Project Title: Surviving the Storm: How Economic Downturns Reshape Investor's Investment Strategies Across the Globe

Project 1

Introduction

This research aims to analyze whether economic downturns, specifically the 2008 financial crisis, have reshaped investors' investment strategies. According to Investment Cycle Theory, it emphasizes that econmic downturns create risks aversion in investors, leading to fewer but more selective investments.(Kenton) This project utilizes a comprehensive Startup Investment dataset from Kaggle, supplemented by a Macro dataset containing information on inflation, interest rates, and unemployment rates across various countries, with the focus on the period from 2000 to 2013. This timeframe allows for a comparative analysis of investment behaviors before, during, and after the crisis.

The null hypothesis of this research question is during the significant economic downturn such as financial crisis, investors will choose the strategy of decreasing the funding amount to the startups. The x-variables will contain both inflation and unemployment rate from macro-economic perspectives, which are crucial defining the concept of economic downturns, and some categorial variables such as country and industry categories of the startups that investor's tend to invest. The categoric variables helped to define whether the strategies caused a change in the characteristics of startups that received higher amount of fundings.

In [191…

```
import pandas as pd
import numpy as np

from IPython.display import display
```

In [191…
```
objects = pd.read_csv("/Users/zzybollar/Downloads/archive/objects.csv")
objects = objects[["id", "entity_type","name", "category_code"]]
objects.groupby("category_code").count().sort_values("id", ascending=False)
objects
```

```
/var/folders/lh/v36j2zlj0t9_xqpxcj3l9qfw0000gn/T/ipykernel_55922/859945751.py:1: DtypeWarning: Columns (3,7,9,10,17,18,21,22,23,25,26,29,30,37) have mixed types. Specify dtype option on import or set low_memory=False.
  objects = pd.read_csv("/Users/zzybollar/Downloads/archive/objects.csv")
```

Out[191…

	id	entity_type	name	category_code
0	c:1	Company	Wetpaint	web
1	c:10	Company	Flektor	games_video
2	c:100	Company	There	games_video
3	c:10000	Company	MYWEBBO	network_hosting
4	c:10001	Company	THE Movie Streamer	games_video
...	...	...	...	...
462646	r:9995	Product	SiteLink, listing feed for Brokerages	NaN
462647	r:9996	Product	EDCLink, listing feed for Economic Development...	NaN
462648	r:9997	Product	Cmail, broadcast email marketing	NaN
462649	r:9998	Product	CatylistCRM, contact database	NaN
462650	r:9999	Product	Catylist Custom Print/Web Design	NaN

462651 rows × 4 columns

In [191…
```
funds = pd.read_csv("/Users/zzybollar/Downloads/archive/funds.csv")
offices = pd.read_csv("/Users/zzybollar/Downloads/archive/offices.csv")
```

In [191…
```
investments = pd.read_csv("/Users/zzybollar/Downloads/archive/investments.csv")
investments = investments.dropna()
investments = investments.dropna(subset=['id', 'investor_object_id'])
investments = investments.drop(columns=['created_at', 'updated_at'])
```

```
investments = investments.merge(funds[['id', 'name']], on='id', how='left')
```

In [191…
```
ipos = pd.read_csv("/Users/zzybollar/Downloads/archive/ipos.csv")
ipos = ipos.dropna(subset=['public_at'])
ipos = ipos.drop(columns=['source_url','source_description','created_at', 'updated_at'])
ipos['public_year'] = pd.to_datetime(ipos['public_at'])
ipos['public_year'] = ipos['public_year'].dt.year
```

In [191…
```
funding_rounds = pd.read_csv("/Users/zzybollar/Downloads/archive/funding_rounds.csv")
```

In [191…
```
inflation_interest_unemployment = pd.read_csv("/Users/zzybollar/Downloads/archive/inflation interest unemployment.c
```

In [191…
```
milestones = pd.read_csv("/Users/zzybollar/Downloads/archive/milestones.csv")
```

In [192…
```
relationships = pd.read_csv("/Users/zzybollar/Downloads/archive/relationships.csv")
```

In [192…
```
wacc_global= pd.read_csv("/Users/zzybollar/Downloads/waccGlobal.csv")
wacc_global.columns=["Industry Name", "Number of Firms","Beta", "Cost of Equity", "E/(D+E)", "Std Dev in Stock", "C
```

Data Cleaning

In [192…
```
inflation_interest_unemployment['year'] = inflation_interest_unemployment['year'].astype(int)
inflation_interest_unemployment_2000_to_2013 = inflation_interest_unemployment[(inflation_interest_unemployment['ye
inflation_interest_unemployment_2000_to_2013 = inflation_interest_unemployment_2000_to_2013.drop(columns=['Deposit
inflation_interest_unemployment_2000_to_2013.dropna(inplace=True)
inflation_interest_unemployment_2000_to_2013.reset_index(drop=True, inplace=True)
```

In [192…
```
funding_rounds = pd.read_csv("/Users/zzybollar/Downloads/archive/funding_rounds.csv")
funding_rounds = funding_rounds.dropna()

funding_rounds['id'] = funding_rounds['id'].astype(str)
funds['id'] = funds['id'].astype(str)

funds = funds.merge(funding_rounds[['id', 'raised_amount_usd']], on='id', how='left')
funds = funds.drop(columns=['source_url', 'source_description', 'created_at', 'updated_at'])
```

In [192…
```
offices_cleaned = offices.dropna(subset=['city', 'region', 'zip_code','state_code'])
```

In [192…
```
funds['funded_at'] = pd.to_datetime(funds['funded_at'], errors='coerce')

start_date = "2000-01-01"
end_date = "2013-12-31"

funding_data = funds[(funds['funded_at'] >= start_date) & (funds['funded_at'] <= end_date)].copy()
funding_data = funding_data[["id"
                              , "fund_id", "object_id", "name", "funded_at", "raised_amount", "raised_currency_code"
funding_data.dropna(subset=["raised_amount"], inplace=True)
funding_data["raised_amount_log"] = np.log1p(funding_data["raised_amount"])
funding_data.reset_index(drop=True, inplace=True)
```

In [192…
```
wacc_global['Industry Name'] = wacc_global['Industry Name'].str.lower()
wacc_global['Industry Name'] = wacc_global['Industry Name'].str.replace(' ', '_')
wacc_global['Industry Name'] = wacc_global['Industry Name'].str.replace('-', '_')
wacc_global['Industry Name'] = wacc_global['Industry Name'].str.replace('&', '_')
wacc_global['Industry Name'] = wacc_global['Industry Name'].str.replace('(', '_')
wacc_global['Industry Name'] = wacc_global['Industry Name'].str.replace(')', '_')
wacc_global = wacc_global.dropna()
wacc_global = wacc_global.drop(index=17)
```

Merge the data

In [192…
```
funding_rounds['id'] = funding_rounds['id'].astype(str)
merged_data1 = funding_rounds.merge(objects, left_on="object_id", right_on="id", how="left")
merged_data1 = merged_data1.drop(columns=["id_y","source_url","source_description","created_at","updated_at",
                                          "pre_money_valuation_usd","post_money_valuation_usd","post_money_valuatio
                                          "pre_money_valuation","created_by"])
merged_data1.drop_duplicates(inplace=True)
merged_data1['year'] = pd.to_datetime(merged_data1['funded_at']).dt.year
merged_data1 = merged_data1[(merged_data1['year'] >= 2000) & (merged_data1['year'] <= 2013)]
```

In [192…
```
merged_data2 = offices.merge(inflation_interest_unemployment_2000_to_2013, left_on='country_code', right_on='iso3c'
merged_data2.dropna(subset=['year'], inplace=True)
merged_data2.drop(columns=['office_id','description','state_code','iso3c','created_at','updated_at'], inplace=True)
```

```
merged_data2['year'] = merged_data2['year'].astype(int)
merged_data2 = merged_data2[(merged_data2['year'] >= 2000) & (merged_data2['year'] <= 2013)]
merged_data2.dropna(subset=['year'], inplace=True)
merged_data2.drop_duplicates("country_code", inplace=True)
```

In [192…

```
merged_data = funding_rounds.merge(investments, on="funding_round_id", how="left", suffixes=('_funding', '_investme

merged_data = merged_data.merge(objects, left_on="funded_object_id", right_on="id", how="left", suffixes=('', '_obj

merged_data = merged_data.merge(funds, on="object_id", how="left", suffixes=('', '_fund'))

merged_data = merged_data.merge(milestones, on="object_id", how="left", suffixes=('', '_milestone'))

merged_data = merged_data.merge(offices, on="object_id", how="left", suffixes=('', '_office'))

merged_data = merged_data.merge(ipos, on="object_id", how="left", suffixes=('', '_ipo'))
merged_data.drop(columns=['public_at','stock_symbol','public_year'], inplace=True)
merged_data.drop_duplicates("object_id", inplace=True)
```

In [193…

```
merged_data_combined = merged_data.merge(merged_data2, on='country_code', how='inner')
venture_funding = merged_data_combined[merged_data_combined['funding_round_type'] == 'venture']
venture_funding_sum = venture_funding.groupby('funding_round_type')['raised_amount'].sum().reset_index()
merged_data_combined['venture_funding'] = merged_data_combined['funding_round_type'].apply(lambda x: 1 if x == 'ven
merged_data_combined['venture_funding_sum'] = merged_data_combined['raised_amount'].apply(lambda x: x if x > 0 else
merged_data_combined['venture_funding_sum'] = merged_data_combined['venture_funding_sum'].fillna(0)
merged_data_combined['venture_funding_sum'] = merged_data_combined['venture_funding_sum'].astype(int)
merged_data_combined['venture_funding_sum'] = merged_data_combined['venture_funding_sum'].apply(lambda x: np.log1p(
```

In [193…

```
merged_data_combined['raised_amount_log'] = np.log1p(merged_data_combined['raised_amount'])
merged_data_combined['raised_amount_log'] = merged_data_combined['raised_amount_log'].fillna(0)
merged_data_combined['raised_amount_log'] = merged_data_combined['raised_amount_log'].astype(int)
merged_data_combined['raised_amount_log'] = merged_data_combined['raised_amount_log'].apply(lambda x: np.log1p(x))
```

In [193…

```
from thefuzz import process

def fuzzy_match(industry, industry_list):
    match, score = process.extractOne(industry, industry_list)
    return match if score >= 50 else None
```

```
merged_data_combined["category_code"] = merged_data_combined["category_code"].astype(str)

merged_data_combined["category_code"] = merged_data_combined["category_code"].apply(lambda x: fuzzy_match(x, wacc_g
merged_data_combined = merged_data_combined.merge(wacc_global, left_on="category_code", right_on="Industry Name", h
merged_data_combined.drop(columns=['Industry Name'], inplace=True)
merged_data_combined
```

Out[193…

	id_funding	funding_round_id	object_id_x	funded_at	funding_round_type	funding_round_code	raised_amount_usd	raised_
0	270	270	c:322	2010-11-16	series-a	a	0.0	
1	1042	1042	c:1448	2007-11-01	series-c+	c	25000000.0	250
2	1525	1525	c:2235	2007-01-15	series-a	a	3700000.0	37
3	2372	2372	c:5693	2008-05-22	series-a	a	1000000.0	10
4	3666	3666	c:12408	2008-08-01	series-a	a	4000000.0	40
...	...	...	...	...	...	...	...	
439	57853	57853	c:236066	2013-12-11	series-a	a	8000000.0	80
440	57866	57866	c:51014	2013-12-12	venture	unattributed	0.0	
441	57867	57867	c:286063	2013-12-12	venture	unattributed	0.0	
442	57868	57868	c:286065	2013-12-12	venture	unattributed	0.0	
443	57876	57876	c:286114	2010-01-29	venture	partial	100000.0	1

444 rows × 97 columns

Summary Statistics tables by time period (before, during and after crisis)

By comparing the summary statistics of pre-, during-, and post-crisis total raised amount, it's not hard to see that the mid 50th percentile have significant increase after crisis. But it could not represent that directly that this significant increase is due to the economic downturns, but we need to take other reasons into consideration such as the lack of data for before-, and during-crisis period.

In [193…
```
before_crisis_end = "2007-09-30"
during_crisis_start = "2007-10-01"
during_crisis_end = "2009-06-30"
after_crisis_start = "2009-07-01"
```

In [193…
```
funding_before_crisis = merged_data_combined[(merged_data_combined['funded_at'] >= "2000-01-01") & (merged_data_com
funding_during_crisis = merged_data_combined[(merged_data_combined['funded_at'] >= during_crisis_start) & (merged_d
funding_after_crisis = merged_data_combined[(merged_data_combined['funded_at'] >= after_crisis_start) & (merged_dat
```

In [193…
```
def drop_duplicate_columns(df):
    duplicate_columns = df.columns[df.columns.duplicated()]
    df = df.loc[:, ~df.columns.duplicated()]
    return df

funding_before_crisis = drop_duplicate_columns(funding_before_crisis)
funding_during_crisis = drop_duplicate_columns(funding_during_crisis)
funding_after_crisis = drop_duplicate_columns(funding_after_crisis)
```

In [193…
```
funding_before_crisis.describe().drop(columns=
                                      ['fund_id','funded_at_fund','raised_amount_fund','pre_money_valuation_usd', '
                                       'post_money_valuation_usd', 'post_money_valuation'])
```

Out[193...

	funding_round_id	raised_amount_usd	raised_amount	participants	is_first_round	is_last_round	id_investment	raised_am
count	5.000000	5.000000e+00	5.000000e+00	5.000000	5.000000	5.000000	5.00000	
mean	10430.400000	1.250000e+07	1.250000e+07	2.400000	0.600000	0.600000	15303.00000	
min	1525.000000	3.700000e+06	3.700000e+06	1.000000	0.000000	0.000000	2269.00000	
25%	9252.000000	6.000000e+06	6.000000e+06	1.000000	0.000000	0.000000	13762.00000	
50%	11948.000000	1.000000e+07	1.000000e+07	3.000000	1.000000	1.000000	16993.00000	
75%	13500.000000	1.800000e+07	1.800000e+07	3.000000	1.000000	1.000000	19510.00000	
max	15927.000000	2.480000e+07	2.480000e+07	4.000000	1.000000	1.000000	23981.00000	
std	5536.987927	8.767554e+06	8.767554e+06	1.341641	0.547723	0.547723	8187.22282	

8 rows × 31 columns

In [193...

```
funding_during_crisis.describe().drop(columns=
                                       ['fund_id','funded_at_fund','raised_amount_fund','pre_money_valuation_usd', '
                                        'post_money_valuation_usd', 'post_money_valuation'])
```

Out[193...

	funding_round_id	raised_amount_usd	raised_amount	participants	is_first_round	is_last_round	id_investment	raised_am
count	14.000000	1.400000e+01	1.400000e+01	14.000000	14.000000	14.000000	14.000000	
mean	9514.000000	2.605571e+07	2.605571e+07	2.285714	0.428571	0.642857	12386.214286	
min	1042.000000	1.000000e+06	1.000000e+06	1.000000	0.000000	0.000000	1526.000000	
25%	4502.250000	6.335000e+06	6.335000e+06	1.000000	0.000000	0.000000	6021.500000	
50%	6197.000000	1.000000e+07	1.000000e+07	2.000000	0.000000	1.000000	8725.500000	
75%	7666.250000	2.400000e+07	2.400000e+07	3.000000	1.000000	1.000000	11389.500000	
max	51998.000000	1.030000e+08	1.030000e+08	5.000000	1.000000	1.000000	58143.000000	
std	12642.757702	3.356773e+07	3.356773e+07	1.266647	0.513553	0.497245	13976.935986	

8 rows × 31 columns

In [193...

```
funding_after_crisis.describe().drop(columns=
                                       ['fund_id','funded_at_fund','raised_amount_fund','pre_money_valuation_usd', '
                                        'post_money_valuation_usd', 'post_money_valuation'])
```

Out[193...

	funding_round_id	raised_amount_usd	raised_amount	participants	is_first_round	is_last_round	id_investment	raised_am
count	425.000000	4.250000e+02	4.250000e+02	425.000000	425.000000	425.000000	326.000000	
mean	44068.527059	1.364969e+07	1.364767e+07	1.343529	0.851765	0.863529	55928.549080	
min	270.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	453.000000	
25%	38907.000000	0.000000e+00	0.000000e+00	1.000000	1.000000	1.000000	51899.000000	
50%	44095.000000	1.000000e+06	1.000000e+06	1.000000	1.000000	1.000000	56109.000000	
75%	52019.000000	8.000000e+06	8.000000e+06	2.000000	1.000000	1.000000	60892.500000	
max	57876.000000	7.000000e+08	7.000000e+08	15.000000	1.000000	1.000000	80795.000000	
std	9546.636862	5.584922e+07	5.584963e+07	1.488830	0.355752	0.343692	10508.296439	

8 rows × 31 columns

Plots, Figures

In [193...

```
import matplotlib.pyplot as plt
import statsmodels.formula.api as sm
import qeds
import seaborn as sns
import numpy as np
```

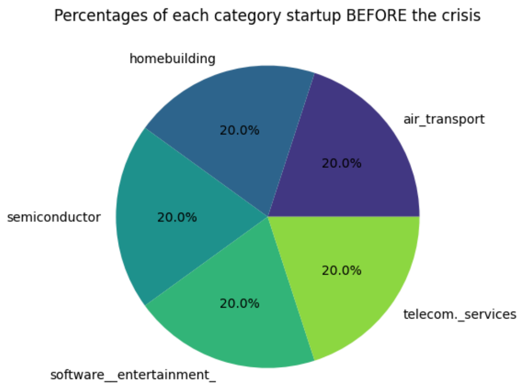
Pie Chart

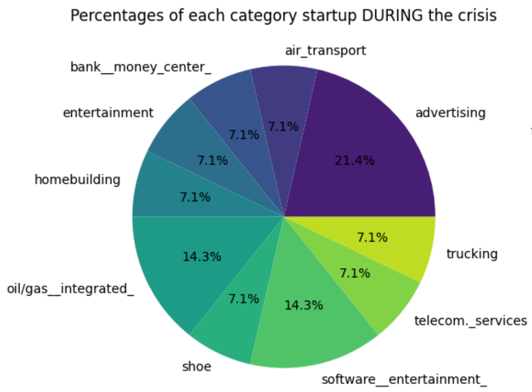
    "Percentages of each category startup BEFORE/DURING/AFTER crisis" clearly shows the proportional
    distribution of each startup categories in the whole market. It provides a clear visualization of

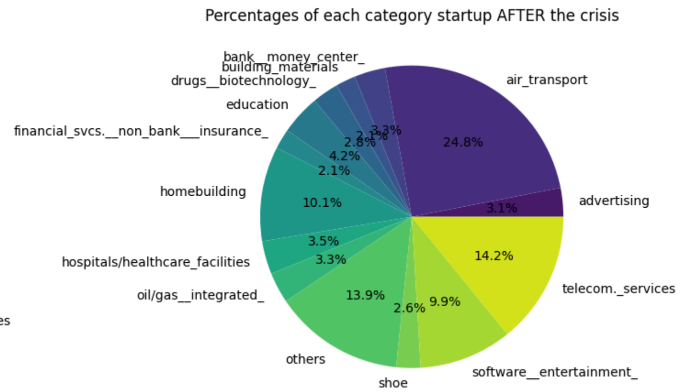
how different categories contribute to the overall startup economic system and could easily see the percentage change in 3 different time period. But there is a limitation that the whole market size are not shown, and it is undoubt that the datasize and whole market size are not the same.

In [194…

```
def calculate_category_percenrtages(data):
    category_percentages = data["category_code"].value_counts(normalize=True) * 100
    category_percentages = category_percentages.reset_index()
    category_percentages.columns = ["category_code", "percentage"]
    category_percentages['category_code'] = category_percentages.apply(lambda row: 'others' if row['percentage'] <
    category_percentages = category_percentages.groupby('category_code')['percentage'].sum().reset_index()
    return category_percentages

category_percentages_before_crisis = calculate_category_percenrtages(funding_before_crisis)
category_percentages_during_crisis = calculate_category_percenrtages(funding_during_crisis)
category_percentages_after_crisis = calculate_category_percenrtages(funding_after_crisis)

def plot_pie_chart(ax, data, title):
    ax.pie(data["percentage"], labels=data["category_code"], autopct='%1.1f%%', colors=sns.color_palette("viridis",
    ax.set_title(title)

fig, ax = plt.subplots(1, 3, figsize=(20, 6))

plot_pie_chart(ax[0],category_percentages_before_crisis, "Percentages of each category startup BEFORE the crisis")
plot_pie_chart(ax[1],category_percentages_during_crisis, "Percentages of each category startup DURING the crisis")
plot_pie_chart(ax[2],category_percentages_after_crisis, "Percentages of each category startup AFTER the crisis")

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





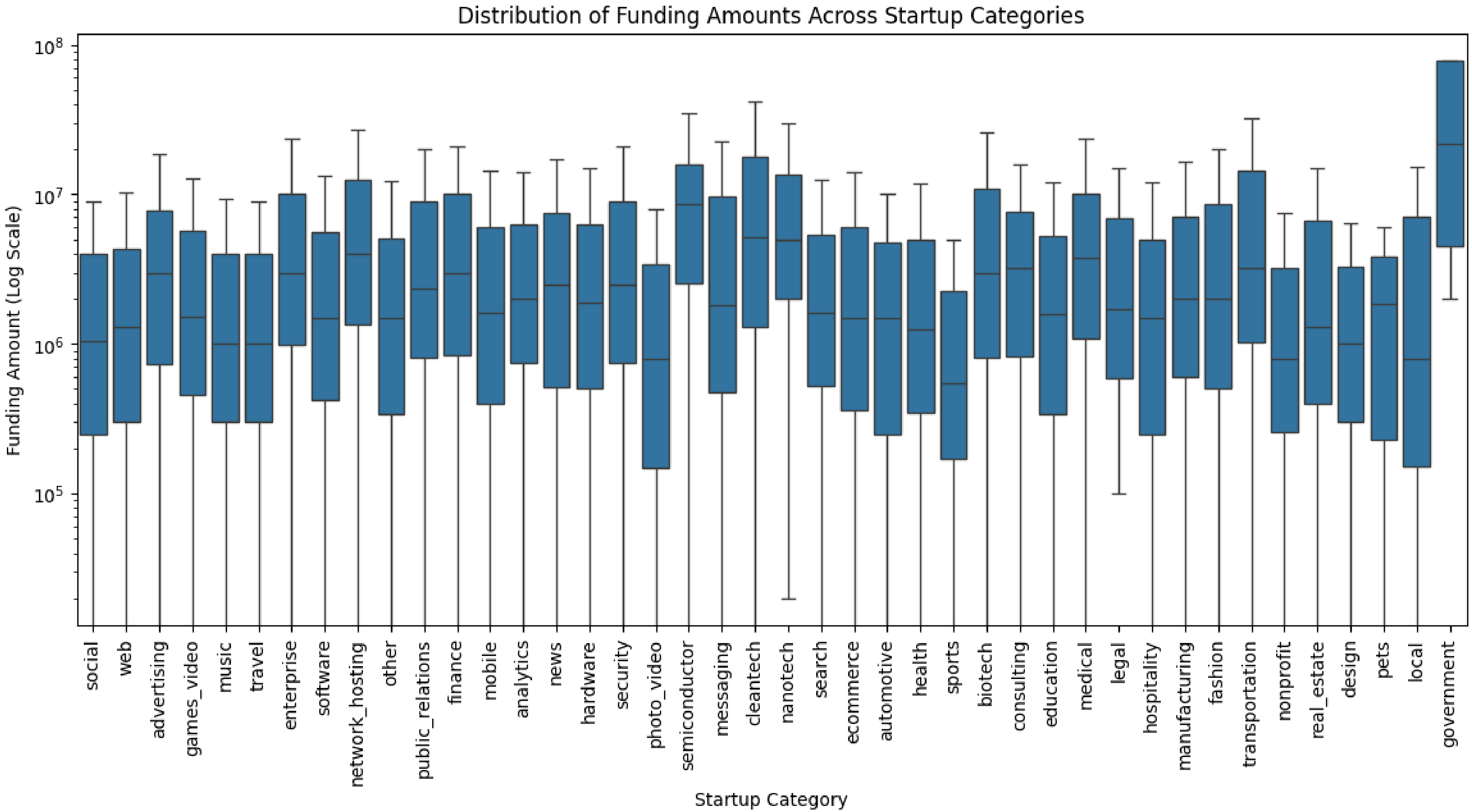


Box Plot

"Distribution of funding amounts across startup categories" is a box plot of the distribution of funding amounts. This plot could know the average funding performance of each industry. It contains more details of mean, median, percentile information than the previous two visualizations. The variable for x and y-axis is appropriate since it is not hard to compare and see the difference. This graph concludes the statistics of mean, median and distribution clearly, which is helpful to conclude the funding performance of different categories. The higher funding amount of a category might due to larger number of firms in that specific category, or extreme high raised funding of top startups in that specific category.

In [194…

```
plt.figure(figsize=(14, 6))
sns.boxplot(x="category_code", y="raised_amount", data=merged_data1, showfliers=False)
plt.xticks(rotation=90)
plt.yscale("log")
plt.title("Distribution of Funding Amounts Across Startup Categories")
plt.xlabel("Startup Category")
plt.ylabel("Funding Amount (Log Scale)")
plt.show()
```



Line plot

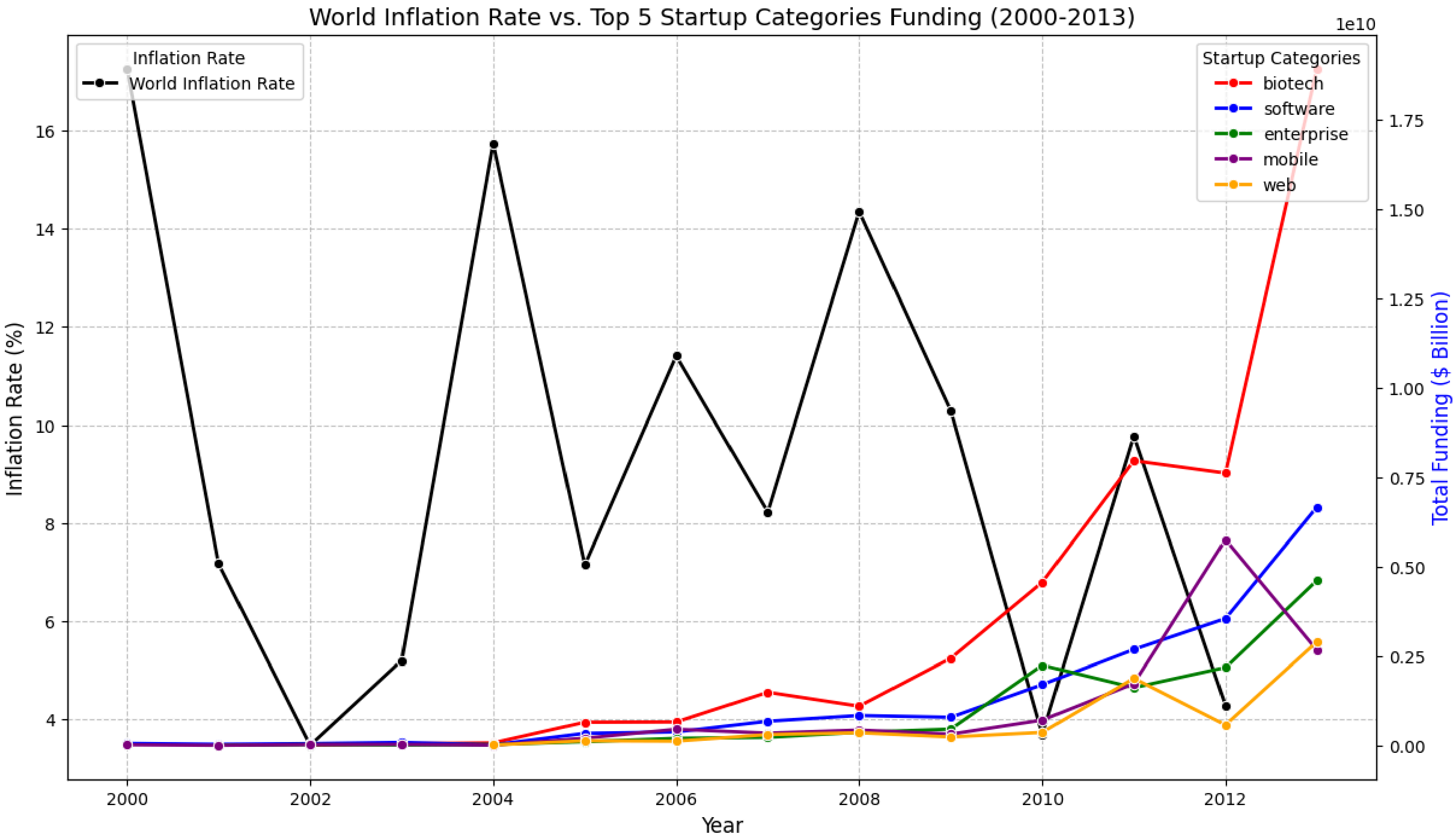
The line plot clearly shows the flunctuation of inflation, and top 5 startup categories funding. Putting the change in total funding and inflation together allows for a clearer look at both at the same time.

In [194…

```
world_inflation_by_year = merged_data2.groupby('year')['Inflation, GDP deflator (annual %)'].mean().reset_index()
```

In [194…

```
merged_data1['funded_at'] = pd.to_datetime(merged_data1['funded_at'], errors='coerce')

merged_data1['year'] = merged_data1['funded_at'].dt.year

top_5_categories = merged_data1['category_code'].value_counts(normalize=True).nlargest(5).index

funding_by_category = merged_data1[merged_data1['category_code'].isin(top_5_categories)]
funding_trends = funding_by_category.groupby(['year', 'category_code'])['raised_amount'].sum().reset_index()

fig, ax1 = plt.subplots(figsize=(14, 8))

ax1.set_title('World Inflation Rate vs. Top 5 Startup Categories Funding (2000-2013)', fontsize=14)
ax1.set_xlabel('Year', fontsize=12)
ax1.set_ylabel('Inflation Rate (%)', fontsize=12, color='black')
sns.lineplot(data=world_inflation_by_year, x='year', y='Inflation, GDP deflator (annual %)',
             marker='o', label='World Inflation Rate', linewidth=2, color='black', ax=ax1)

ax2 = ax1.twinx()
ax2.set_ylabel('Total Funding ($ Billion)', fontsize=12, color='blue')

colors = ['red', 'blue', 'green', 'purple', 'orange']
for i, category in enumerate(top_5_categories):
    category_data = funding_trends[funding_trends['category_code'] == category]
    sns.lineplot(data=category_data, x='year', y='raised_amount', marker='o',
                 label=category, linewidth=2, color=colors[i], ax=ax2)

ax1.legend(loc='upper left', title="Inflation Rate", fontsize=10)
ax2.legend(loc='upper right', title="Startup Categories", fontsize=10)
ax1.grid(True, linestyle="--", alpha=0.7)

plt.show()
```

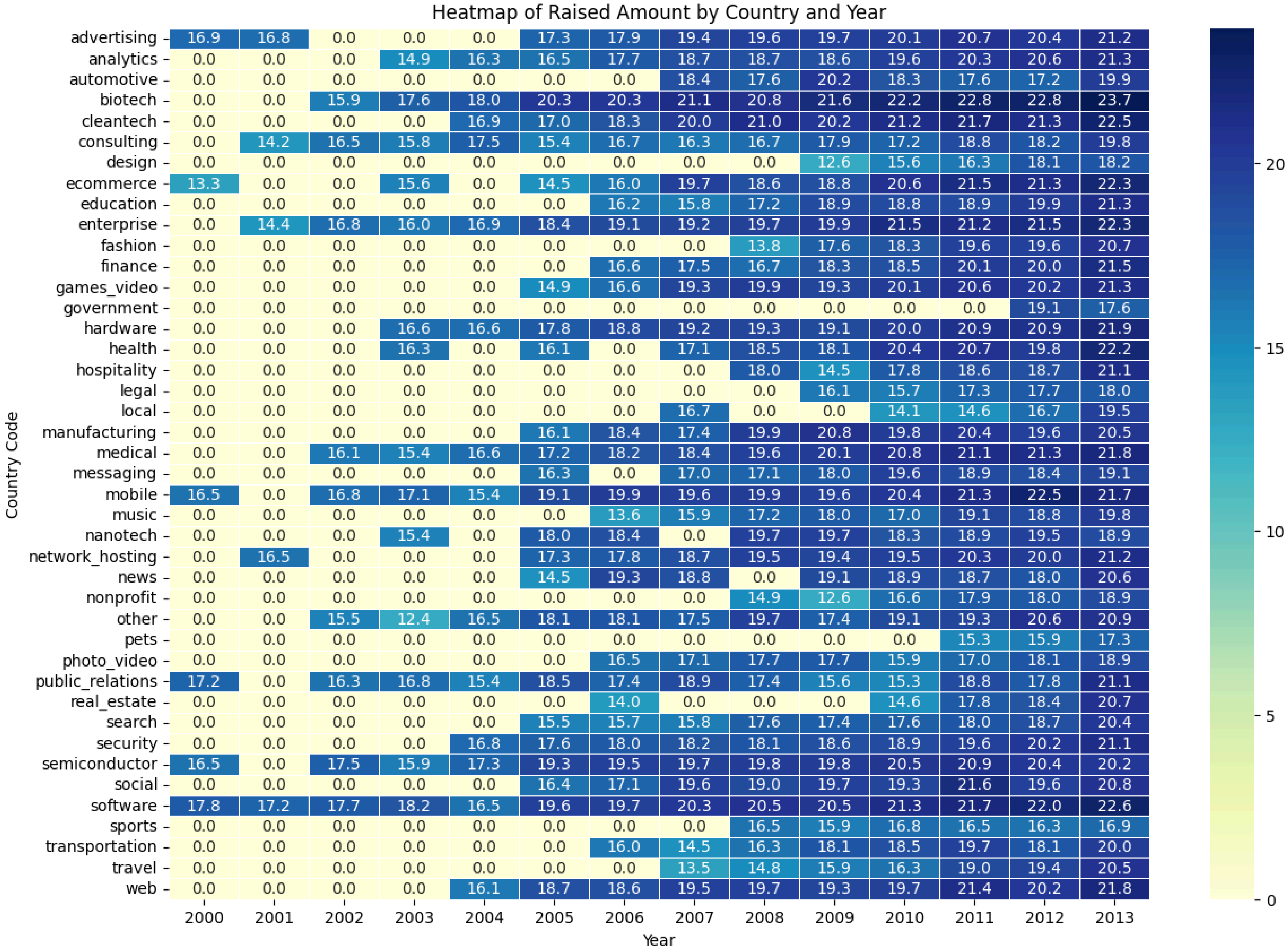

World Inflation Rate vs. Top 5 Startup Categories Funding (2000-2013)
Inflation Rate
World Inflation Rate
Startup Categories
biotech
software
enterprise
mobile
web
Inflation Rate (%)
Total Funding ($ Billion)
1e10
Year
16
14
12
10
8
6
4
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
2000
2002
2004
2006
2008
2010
2012

Heatmap

The heatmap could visualize industry-specific investment trends and detect the impact of 2008 financial crisis. This graph shows that investors stopped investing on several industries during financial crisis period, especially real_estate, whereas some industries even get more investment during that time.

In [194…

```
df=merged_data1
df=df.pivot_table(values='raised_amount', index='category_code', columns='year', aggfunc='sum', fill_value=0)
df = np.log1p(df)
plt.figure(figsize=(14, 10))
sns.heatmap(df, cmap="YlGnBu", linewidths=0.5, annot=True, fmt=".1f")
plt.title("Heatmap of Raised Amount by Country and Year")
plt.xlabel("Year")
plt.ylabel("Country Code")
plt.show()
```


Heatmap of Raised Amount by Country and Year
Country Code
Year
advertising 16.9 16.8 0.0 0.0 0.0 17.3 17.9 19.4 19.6 19.7 20.1 20.7 20.4 21.2
analytics 0.0 0.0 0.0 14.9 16.3 16.5 17.7 18.7 18.7 18.6 19.6 20.3 20.6 21.3
automotive 0.0 0.0 0.0 0.0 0.0 0.0 0.0 18.4 17.6 20.2 18.3 17.6 17.2 19.9
biotech 0.0 0.0 15.9 17.6 18.0 20.3 20.3 21.1 20.8 21.6 22.2 22.8 22.8 23.7
cleantech 0.0 0.0 0.0 0.0 16.9 17.0 18.3 20.0 21.0 20.2 21.2 21.7 21.3 22.5
consulting 0.0 14.2 16.5 15.8 17.5 15.4 16.7 16.3 16.7 17.9 17.2 18.8 18.2 19.8
design 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 12.6 15.6 16.3 18.1 18.2
ecommerce 13.3 0.0 0.0 15.6 0.0 14.5 16.0 19.7 18.6 18.8 20.6 21.5 21.3 22.3
education 0.0 0.0 0.0 0.0 0.0 0.0 16.2 15.8 17.2 18.9 18.8 18.9 19.9 21.3
enterprise 0.0 14.4 16.8 16.0 16.9 18.4 19.1 19.2 19.7 19.9 21.5 21.2 21.5 22.3
fashion 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 13.8 17.6 18.3 19.6 19.6 20.7
finance 0.0 0.0 0.0 0.0 0.0 0.0 16.6 17.5 16.7 18.3 18.5 20.1 20.0 21.5
games_video 0.0 0.0 0.0 0.0 0.0 14.9 16.6 19.3 19.9 19.3 20.1 20.6 20.2 21.3
government 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 19.1 17.6
hardware 0.0 0.0 0.0 16.6 16.6 17.8 18.8 19.2 19.3 19.1 20.0 20.9 20.9 21.9
health 0.0 0.0 0.0 16.3 0.0 16.1 0.0 17.1 18.5 18.1 20.4 20.7 19.8 22.2
hospitality 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 18.0 14.5 17.8 18.6 18.7 21.1
legal 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 16.1 15.7 17.3 17.7 18.0
local 0.0 0.0 0.0 0.0 0.0 0.0 0.0 16.7 0.0 0.0 14.1 14.6 16.7 19.5
manufacturing 0.0 0.0 0.0 0.0 0.0 16.1 18.4 17.4 19.9 20.8 19.8 20.4 19.6 20.5
medical 0.0 0.0 16.1 15.4 16.6 17.2 18.2 18.4 19.6 20.1 20.8 21.1 21.3 21.8
messaging 0.0 0.0 0.0 0.0 0.0 16.3 0.0 17.0 17.1 18.0 19.6 18.9 18.4 19.1
mobile 16.5 0.0 16.8 17.1 15.4 19.1 19.9 19.6 19.9 19.6 20.4 21.3 22.5 21.7
music 0.0 0.0 0.0 0.0 0.0 0.0 13.6 15.9 17.2 18.0 17.0 19.1 18.8 19.8
nanotech 0.0 0.0 0.0 15.4 0.0 18.0 18.4 0.0 19.7 19.7 18.3 18.9 19.5 18.9
network_hosting 0.0 16.5 0.0 0.0 0.0 17.3 17.8 18.7 19.5 19.4 19.5 20.3 20.0 21.2
news 0.0 0.0 0.0 0.0 0.0 14.5 19.3 18.8 0.0 19.1 18.9 18.7 18.0 20.6
nonprofit 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 14.9 12.6 16.6 17.9 18.0 18.9
other 0.0 0.0 15.5 12.4 16.5 18.1 18.1 17.5 19.7 17.4 19.1 19.3 20.6 20.9
pets 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 15.3 15.9 17.3
photo_video 0.0 0.0 0.0 0.0 0.0 0.0 16.5 17.1 17.7 17.7 15.9 17.0 18.1 18.9
public_relations 17.2 0.0 16.3 16.8 15.4 18.5 17.4 18.9 17.4 15.6 15.3 18.8 17.8 21.1
real_estate 0.0 0.0 0.0 0.0 0.0 0.0 14.0 0.0 0.0 0.0 14.6 17.8 18.4 20.7
search 0.0 0.0 0.0 0.0 0.0 15.5 15.7 15.8 17.6 17.4 17.6 18.0 18.7 20.4
security 0.0 0.0 0.0 0.0 16.8 17.6 18.0 18.2 18.1 18.6 18.9 19.6 20.2 21.1
semiconductor 16.5 0.0 17.5 15.9 17.3 19.3 19.5 19.7 19.8 19.8 20.5 20.9 20.4 20.2
social 0.0 0.0 0.0 0.0 0.0 16.4 17.1 19.6 19.0 19.7 19.3 21.6 19.6 20.8
software 17.8 17.2 17.7 18.2 16.5 19.6 19.7 20.3 20.5 20.5 21.3 21.7 22.0 22.6
sports 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 16.5 15.9 16.8 16.5 16.3 16.9
transportation 0.0 0.0 0.0 0.0 0.0 0.0 16.0 14.5 16.3 18.1 18.5 19.7 18.1 20.0
travel 0.0 0.0 0.0 0.0 0.0 0.0 0.0 13.5 14.8 15.9 16.3 19.0 19.4 20.5
web 0.0 0.0 0.0 0.0 16.1 18.7 18.6 19.5 19.7 19.3 19.7 21.4 20.2 21.8
2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013
20
15
10
5
0

Project 2

The Message

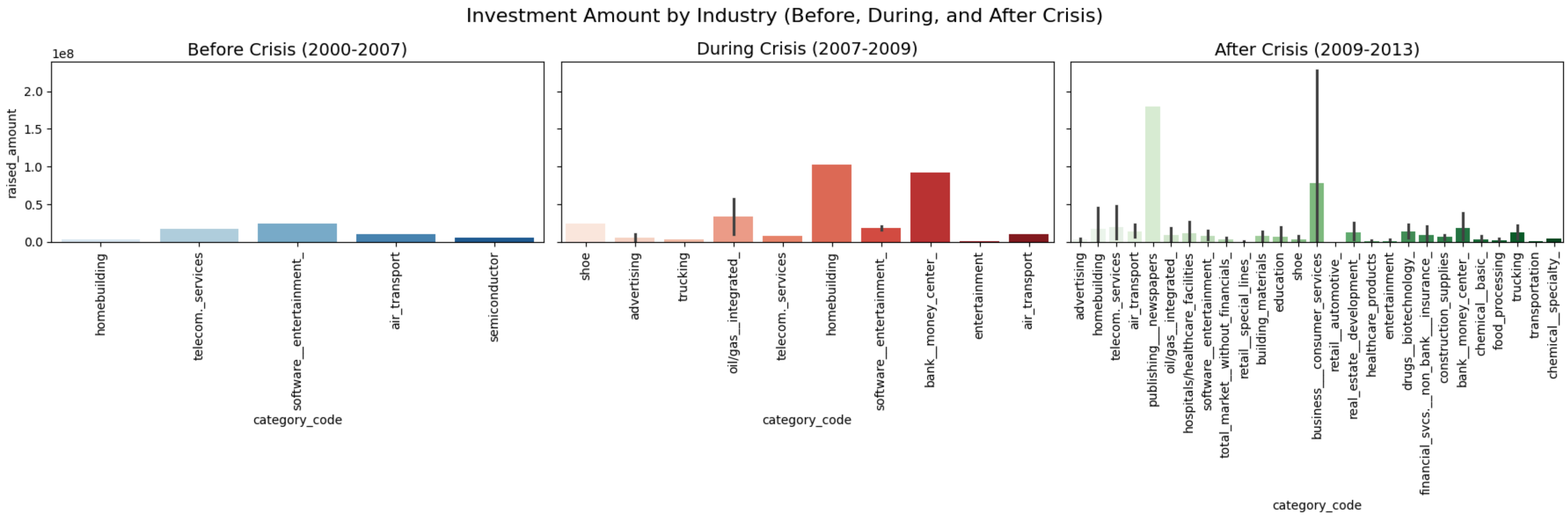
My project is about how economic downturns reshape the investors' investing strategies. The economic downturns might have certain affect on investor's changing their investing strategies but there might also have certain other reasons such as reponse behavior could be non-random. (Author links open overlay panelAnnamaria Conti a b et al., 2018)

In [194…

```
import geopandas as gpd
from shapely.geometry import Point
world = gpd.read_file('/Users/zzybollar/Downloads/110m_cultural/ne_110m_admin_0_countries.shp')
```

In [194…

```
funding_by_category_year = merged_data_combined.groupby(['category_code', 'year'])['raised_amount'].sum().reset_ind

fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)

sns.barplot(data=funding_before_crisis, x='category_code', y='raised_amount', hue='category_code', palette='Blues',
axes[0].set_title('Before Crisis (2000-2007)', fontsize=14)

sns.barplot(data=funding_during_crisis, x='category_code', y='raised_amount', hue='category_code', palette='Reds',
axes[1].set_title('During Crisis (2007-2009)', fontsize=14)

sns.barplot(data=funding_after_crisis, x='category_code', y='raised_amount', hue='category_code', palette='Greens',
axes[2].set_title('After Crisis (2009-2013)', fontsize=14)

for ax in axes:
    ax.set_xticks(range(len(ax.get_xticklabels())))
    ax.set_xticklabels(ax.get_xticklabels(), rotation=90)

plt.suptitle('Investment Amount by Industry (Before, During, and After Crisis)', fontsize=16)
plt.tight_layout()
plt.show()
```



Map and Interpretations

Two groups of maps shows the total raised amount of high-/low- risks industries during three different time period. The geographic distribution and visualization highlights how different regions impact on the investing strategies or decisions of investors', and how investment strategies varied globally. It is not hard to tell that the raised amount for low-risks industries in some specific region has high stability that did not affected by the significant economic downturns. Whereas looking at the map for high-risks industries, we could see that the speed of the recovery of investment amount varies by continents, some area, for example Asia, showing a stronger rebounds. After the economic downturns, more area on the map were colored which might lead to a conclusion of the investor's tend to invest globally, however, this might also due to the lack of the data in some specific timeline such as before-crisis period and in some specific geographic area.

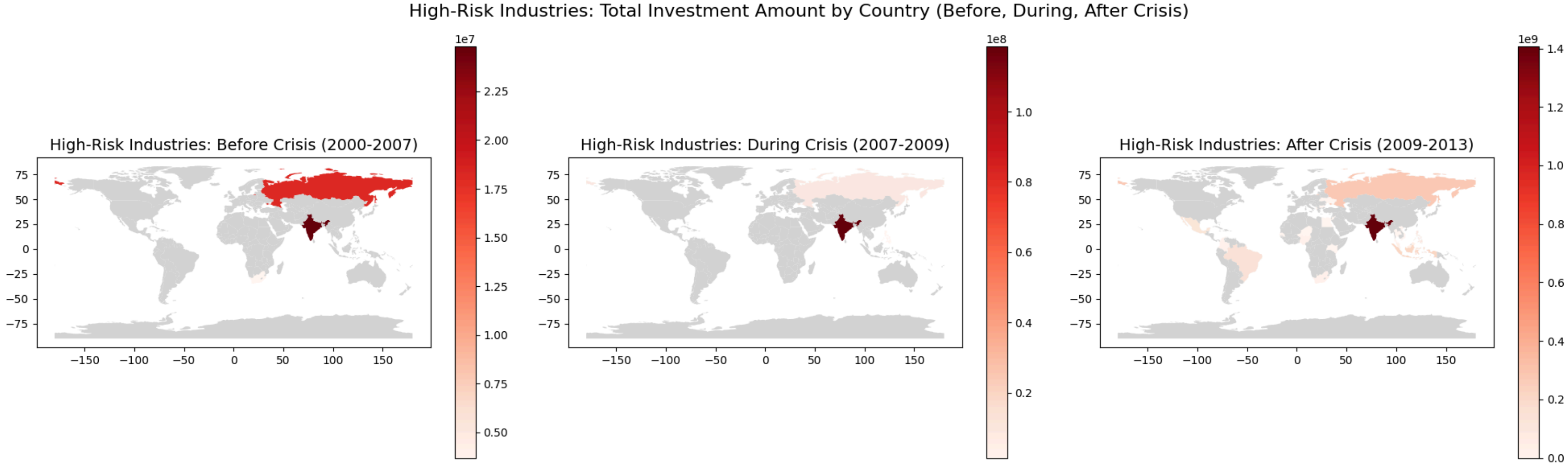
In [194…

```
import geopandas as gpd
from shapely.geometry import Point
world = gpd.read_file('/Users/zzybollar/Downloads/110m_cultural/ne_110m_admin_0_countries.shp')
```

In [194…
```
high_risk_industries = [
    'advertising', 'shoe', 'homebuilding', 'trucking', 'oil/gas__integrated_',
    'telecom._services', 'software__entertainment_', 'bank__money_center_',
    'air_transport', 'publishing___newspapers', 'entertainment', 'semiconductor',
    'real_estate__development_', 'retail__special_lines_', 'retail__automotive_',
    'financial_svcs.__non_bank___insurance_', 'chemical__basic_', 'chemical__specialty_'
]

low_risk_industries = [
    'hospitals/healthcare_facilities', 'education', 'healthcare_products',
    'drugs__biotechnology_', 'construction_supplies', 'food_processing',
    'transportation', 'building_materials', 'business___consumer_services'
]

funding_before_crisis['risk_category'] = funding_before_crisis['category_code'].apply(
        lambda x: 'High-Risk' if x in high_risk_industries else ('Low-Risk' if x in low_risk_industries else 'Other
)
funding_during_crisis['risk_category'] = funding_during_crisis['category_code'].apply(
        lambda x: 'High-Risk' if x in high_risk_industries else ('Low-Risk' if x in low_risk_industries else 'Other
)
funding_after_crisis['risk_category'] = funding_after_crisis['category_code'].apply(
        lambda x: 'High-Risk' if x in high_risk_industries else ('Low-Risk' if x in low_risk_industries else 'Other
)
```

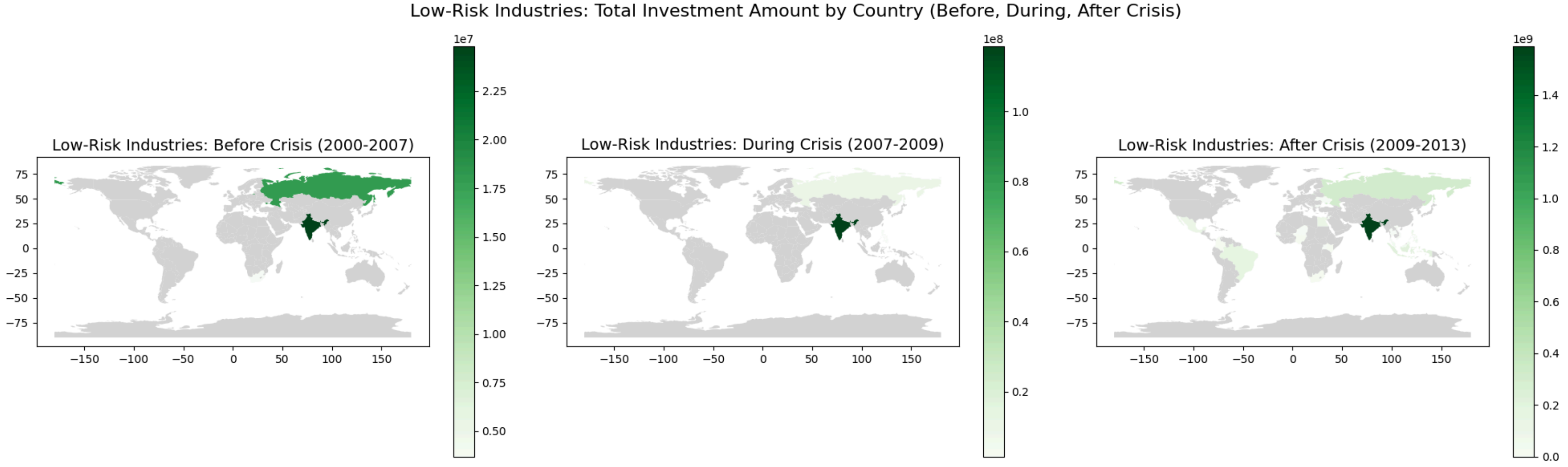
In [194…
```
high_risk_data_before = funding_before_crisis[funding_before_crisis['risk_category'] == 'High-Risk']
high_risk_data_during = funding_during_crisis[funding_during_crisis['risk_category'] == 'High-Risk']
high_risk_data_after = funding_after_crisis[funding_after_crisis['risk_category'] == 'High-Risk']

country_funding_before_crisis = high_risk_data_before.groupby("country_code")["raised_amount"].sum().reset_index()
country_funding_during_crisis = high_risk_data_during.groupby("country_code")["raised_amount"].sum().reset_index()
country_funding_after_crisis = high_risk_data_after.groupby("country_code")["raised_amount"].sum().reset_index()

world_before_crisis = world.merge(country_funding_before_crisis, left_on="SOV_A3", right_on="country_code", how="le
world_during_crisis = world.merge(country_funding_during_crisis, left_on="SOV_A3", right_on="country_code", how="le
world_after_crisis = world.merge(country_funding_after_crisis, left_on="SOV_A3", right_on="country_code", how="left

fig, axes = plt.subplots(1, 3, figsize=(20, 6))

world_before_crisis.plot(column='raised_amount', cmap='Reds', legend=True, ax=axes[0], missing_kwds={'color': 'ligh
axes[0].set_title('High-Risk Industries: Before Crisis (2000-2007)', fontsize=14)
```

```
world_during_crisis.plot(column='raised_amount', cmap='Reds', legend=True, ax=axes[1], missing_kwds={'color': 'ligh
axes[1].set_title('High-Risk Industries: During Crisis (2007-2009)', fontsize=14)

world_after_crisis.plot(column='raised_amount', cmap='Reds', legend=True, ax=axes[2], missing_kwds={'color': 'light
axes[2].set_title('High-Risk Industries: After Crisis (2009-2013)', fontsize=14)

plt.suptitle('High-Risk Industries: Total Investment Amount by Country (Before, During, After Crisis)', fontsize=16
plt.tight_layout()
plt.show()
```



In [ ]:
```
Low_risk_data_before = funding_before_crisis[funding_before_crisis['risk_category'] == 'Low-Risk']
low_risk_data_during = funding_during_crisis[funding_during_crisis['risk_category'] == 'Low-Risk']
low_risk_data_after = funding_after_crisis[funding_after_crisis['risk_category'] == 'Low-Risk']

country_funding_before_crisis1 = Low_risk_data_before.groupby("country_code")["raised_amount"].sum().reset_index()
country_funding_during_crisis1 = low_risk_data_during.groupby("country_code")["raised_amount"].sum().reset_index()
country_funding_after_crisis1 = low_risk_data_after.groupby("country_code")["raised_amount"].sum().reset_index()

world_before_crisis1 = world.merge(country_funding_before_crisis1, left_on="SOV_A3", right_on="country_code", how="
world_during_crisis1 = world.merge(country_funding_during_crisis1, left_on="SOV_A3", right_on="country_code", how="
world_after_crisis1 = world.merge(country_funding_after_crisis1, left_on="SOV_A3", right_on="country_code", how="le
fig, axes = plt.subplots(1, 3, figsize=(20, 6))
```

```
world_before_crisis1.plot(column='raised_amount', cmap='Greens', legend=True, ax=axes[0], missing_kwds={'color': 'l
axes[0].set_title('Low-Risk Industries: Before Crisis (2000-2007)', fontsize=14)

world_during_crisis1.plot(column='raised_amount', cmap='Greens', legend=True, ax=axes[1], missing_kwds={'color': 'l
axes[1].set_title('Low-Risk Industries: During Crisis (2007-2009)', fontsize=14)

world_after_crisis1.plot(column='raised_amount', cmap='Greens', legend=True, ax=axes[2], missing_kwds={'color': 'li
axes[2].set_title('Low-Risk Industries: After Crisis (2009-2013)', fontsize=14)

plt.suptitle('Low-Risk Industries: Total Investment Amount by Country (Before, During, After Crisis)', fontsize=16)
plt.tight_layout()
plt.show()
```



Regressions

This is a regression equation:

$$logRaisedAmount = \beta_0 + \beta_1 \times \text{Macroeconomic interpretations} + \beta_2 \times \text{Categorial variables} + \beta_3 \times \text{Participants} + \beta_4 \times \text{years}$$

Regression table 1: with 4 models generating the Macroeconomic indicators Impacts on Total raised amount

```
Regression table 1 shows how the macro economic indicators such as inflation rate, real interest
rate affect the total raised amount. The result, especially the result for model 8 in this
regression model, suggests a diminishing return effect at higher funding levels. The R^2 of model
6 and 8 almost explains all variance. In summary, in this model we could conclude that the
participants have negative impact in raise fundings, inflation and VC fundings are highly
significant. We could interprets that macroeconomic factors are weak predictors of total funding
amount, in other words, investor's is unlikely to change their strategies by decreasing the
funding amount during significant economic downturns.
```

In [195…

```python
import statsmodels.api as sm
from stargazer.stargazer import Stargazer

merged_data_combined['Inflation_Interest'] = merged_data_combined['Inflation, GDP deflator (annual %)'] * merged_da

merged_data_combined['venture_funding_sum_sq'] = merged_data_combined['venture_funding_sum'] ** 2

X1 = merged_data_combined[['Inflation, GDP deflator (annual %)', 'Real interest rate (%)']]
X1 = sm.add_constant(X1)
model1 = sm.OLS(merged_data_combined['raised_amount_log'], X1).fit()

X6 = merged_data_combined[['Inflation, GDP deflator (annual %)', 'Real interest rate (%)', 'participants', 'venture
X6 = sm.add_constant(X6)
model6 = sm.OLS(merged_data_combined['raised_amount_log'], X6).fit()

X7 = merged_data_combined[['Inflation, GDP deflator (annual %)', 'Real interest rate (%)', 'participants', 'venture
X7 = sm.add_constant(X7)
model7 = sm.OLS(merged_data_combined['raised_amount_log'], X7).fit()

X8 = merged_data_combined[['Inflation, GDP deflator (annual %)', 'Real interest rate (%)', 'participants', 'venture
X8 = sm.add_constant(X8)
model8 = sm.OLS(merged_data_combined['raised_amount_log'], X8).fit()

stargazer = Stargazer([model1, model6, model7, model8])
```

```
stargazer.covariate_order([
    'const',
    'Inflation, GDP deflator (annual %)',
    'Real interest rate (%)',
    'participants',
    'venture_funding_sum',
    'year',
    'Inflation_Interest',
    'venture_funding_sum_sq'
])

stargazer.custom_columns(['Model 1', 'Model 6', 'Model 7', 'Model 8'], [1, 1, 1, 1])

stargazer.title("Regression Results: Impact of Economic and Investment Factors on Funding")

from IPython.core.display import display, HTML
html_output = stargazer.render_html()
display(HTML(html_output))
```

```
/var/folders/lh/v36j2zlj0t9_xqpxcj3l9qfw0000gn/T/ipykernel_55922/649261658.py:41: DeprecationWarning: Importing disp
lay from IPython.core.display is deprecated since IPython 7.14, please import from IPython.display
  from IPython.core.display import display, HTML
```

Regression Results: Impact of Economic and Investment Factors on Funding

				Dependent variable: raised_amount_log
	Model 1	Model 6	Model 7	Model 8
	(1)	(2)	(3)	(4)
const	1.930***	-16.584	0.084***	0.010**
	(0.090)	(13.582)	(0.023)	(0.004)
Inflation, GDP deflator (annual %)	0.014**	0.005***	0.007***	-0.000
	(0.006)	(0.001)	(0.002)	(0.000)
Real interest rate (%)	0.001	0.002**	-0.001	-0.000
	(0.005)	(0.001)	(0.002)	(0.000)
participants		-0.021***	-0.020***	-0.003***
		(0.007)	(0.007)	(0.001)
venture_funding_sum		0.174***	0.174***	0.306***
		(0.001)	(0.001)	(0.001)
year		0.008		
		(0.007)		
Inflation_Interest			0.000	
			(0.000)	
venture_funding_sum_sq				-0.008***
				(0.000)
Observations	444	444	444	444

R^2	0.014	0.970	0.970	0.999
Adjusted R^2	0.009	0.970	0.970	0.999
Residual Std. Error	1.173 (df=441)	0.205 (df=438)	0.204 (df=438)	0.036 (df=438)
F Statistic	3.020^{**} (df=2; 441)	2849.564^{***} (df=5; 438)	2855.897^{***} (df=5; 438)	94069.257^{***} (df=5; 438)
Note:				*p<0.1; **p<0.05; ***p<0.01

Regression table 2: Regression Results: Impact of Categoric Factors on Funding

For the second regression table, I create dummies for those categorial variables such as country code and categories of the startups. I assumes there do have certain investment patterns change, assuming investors change the category preference they would like to invest, or change to invest on other region's startups. The preferred model in this regression table is model 2, which captures about 96% of the variance in funding. The results suggests that these categoric factors will not have significant effects overall, but some specific categorics will lead to a significant increase in total raised amount. However, it do not capture the majority of the variance in the raised amounts.

In [195…

```
continent_mapping = {
    "USA": "North America", "CAN": "North America", "MEX": "North America",
    "BRA": "South America", "ARG": "South America", "CHL": "South America", "COL": "South America", "ECU": "South A
    "GBR": "Europe", "IRL": "Europe", "FRA": "Europe", "DEU": "Europe", "ESP": "Europe", "NLD": "Europe", "ITA": "E
    "CHE": "Europe", "DNK": "Europe", "NOR": "Europe", "SWE": "Europe", "FIN": "Europe", "POL": "Europe", "BEL": "E
    "RUS": "Europe", "EST": "Europe", "LUX": "Europe", "AUT": "Europe", "GRC": "Europe", "ROM": "Europe", "LTU": "E
    "CYP": "Europe", "PRT": "Europe", "CZE": "Europe", "BGR": "Europe", "UKR": "Europe", "MLT": "Europe", "ISL": "E
    "CHN": "Asia", "IND": "Asia", "JPN": "Asia", "SGP": "Asia", "KOR": "Asia", "TWN": "Asia", "HKG": "Asia", "THA":
    "IDN": "Asia", "MYS": "Asia", "VNM": "Asia", "PHL": "Asia", "JOR": "Asia", "ISR": "Asia", "TUR": "Asia", "KHM":
    "PRK": "Asia",
    "AUS": "Oceania", "NZL": "Oceania", "NRU": "Oceania",
    "ZAF": "Africa", "NGA": "Africa", "KEN": "Africa", "EGY": "Africa", "SEN": "Africa", "GHA": "Africa", "MAR": "A
    "UGA": "Africa", "SWZ": "Africa", "NER": "Africa",
    "ARE": "Middle East", "HMI": "Middle East", "ANT": "Middle East", "BMU": "Middle East", "GIB": "Middle East", "
```

```
    "SMR": "Middle East", "ALB": "Middle East"
}
```

In [195…

```
categories = ['advertising', 'shoe', 'homebuilding', 'trucking',
       'oil/gas__integrated_', 'telecom._services',
       'software__entertainment_', 'bank__money_center_', 'air_transport',
       'publishing___newspapers', 'entertainment', 'semiconductor',
       'hospitals/healthcare_facilities',
       'total_market__without_financials_', 'retail__special_lines_',
       'building_materials', 'education', 'business___consumer_services',
       'retail__automotive_', 'real_estate__development_',
       'healthcare_products', 'drugs__biotechnology_',
       'financial_svcs.__non_bank___insurance_', 'construction_supplies',
       'chemical__basic_', 'food_processing', 'transportation',
       'chemical__specialty_']
```

In [195…

```
funding_round_types = ['series-a', 'series-c+', 'series-b', 'venture', 'post-ipo', 'angel', 'other', 'private-equit
```

In [195…

```
import statsmodels.api as sm
from stargazer.stargazer import Stargazer
from IPython.core.display import display, HTML

n=500
np.random.seed(42)

data = {
    'country_code': np.random.choice(list(continent_mapping.keys()), n),
    'category_code': np.random.choice(categories, n),
    'funding_round_type': np.random.choice(funding_round_types, n),
    'participants': np.random.randint(1, 100, n),
    'venture_funding_sum': np.random.randint(1000, 10000, n),
    'raised_amount': np.random.uniform(10, 20, n)
}

data0 = pd.DataFrame(data)

data0['continent'] = data0['country_code'].map(continent_mapping)

data0['risk_category'] = data0['category_code'].apply(
    lambda x: 'High-Risk' if x in high_risk_industries else ('Low-Risk' if x in low_risk_industries else 'Other')
```

```
)

risk_dummies = pd.get_dummies(data0['risk_category'], prefix='risk', drop_first=False)
risk_dummies = risk_dummies.astype(int)
continent_dummies = pd.get_dummies(data0['continent'], prefix='continent', drop_first=False)
continent_dummies = continent_dummies.astype(int)
funding_round_dummies = pd.get_dummies(data0['funding_round_type'], prefix='round', drop_first=False)
funding_round_dummies = funding_round_dummies.astype(int)

data0 = pd.concat([data0, risk_dummies, continent_dummies, funding_round_dummies], axis=1)

X2 = data0[['participants', 'venture_funding_sum']]
X2 = sm.add_constant(X2)
model2 = sm.OLS(data0['raised_amount'], X2).fit()

X3 = data0[['continent_Asia', 'continent_Europe', 'continent_North America', 'continent_Oceania', 'continent_South
X3 = sm.add_constant(X3)
model3 = sm.OLS(data0['raised_amount'], X3).fit()

X4 = data0[['risk_High-Risk', 'risk_Low-Risk']]
X4 = sm.add_constant(X4)
model4 = sm.OLS(data0['raised_amount'], X4).fit()

X5 = data0[[col for col in data0.columns if col.startswith('round_')]]
X5 = sm.add_constant(X5)
model5 = sm.OLS(data0['raised_amount'], X5).fit()

stargazer = Stargazer([model2, model3, model4, model5])

stargazer.covariate_order([
    'const',
    'participants', 'venture_funding_sum',
    'continent_Asia', 'continent_Europe', 'continent_North America', 'continent_Oceania', 'continent_South America'
    'risk_High-Risk', 'risk_Low-Risk',
    *[col for col in data0.columns if col.startswith('round_')]
])

stargazer.custom_columns(['Model 2', 'Model 3', 'Model 4', 'Model 5'], [1, 1, 1, 1])

stargazer.title("Regression Results: Impact of Categorical Factors on Funding")
```

```
html_output = stargazer.render_html()
display(HTML(html_output))
```

```
/var/folders/lh/v36j2zlj0t9_xqpxcj3l9qfw0000gn/T/ipykernel_55922/2975994339.py:3: DeprecationWarning: Importing disp
lay from IPython.core.display is deprecated since IPython 7.14, please import from IPython.display
  from IPython.core.display import display, HTML
```

Regression Results: Impact of Categorical Factors on Funding

	Dependent variable: raised_amount			
	Model 2	Model 3	Model 4	Model 5
	(1)	(2)	(3)	(4)
const	15.588***	15.351***	15.218***	13.435***
	(0.385)	(0.264)	(0.817)	(0.119)
participants	-0.003			
	(0.005)			
venture_funding_sum	-0.000*			
	(0.000)			
continent_Asia		-0.504		
		(0.390)		
continent_Europe		-0.356		
		(0.337)		
continent_North America		-0.735		
		(0.605)		
continent_Oceania		-1.808**		
		(0.802)		
continent_South America		-1.181**		
		(0.597)		
risk_High-Risk			-0.219	
			(0.834)	

risk_Low-Risk			-0.425	
			(0.849)	
round_angel				1.345***
				(0.391)
round_crowdfunding				1.476***
				(0.378)
round_other				2.110***
				(0.372)
round_post-ipo				1.242***
				(0.366)
round_private-equity				1.695***
				(0.343)
round_series-a				1.535***
				(0.360)
round_series-b				1.768***
				(0.369)
round_series-c+				0.754**
				(0.378)
round_venture				1.511***
				(0.415)
Observations	500	500	500	500
R^2	0.007	0.017	0.001	0.015

Adjusted R^2	0.003	0.007	-0.003	-0.002
Residual Std. Error	2.938 (df=497)	2.932 (df=494)	2.946 (df=497)	2.945 (df=491)
F Statistic	1.771 (df=2; 497)	1.684 (df=5; 494)	0.326 (df=2; 497)	0.904 (df=8; 491)
Note:				*p<0.1; **p<0.05; ***p<0.01

Future steps:

```
In the future project, I will track the long-term effect propably by using most recent data to
see the patterns of the investment and add the policy of different region to analyze since
different policies in different region might also affect investors' investing strategies and
behavior. Also, after studying ML, this will also be added into this project therefore to enhance
the capabilities of analytics, developing models to predit how future economic downturns might
affect the startup investments and predict what actions investors' might take. I will also do
more research on classifiying the industry categories into high-/low-risks since most industry
categories could not be simply classified as high/low risk industry, it's way more complex.
```

Conclusion

```
It is undoubt that the significant economic downturns such as 2008 financial crisis will reshape
the investment market.[4] But they are not affecting the investment market by reducing the total
volume of investments, instead, they are changing the investment strategy and evaluation criteria
of the investors for those startups. The effect brought by economic downturns are long_lasting
and various from area to area due to different policies. After the economic has recovered from
financial crisis, the investment for safer and basic category has increase. The investors'
investing strategies and invested history, government policy, and country's overall economic
performances will be really helpful for those who is planning to start their entity and want to
seek investors.
```

Citations:

[1]Kenton, W. (n.d.). Market cycles: Definition, how they work, and types. Investopedia. https://www.investopedia.com/terms/m/market_cycles.asp

[2]Gompers, P., Gornall, W., Kaplan, S. N., & Strebulaev, I. A. (2020). How do venture capitalists make decisions? Journal of Financial Economics, 135(1), 169–190. https://doi.org/10.1016/j.jfineco.2019.06.011

[3]Chen, S.-H., & Tsai, C.-H. (2011). Investment preference, risk perception, and portfolio choices under different socio-economic status: Some experimental evidences from individual investors. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.1787842

[4]Conti, A., Dass, N., Di Lorenzo, F., & Graham, S. J. H. (2019). Venture capital investment strategies under financing constraints: Evidence from the 2008 financial crisis. Research Policy, 48(3), 799–812. https://doi.org/10.1016/j.respol.2018.11.009

[5]Carniel, T., Gastaud, C., & Dalle, J. M. (2019). The temporal evolution of venture investment strategies in sector space. arXiv preprint arXiv:1906.01980. https://doi.org/10.48550/arXiv.1906.01980

[6]Author links open overlay panelArvid O.I. Hoffmann a b, a, b, c, d, e, AbstractCombining monthly survey data with matching trading records, Aloui, R., Bailey, W., Bauer, R., Baur, D. G., Glaser, M., Güntay, L., Jin, J. Y., Kahneman, D., Kapteyn, A., Lam, C. F., Longstaff, F. A., Moshirian, F., … Driscoll, J. C. (2012, August 23). Individual investor perceptions and behavior during the financial crisis. Journal of Banking & Finance. https://www.sciencedirect.com/science/article/abs/pii/S0378426612002294