py", line 1902, in update
    self._batch_call_watchers()
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/param/parameterized.py", line 2063, in _batch_call_watchers
    self._execute_watcher(watcher, events)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/param/parameterized.py", line 2025, in _execute_watcher
```

```

    watcher.fn(*args, **kwargs)
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/streams.py", line 760, in _watcher
    self.trigger([self])
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/streams.py", line 186, in trigger
    subscriber(**dict(union))
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/plotting/plot.py", line 232, in refresh
    raise e
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/plotting/plot.py", line 228, in refresh
    self._trigger_refresh(stream_key)
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/plotting/plot.py", line 245, in _trigger_refresh
    self.update(key)
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/plotting/plot.py", line 943, in update
    return self.initialize_plot()
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/plotting/bokeh/element.py", line 2754, in initialize_plot
    raise SkipRendering('All Overlays empty, cannot initialize plot.')
holoviews.core.options.SkipRendering: All Overlays empty, cannot initialize plot.

```

Out[26]: Year: 2009



In [27]: *# Step 3: can choose to visualize either the number of projects or the number of success projects*

```

situation = pn.widgets.RadioButtonGroup(
    name = 'Status',
    options = ['project_count', 'success_project_count'],
    button_type='success',
    width_policy='max',
    sizing_mode='stretch_width'
)

```

situation

Out[27]:

project_count

success_project_count

In [28]: *# combine*

```

dashboard = (idf[(idf.year_launched == year) & (idf.country == country)]
              .groupby(['country', 'year_launched', 'month_launched'])[situation].sum()
              .to_frame()
              .reset_index()
              .reset_index(drop=True))

```



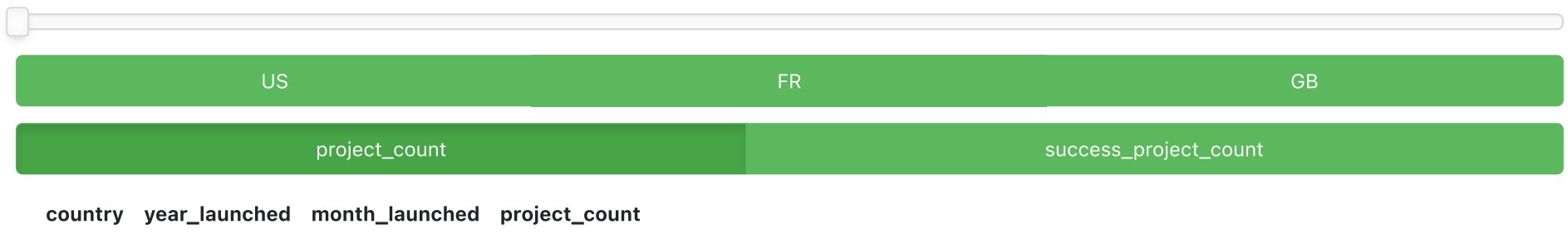
```
Traceback (most recent call last):
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/pyviz_comms/__init__.py", line 340, in _handle
_msg
    self._on_msg(msg)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/viewable.py", line 472, in _on_msg
    doc.unhold()
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/bokeh/document/document.py", line 776, in unho
ld
    self.callbacks.unhold()
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/bokeh/document/callbacks.py", line 431, in unh
old
    self.trigger_on_change(event)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/bokeh/document/callbacks.py", line 408, in tri
gger_on_change
    invoke_with_curdoc(doc, event.callback_invoker)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/bokeh/document/callbacks.py", line 443, in inv
oke_with_curdoc
    return f()
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/bokeh/util/callback_manager.py", line 185, in
invoke
    callback(attr, old, new)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/reactive.py", line 468, in _comm_change
    state._handle_exception(e)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/io/state.py", line 436, in _handle_excep
tion
    raise exception
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/reactive.py", line 466, in _comm_change
    self._schedule_change(doc, comm)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/reactive.py", line 448, in _schedule_cha
nge
    self._change_event(doc)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/reactive.py", line 444, in _change_event
    self._process_events(events)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/panel/reactive.py", line 383, in _process_even
ts
    self.param.update(**self_events)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/param/parameterized.py", line 1902, in update
    self._batch_call_watchers()
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/param/parameterized.py", line 2063, in _batch_
call_watchers
    self._execute_watcher(watcher, events)
  File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/param/parameterized.py", line 2025, in _execut
e_watcher
```

```

    watcher.fn(*args, **kwargs)
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/streams.py", line 760, in _watcher
    self.trigger([self])
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/streams.py", line 186, in trigger
    subscriber(**dict(union))
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/plotting/plot.py", line 232, in refresh
    raise e
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/plotting/plot.py", line 228, in refresh
    self._trigger_refresh(stream_key)
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/plotting/plot.py", line 245, in _trigger_refresh
    self.update(key)
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/plotting/plot.py", line 943, in update
    return self.initialize_plot()
File "/Users/yumi/anaconda3/envs/homl3/lib/python3.10/site-packages/holoviews/plotting/bokeh/element.py", line 2754, in initialize_plot
    raise SkipRendering('All Overlays empty, cannot initialize plot.')
holoviews.core.options.SkipRendering: All Overlays empty, cannot initialize plot.

```

Out[28]: Year: 2009

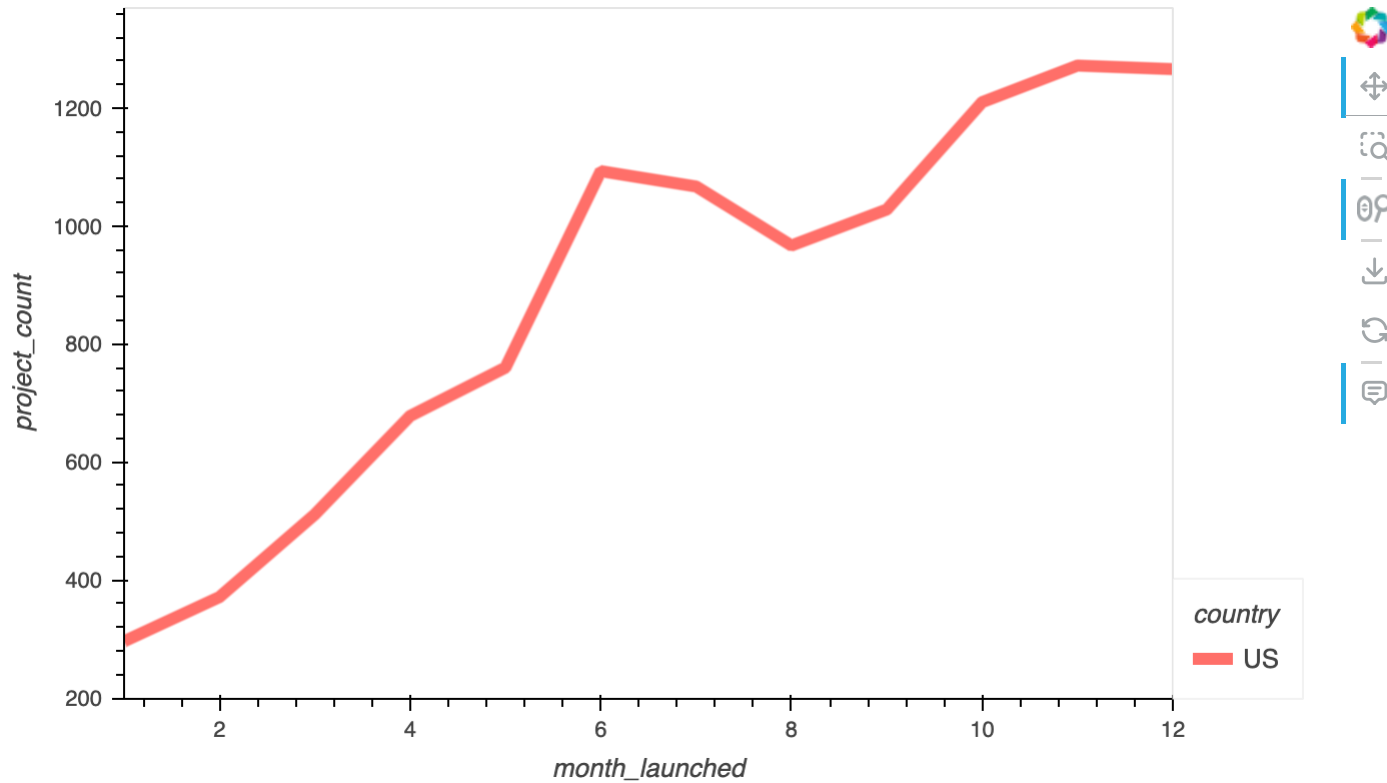
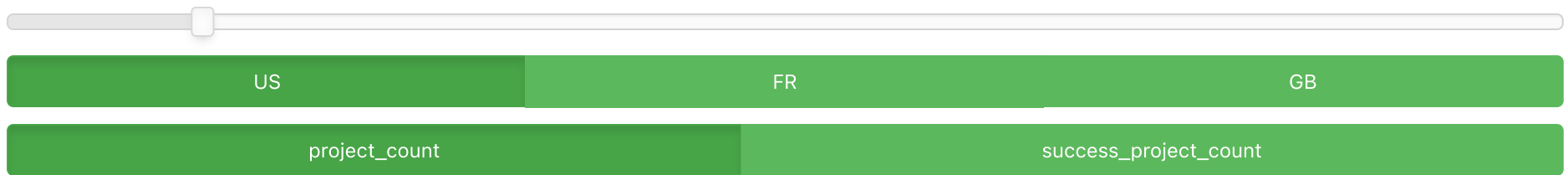


```

In [41]: PALETTE = ["#ff6f69", "#ffcc5c", "#88d8b0"]
ihvplot = dashboard.hvplot(x='month_launched', y=situation, by='country', color=PALETTE, line_width=6, height=400)
ihvplot

```


Out [41]: Year: 2010



```
In [42]: def environment():
    try:
        get_ipython()
        print('notebook')
        return str(get_ipython())
    except:
        print('server')
        return 'server'

env = environment()

if env=="server":
    theme="fast"
else:
```

theme="simple"

```
pn.extension('tabulator', sizing_mode="stretch_width")
hv.extension('bokeh')
```

notebook

```
In [43]: # create table
itable = dashboard.pipe(pn.widgets.Tabulator, pagination='remote', page_size=10, theme=theme)
itable
```

Out [43]: Year: 2010

US			FR		GB	
project_count				success_project_count		
index ▲	country ▲	year_launched ▲	month_launched ▲	project_count ▲		
0	US	2,010	1	296		
1	US	2,010	2	371		
2	US	2,010	3	511		
3	US	2,010	4	678		
4	US	2,010	5	760		
5	US	2,010	6	1,093		
6	US	2,010	7	1,067		
7	US	2,010	8	967		
8	US	2,010	9	1,028		
9	US	2,010	10	1,210		

First

Prev

1

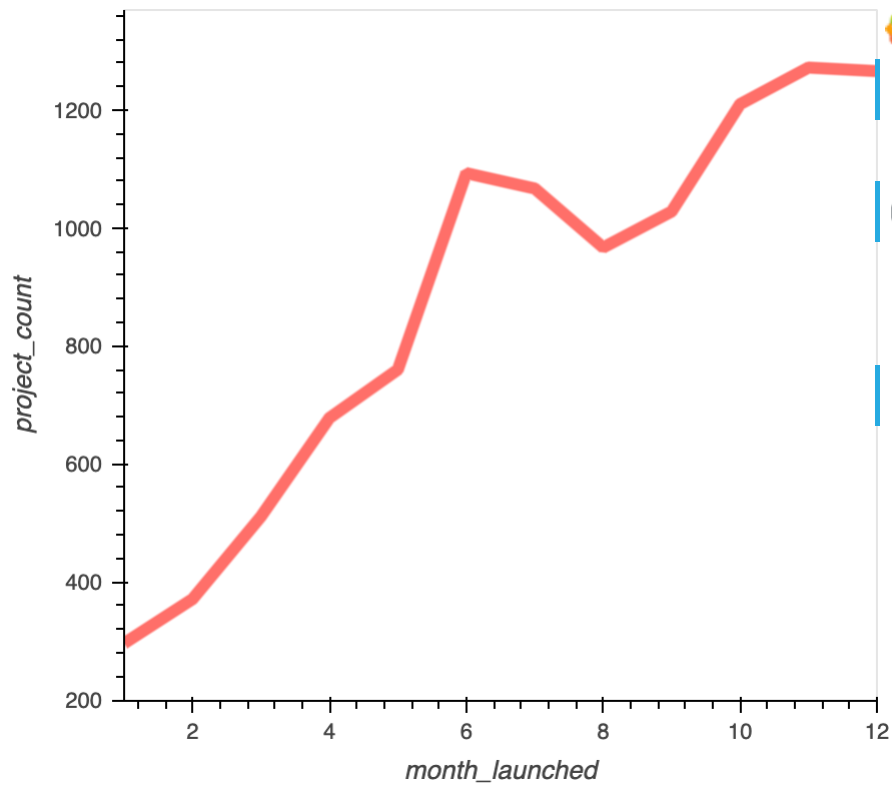
2

Next

Last

```
In [45]: final_dashboard = pn.Column(pn.Row(year, country, situation), pn.Row(ihvplot.panel(),itable.panel()))
         final_dashboard.servable()
```

Out[45]: Year: 2010



index	country	year_launched	month_launched
0	US	2,010	
1	US	2,010	
2	US	2,010	
3	US	2,010	
4	US	2,010	
5	US	2,010	
6	US	2,010	
7	US	2,010	
8	US	2,010	
9	US	2,010	

[First](#)
[Prev](#)
[1](#)
[2](#)
[Next](#)
[Last](#)