Project Name :- Startup Trend Analysis(2015-2019)

Introduction

- Startup Trend Analysis (2015-2019): This analysis delves into the dynamic landscape of startups spanning the years 2015 to 2019, aiming to uncover overarching trends and patterns within the startup ecosystem during this period. Through an exploration of various industries, geographical locations, and emerging technologies, this analysis seeks to provide valuable insights into the evolution of startups over the five-year timeframe, shedding light on key factors influencing their growth, success, and adaptation in the ever-changing business landscape.

Pre-processing

- The DataFrames for the years 2015 to 2019 have undergone preprocessing to ensure data integrity and consistency. These steps include handling missing values, duplicate rows,data formats, and resolving any discrepancies, preparing the datasets for comprehensive analysis of startup trends across the five-year period.

2015 Analysis

In [37]:
```
# Loading the dataset
import pandas as pd
df_2015 = pd.read_csv('C:/Users/Snehal/Downloads/2015_data.csv')
df_2015.head()
```

Out[37]:

	Sr.No	Date(dd/mm/yyyy)	Startup Name	Industry Vertical	City / Location	Investors' Name	InvestmentType	Amount (in USD)	Remarks
0	1	01 September 2015	TOFlo	FinTech Startup Incubation platform	Mumbai	Tania Johny Palathinkal	Seed Funding	1,00,000	NaN
1	2	01 September 2015	FXMartIndia	Payment Services platform	Chandigarh	Flipkart	Private Equity	NaN	Strategic Investment (Majority Stake)
2	3	01 September 2015	Stylecracker	Personalized Styling platform	Mumbai	Group of HNI investors	Private Equity	10,00,000	Series A
3	4	01 September 2015	Luxuryhues	Luxury goods Shopping Platform	Gurgaon	Reliance Capital	Private Equity	9,00,000	Series A
4	5	02 September 2015	HolaChef	Food Delivery Platform	Mumbai	Ratan Tata	Private Equity	NaN	Part of Series A raised inJune 2015

In [38]:
```
# Renaming columns for our convinience

def renaming_columns(df_2015):
    df_2015.rename(columns={
    'Date(dd/mm/yyyy)': 'Date',
    'Startup Name': 'Startup_Name',
    'City / Location': 'Location',
    'Investors' Name': 'Investors',
    'InvestmentType': 'Investment_Type',
    'Amount (in USD)': 'Amount($)',
    'Industry Vertical': 'Sub_Industry'
     }, inplace=True)
renaming_columns(df_2015)

# Extracting required columns
df_2015 = df_2015[['Date', 'Startup_Name','Sub_Industry','Location','Investors','Investment_Type','Amount($)']]

# Dealing with Date column to extract Year & Month
def date_opertion(df_2015):
    df_2015['Date'] = pd.to_datetime(df_2015['Date'], format="%d %B %Y")
    df_2015['Month'] = df_2015['Date'].dt.strftime('%B')
    df_2015['Year'] = df_2015['Date'].dt.year
date_opertion(df_2015)

# Dealing with duplicate rows
def duplicate_rows(data):
    duplicate_rows = data[data.duplicated()]
    if len(duplicate_rows) > 0:
        data = data.drop_duplicates()
        print('Droped',len(duplicate_rows),'Duplicate Rows.')
    else:
        print('No Duplicate Rows.')
duplicate_rows(df_2015)

# Dealing with Amount column data type
def amount_column(data):
    data['Amount($)'] = data['Amount($)'].fillna(0)
    data['Amount($)'] = data['Amount($)'].astype(str)
    data['Amount($)'] = data['Amount($)'].str.replace(',', '')
```

```
    data['Amount($)'] = data['Amount($)'].astype(int)
amount_column(df_2015)

# Replacing values from location column which are not location
import numpy as np
df_2015['Location'] = df_2015['Location'].astype(str)
values_to_replace = ['Seed Funding', 'SeedFunding', 'PrivateEquity', 'Crowd funding', 'Crowd Funding','Private
df_2015['Location'] = df_2015['Location'].replace(values_to_replace, np.nan, regex=True)

# Replacing odd values from Investors column
values_to_replace = [ '55,00,000', '1,20,000', '1,50,00,000', '10,00,000', '25,00,000',
       '12,00,000', '5,00,000', '1,50,000', '50,00,000', '2,00,00,000',
       '11,00,00,000', '2,00,000', '1,66,000', '13,00,000', '5,00,00,000',
       '20,00,000', '28,00,000', '30,00,000', '1,80,000', '60,00,000',
       '1,60,000', '2,50,00,000', '17,50,000', '1,00,000', '1,15,000',
       '3,00,00,000', '15,00,000', '1,00,00,000', '7,50,00,000',
       '1,60,00,000','21,50,000',
       '3,15,000', '3,80,000', '1,35,000', '2,85,000', '6,00,000',
       '1,25,000', '30,768', '1,30,00,000', '2,90,000', '6,50,000',
       '1,61,000', '10,00,00,000', '3,30,000', '16,000', '1,10,00,000',
       '1,47,50,000', '3,25,000', '32,50,000', '5,60,00,000',
       '3,10,00,000', '45,00,000', '8,25,000', '1,40,000', '16,600',
       '2,50,000', '2,60,00,000', '5,18,000', '6,00,00,000',
       '1,80,00,000', '8,00,00,000', '4,00,00,000', '1,65,00,000',
       '41,50,000', '8,00,000', '16,00,000', '3,20,000',]

df_2015['Investors'] = df_2015['Investors'].replace(values_to_replace, float('nan'))
```

```
Droped 1 Duplicate Rows.
```

In [39]:
```
df_2015.head()
```

Out[39]:

	Date	Startup_Name	Sub_Industry	Location	Investors	Investment_Type	Amount($)	Month	Year
0	2015-09-01	TOFlo	FinTech Startup Incubation platform	Mumbai	Tania Johny Palathinkal	Seed Funding	100000	September	2015
1	2015-09-01	FXMartIndia	Payment Services platform	Chandigarh	Flipkart	Private Equity	0	September	2015
2	2015-09-01	Stylecracker	Personalized Styling platform	Mumbai	Group of HNI investors	Private Equity	1000000	September	2015
3	2015-09-01	Luxuryhues	Luxury goods Shopping Platform	Gurgaon	Reliance Capital	Private Equity	900000	September	2015
4	2015-09-02	HolaChef	Food Delivery Platform	Mumbai	Ratan Tata	Private Equity	0	September	2015

Summary of the year 2015

- Shape = (938, 9)
- Unique Sub_Industry = 874
- Unique Location = 48
- Unique Investment_Type = 24

In [ ]:
```
```

2016 Analysis

In [40]:
```
# Loading the dataset
import pandas as pd
df_2016 = pd.read_excel('C:/Users/Snehal/Downloads/2016_data.xlsx')
df_2016.head()
```

Out[40]:

	Sr. No.	Date(dd-mm-yyyy)	Startup Name	Industry/ Vertical	Sub-Vertical	City / Location	Investors' Name	Invest-mentType	Amount (in USD)
0	1.0	2016-09-01	Mad Street Den	Technology	Artificial Intelligence platform	Chennai	Sequoia India, Exfinity Ventures, growX ventures,	Private Equity	NaN
1	2.0	2016-09-01	Mihup	Technology	Personal Digital Assistant	Kolkata	Accel Partners	Private Equity	6,700,000
2	3.0	2016-09-01	Renowala	eCommerce	Home Improvement Marketplace	Hyderabad	Pradeep Dhobale	Seed Funding	NaN
3	4.0	2016-09-01	Lucideus	Technology	IT Risk Assessment and Digital Security Servic...	New Delhi	Amit Choudhary	Seed Funding	NaN
4	5.0	2016-09-04	Trackbizz	Technology	Field Force Automation System	Kochi	Grasshoppers	Private Equity	NaN

In [41]:
```
# Renaming columns for our convinience

def renaming_columns(df_2016):
    df_2016.rename(columns={
```

```
    'Date(dd-mm-yyyy)': 'Date',
    'Startup Name': 'Startup_Name',
    'City / Location': 'Location',
    'Investors' Name': 'Investors',
    'Invest-mentType': 'Investment_Type',
    'Amount (in USD)': 'Amount($)',
    'Sub-Vertical': 'Sub_Industry',
    'Industry/ Vertical':'Industry'
     }, inplace=True)
renaming_columns(df_2016)

# Extracting required columns
df_2016 = df_2016[['Date', 'Startup_Name', 'Industry','Sub_Industry','Location','Investors','Investment_Type','

# Dealing with duplicate rows
def duplicate_rows(data):
    duplicate_rows = data[data.duplicated()]
    if len(duplicate_rows) > 0:
        data = data.drop_duplicates()
        print('Droped',len(duplicate_rows),'Duplicate Rows.')
    else:
        print('No Duplicate Rows.')
duplicate_rows(df_2016)


# Converting Date column to datetime to extract Year & Month
def date_opertion(df_2016):
    df_2016['Date'] = pd.to_datetime(df_2016['Date'], format="%d %B %Y")
    df_2016['Month'] = df_2016['Date'].dt.strftime('%B')
    df_2016['Year'] = df_2016['Date'].dt.year
date_opertion(df_2016)

# Dealing with Amount column data type
def amount_column(data):
    data['Amount($)'] = data['Amount($)'].fillna(0)
    data['Amount($)'] = data['Amount($)'].astype(str)
    data['Amount($)'] = data['Amount($)'].str.replace(',', '')
    data['Amount($)'] = data['Amount($)'].astype(int)
amount_column(df_2016)

# Editing Industry column
values_to_replace = {'eCommerce' : 'E-Commerce',
                     'ECommerce' : 'E-Commerce',
                     'Ecommerce' : 'E-Commerce',
                     'ecommerce' : 'E-Commerce',
                     'healthcare' : 'Healthcare',
                     'Consumer Interne' : 'Consumer Internet'}
def replace_values(df):
    df['Industry'] = df['Industry'].replace(values_to_replace)
replace_values(df_2016)

# Editing Investor column
values_to_replace = {'Undisclosed investor' : 'Undisclosed Investors',
                     'Undisclosed investors' : 'Undisclosed Investors',
                      'Undisclosed Investor' : 'Undisclosed Investors',
                      'undisclosed investors' : 'Undisclosed Investors'
                     }
def replace_values(df):
    df['Investors'] = df['Investors'].replace(values_to_replace)
replace_values(df_2016)

# Editing Location column

def replace_values(df):
    value_to_replace = {'US': 'United States', 'USA' : 'United States'}
    df['Location'] = df['Location'].replace(value_to_replace)
replace_values(df_2016)
```

```
Droped 17 Duplicate Rows.
```

In [42]:
```
df_2016.head()
```

Out[42]:

	Date	Startup_Name	Industry	Sub_Industry	Location	Investors	Investment_Type	Amount($)	Month	Year
0	2016-09-01	Mad Street Den	Technology	Artificial Intelligence platform	Chennai	Sequoia India, Exfinity Ventures, growX ventures,	Private Equity	0	September	2016.0
1	2016-09-01	Mihup	Technology	Personal Digital Assistant	Kolkata	Accel Partners	Private Equity	6700000	September	2016.0
2	2016-09-01	Renowala	E-Commerce	Home Improvement Marketplace	Hyderabad	Pradeep Dhobale	Seed Funding	0	September	2016.0
3	2016-09-01	Lucideus	Technology	IT Risk Assessment and Digital Security Servic...	New Delhi	Amit Choudhary	Seed Funding	0	September	2016.0
4	2016-09-04	Trackbizz	Technology	Field Force Automation System	Kochi	Grasshoppers	Private Equity	0	September	2016.0

Summary of the year 2016

- Shape = (1041, 10)
- Unique Industry = 14
- Unique Sub_Industry = 987
- Unique Location = 41
- Unique Investment_Type = 3

In [ ]:

2017 Analysis

In [43]:
```
#Loading the dataset
import pandas as pd
df_2017 = pd.read_excel('C:/Users/Snehal/Downloads/2017_data.xlsx')
df_2017.head()
```

Out[43]:

	Sr. No.	Date	Startup Name	Industry/Vertical	Sub-Vertical	City	Investor Name	Investment Type	Amount(in USD)	InvestmentType
0	1.0	2017-09-01	Aahaa Stores	eCommece	Online B2B store for office supplies	Chennai	YourNest Angel Fund	Private Equity	1,000,000	NaN
1	2.0	2017-09-01	MFine	Consumer Internet	Online Doctor Discovery platform	Bangalore	Stellaris Venture Partners, Mayur Abhaya, Rohi...	Private Equity	1,500,000	NaN
2	3.0	2017-09-01	Canvera	Consumer Internet	Online Photography platform	Mumbai	InfoEdge	Private Equity	1,300,000	NaN
3	4.0	2017-09-04	PrimaryIO	Technology	Application Performance Acceleration	Pune	Accel Partners, Exfinity Ventures, Partech Ven...	Private Equity	5,600,000	NaN
4	5.0	2017-09-05	Shubh Loans	Consumer Internet	online lending platform	Bangalore	SRI Capital, BeeNext, Pravega Ventures	Private Equity	1,500,000	NaN

In [44]:
```
# Renaming columns for our convinience

def renaming_columns(df_2017):
    df_2017.rename(columns={
    'Startup Name': 'Startup_Name',
    'City': 'Location',
    'Investor Name': 'Investors',
    'InvestmentType': 'Investment_Type',
    'Amount(in USD)': 'Amount($)',
    'Sub-Vertical': 'Sub_Industry',
    'Industry/Vertical':'Industry'
     }, inplace=True)
renaming_columns(df_2017)

# Extracting required columns
df_2017 = df_2017[['Date','Startup_Name','Industry','Sub_Industry','Location','Investors','Investment_Type','Am

# Converting Date column to datetime to extract Year & Month

def date_opertion(df_2017):
    df_2017['Date'] = pd.to_datetime(df_2017['Date'], format="%d %B %Y")
    df_2017['Month'] = df_2017['Date'].dt.strftime('%B')
    df_2017['Year'] = df_2017['Date'].dt.year
date_opertion(df_2017)

# Dealing with duplicate rows
def duplicate_rows(data):
    duplicate_rows = data[data.duplicated()]
    if len(duplicate_rows) > 0:
        data = data.drop_duplicates()
        print('Droped',len(duplicate_rows),'Duplicate Rows.')
    else:
        print('No Duplicate Rows.')
duplicate_rows(df_2017)

# Dealing with Amount column data type
def amount_column(data):
    data['Amount($)'] = data['Amount($)'].fillna(0)
    data['Amount($)'] = data['Amount($)'].astype(str)
    data['Amount($)'] = data['Amount($)'].str.replace(',', '')
    data['Amount($)'] = data['Amount($)'].astype(float)
amount_column(df_2017)

# Editing Industry column
values_to_replace = {'eCommece' : 'E-Commerce',
                     'eCommerce' : 'E-Commerce',
                     'ECommerce' : 'E-Commerce',
                     'Ecommerce' : 'E-Commerce',
                     'Health Care' : 'Healthcare',}
```

```
def replace_values(df):
    df['Industry'] = df['Industry'].replace(values_to_replace)
replace_values(df_2017)

# Editing Location column

def replace_values(df):
    value_to_replace = {'Bengaluru': 'Bangalore', 'Nw Delhi' : 'New Delhi', 'Delhi':'New Delhi'}
    df['Location'] = df['Location'].replace(value_to_replace)
replace_values(df_2017)

# Editing Investor column
values_to_replace = {'Undisclosed investor' : 'Undisclosed Investors',
                     'Undisclosed investors' : 'Undisclosed Investors',
                      'Undisclosed Investor' : 'Undisclosed Investors',
                      'undisclosed investors' : 'Undisclosed Investors'
                     }
def replace_values(df):
    df['Investors'] = df['Investors'].replace(values_to_replace)
replace_values(df_2017)
```

```
Droped 9 Duplicate Rows.
```

In [45]:
```
df_2017.head()
```

Out[45]:

	Date	Startup_Name	Industry	Sub_Industry	Location	Investors	Investment_Type	Amount($)	Month	Year
0	2017-09-01	Aahaa Stores	E-Commerce	Online B2B store for office supplies	Chennai	YourNest Angel Fund	NaN	1000000.0	September	2017.0
1	2017-09-01	MFine	Consumer Internet	Online Doctor Discovery platform	Bangalore	Stellaris Venture Partners, Mayur Abhaya, Rohi...	NaN	1500000.0	September	2017.0
2	2017-09-01	Canvera	Consumer Internet	Online Photography platform	Mumbai	InfoEdge	NaN	1300000.0	September	2017.0
3	2017-09-04	PrimaryIO	Technology	Application Performance Acceleration	Pune	Accel Partners, Exfinity Ventures, Partech Ven...	NaN	5600000.0	September	2017.0
4	2017-09-05	Shubh Loans	Consumer Internet	online lending platform	Bangalore	SRI Capital, BeeNext, Pravega Ventures	NaN	1500000.0	September	2017.0

Summary of the year 2017

- Shape = (700, 10)
- Unique Industry = 12
- Unique Sub_Industry = 656
- Unique Location = 17
- Unique Investment_Type = 3

In [ ]:
```

```

2018 Analysis

In [46]:
```
# Loading the dataset
import pandas as pd
df_2018 = pd.read_excel('C:/Users/Snehal/Downloads/2018_data.xlsx')
df_2018.head()
```

Out[46]:

	Sr. No.	Date	Startup Name	Industry/Vertical	Sub-Vertical	City	Investor Name	Investment Type	Amount(in USD)
0	1.0	2018-09-01 00:00:00	Netmeds	Consumer Internet	Online Pharmacy Chain	Chennai	Sistema Asia Fund, Sistema JSFC and Tanncam In...	Private Equity	35,000,000
1	2.0	2018-09-03 00:00:00	Udaan	B2B Platform	Logistics and Shipping	Bengaluru	DST Global and Lightspeed Venture Partners' gl...	Private Equity	225,000,000
2	3.0	2018-09-03 00:00:00	Daily hunt	Consumer Internet	News and ebooks Mobile App	Bengaluru	Falcon Edge	Private Equity	63,90,000
3	4.0	2018-09-04 00:00:00	3HCare	Healthcare	Healthcare Service Provider	Delhi	NaN	Seed / Angel Funding	1,000,000
4	5.0	2018-09-04 00:00:00	HappyGoEasy	Consumer Internet	Online Travel Agecy	Gurugram	Korea Investment Partners (KIP), Samsung and C...	Private Equity	NaN

In [47]:
```
# Renaming columns for our convinience

def renaming_columns(df_2018):
    df_2018.rename(columns={
    'Startup Name': 'Startup_Name',
    'City': 'Location',
    'Investor Name': 'Investors',
```

```
    'Investment Type': 'Investment_Type',
    'Amount(in USD)': 'Amount($)',
    'Sub-Vertical': 'Sub_Industry',
    'Industry/Vertical':'Industry'
     }, inplace=True)
renaming_columns(df_2018)


# Extracting required columns
df_2018 = df_2018[['Date','Startup_Name','Industry','Sub_Industry','Location','Investors','Investment_Type','Am

# Converting Date column to datetime to extract Year & Month
def date_opertion(df_2018):
    df_2018['Date'] = pd.to_datetime(df_2018['Date'], format="%d-%m%Y")
    df_2018['Month'] = df_2018['Date'].dt.strftime('%B')
    df_2018['Year'] = df_2018['Date'].dt.year
date_opertion(df_2018)


# Dealing with duplicate rows
def duplicate_rows(data):
    duplicate_rows = data[data.duplicated()]
    if len(duplicate_rows) > 0:
        data = data.drop_duplicates()
        print('Droped',len(duplicate_rows),'Duplicate Rows.')
    else:
        print('No Duplicate Rows.')
duplicate_rows(df_2018)


# Dealing with Amount column data type
def amount_column(data):
    data['Amount($)'] = data['Amount($)'].fillna(0)
    data['Amount($)'] = data['Amount($)'].astype(str)
    data['Amount($)'] = data['Amount($)'].str.replace(',', '')
    data['Amount($)'] = pd.to_numeric(data['Amount($)'], errors='coerce')
    data['Amount($)'] = data['Amount($)'].fillna(0).astype(int)
amount_column(df_2018)


# Editing Location column

def replace_values(df):
    value_to_replace = {'Ahmedabad': 'Ahemadabad', 'Ahemdabad' : 'Ahemadabad', 'Bengaluru' : 'Bangalore','Kolka
                        'Bhubneswar' : 'Bhubaneswar'}
    df['Location'] = df['Location'].replace(value_to_replace)
replace_values(df_2018)


# Editing Industry column
values_to_replace = {'Ecommerce' : 'E-Commerce',
                     'E-commerce' : 'E-Commerce',
                     'E-Commerce' : 'E-Commerce',
                     'Ecommerce' : 'E-Commerce',
                     'B2B Platform' : 'B2B',
                     'Consumer internet' : 'Consumer Internet',
                     'Ed-tech' : 'Ed-Tech',
                     'Fiinance' : 'Finance',
                     'Food Tech' : 'Food-Tech','Food and Beverages' : 'Food & Beverages','Food and Beverage':'Fo
                     'Services':'Services Platform','Finance':'Financial Tech'}
def replace_values(df):
    df['Industry'] = df['Industry'].replace(values_to_replace)
replace_values(df_2018)
```

```
No Duplicate Rows.
```

In [48]:
```
df_2018.head()
```

Out[48]:

	Date	Startup_Name	Industry	Sub_Industry	Location	Investors	Investment_Type	Amount($)	Month	Year
0	2018-09-01	Netmeds	Consumer Internet	Online Pharmacy Chain	Chennai	Sistema Asia Fund, Sistema JSFC and Tanncam In...	Private Equity	35000000	September	2018.0
1	2018-09-03	Udaan	B2B	Logistics and Shipping	Bangalore	DST Global and Lightspeed Venture Partners' gl...	Private Equity	225000000	September	2018.0
2	2018-09-03	Daily hunt	Consumer Internet	News and ebooks Mobile App	Bangalore	Falcon Edge	Private Equity	6390000	September	2018.0
3	2018-09-04	3HCare	Healthcare	Healthcare Service Provider	Delhi	NaN	Seed / Angel Funding	1000000	September	2018.0
4	2018-09-04	HappyGoEasy	Consumer Internet	Online Travel Agecy	Gurugram	Korea Investment Partners (KIP), Samsung and C...	Private Equity	0	September	2018.0

Summary of the year 2018

- Shape = (309, 10)
- Unique Industry = 38
- Unique Sub_Industry = 268
- Unique Location = 29
- Unique Investment_Type = 23

In [ ]:

2019 Analysis

In [49]:
```
# Loading the dataset
df_2019 = pd.read_excel('C:/Users/Snehal/Downloads/2019_data.xlsx')
df_2019.head()
```

Out[49]:

	Sr. No.	Date	Startup Name	Industry/Vertical	Sub-Vertical	City	Investor Name	Investment Type	Amount(in USD)
0	1.0	2019-09-05	FPL Technologies	FinTech	Financial Services	Pune	Matrix Partners India, Sequoia India	Maiden Round	4,500,000
1	2.0	2019-09-04	Cashflo	FinTech	Invoice discounting platform and SME lending m...	Mumbai	SAIF Partners	Series A	3,300,000
2	3.0	2019-09-04	Digital F5	Advertising, Marketing	Digital marketing firm	Mumbai	TIW Private Equity	Private Equity Round	6,000,000
3	4.0	2019-09-04	3rdFlix	SaaS	Education Technology	Hyderabad	Exfinity Venture Partners	pre-series A	5,000,000
4	5.0	2019-09-04	75F	IoT	Building automation system	Burnsville	Breakthrough Energy Ventures	Series A	18,000,000

In [50]:
```
# Renaming columns for our convinience

def renaming_columns(df_2019):
    df_2019.rename(columns={
    'Startup Name': 'Startup_Name',
    'City': 'Location',
    'Investor Name': 'Investors',
    'Investment Type': 'Investment_Type',
    'Amount(in USD)': 'Amount($)',
    'Sub-Vertical': 'Sub_Industry',
    'Industry/Vertical':'Industry'
     }, inplace=True)
renaming_columns(df_2019)

# Extracting required columns
df_2019 = df_2019[['Date','Startup_Name','Industry','Sub_Industry','Location','Investors','Investment_Type','Am

# Converting date column to datetime to extract Year & Month
def date_opertion(df_2019):
    df_2019['Date'] = pd.to_datetime(df_2019['Date'], format="%d-%m%Y")
    df_2019['Month'] = df_2019['Date'].dt.strftime('%B')
    df_2019['Year'] = df_2019['Date'].dt.year
date_opertion(df_2019)

# Dealing with duplicate rows
def duplicate_rows(data):
    duplicate_rows = data[data.duplicated()]
    if len(duplicate_rows) > 0:
        data = data.drop_duplicates()
        print('Droped',len(duplicate_rows),'Duplicate Rows.')
    else:
        print('No Duplicate Rows.')
duplicate_rows(df_2019)

# Dealing with Amount column data type
def amount_column(data):
    data['Amount($)'] = data['Amount($)'].fillna(0)
    data['Amount($)'] = data['Amount($)'].astype(str)
    data['Amount($)'] = data['Amount($)'].str.replace(',', '')
    data['Amount($)'] = pd.to_numeric(data['Amount($)'], errors='coerce')
    data['Amount($)'] = data['Amount($)'].fillna(0).astype(int)
amount_column(df_2019)

# Editing Industry column
values_to_replace = {'AI' : 'Artificial Intelligence',
                     'Customer Service' : 'Customer Service Platform',
                     'Ecommerce' : 'E-Commerce',
                     'E-commerce' : 'E-Commerce',
                     'EdTech' : 'Ed-Tech',
                     'Education' : 'Ed-Tech','Edtech': 'Ed-Tech','FinTech' : 'Fin-Tech', 'Fintech':'Fin-Tech',
                     'Health Care' : 'Healthcare','Health and wellness' : 'Healthcare','Health and Wellness':'He
                     'Saas':'SaaS','Tech':'Technology','Transport':'Transportation',}
def replace_values(df):
    df['Industry'] = df['Industry'].replace(values_to_replace)
```

```
replace_values(df_2019)
```

```
No Duplicate Rows.
```

In [51]:
```
df_2019.head()
```

Out[51]:

	Date	Startup_Name	Industry	Sub_Industry	Location	Investors	Investment_Type	Amount($)	Month	Year
0	2019-09-05	FPL Technologies	Fin-Tech	Financial Services	Pune	Matrix Partners India, Sequoia India	Maiden Round	4500000	September	2019.0
1	2019-09-04	Cashflo	Fin-Tech	Invoice discounting platform and SME lending m...	Mumbai	SAIF Partners	Series A	3300000	September	2019.0
2	2019-09-04	Digital F5	Advertising, Marketing	Digital marketing firm	Mumbai	TIW Private Equity	Private Equity Round	6000000	September	2019.0
3	2019-09-04	3rdFlix	SaaS	Education Technology	Hyderabad	Exfinity Venture Partners	pre-series A	5000000	September	2019.0
4	2019-09-04	75F	IoT	Building automation system	Burnsville	Breakthrough Energy Ventures	Series A	18000000	September	2019.0

Summary of the year 2019

- Shape = (114, 10)
- Unique Industry = 41
- Unique Sub_Industry = 103
- Unique Location = 34
- Unique Investment_Type = 36

In [ ]:
```
```

- The "final" dataframe is a concatenated version of individual preprocessed dataframes from the years 2015 to 2019. This combined dataset serves as a comprehensive repository of startup information across these five years, facilitating a holistic analysis of startup trends, investment patterns, and industry dynamics over the specified period.

In [52]:
```
# Concating all dataframes after preprocessing individually
final = pd.concat([df_2015, df_2016,df_2017,df_2018,df_2019], ignore_index=True)
final.head()
```

Out[52]:

	Date	Startup_Name	Sub_Industry	Location	Investors	Investment_Type	Amount($)	Month	Year	Industry
0	2015-09-01	TOFlo	FinTech Startup Incubation platform	Mumbai	Tania Johny Palathinkal	Seed Funding	100000.0	September	2015.0	NaN
1	2015-09-01	FXMartIndia	Payment Services platform	Chandigarh	Flipkart	Private Equity	0.0	September	2015.0	NaN
2	2015-09-01	Stylecracker	Personalized Styling platform	Mumbai	Group of HNI investors	Private Equity	1000000.0	September	2015.0	NaN
3	2015-09-01	Luxuryhues	Luxury goods Shopping Platform	Gurgaon	Reliance Capital	Private Equity	900000.0	September	2015.0	NaN
4	2015-09-02	HolaChef	Food Delivery Platform	Mumbai	Ratan Tata	Private Equity	0.0	September	2015.0	NaN

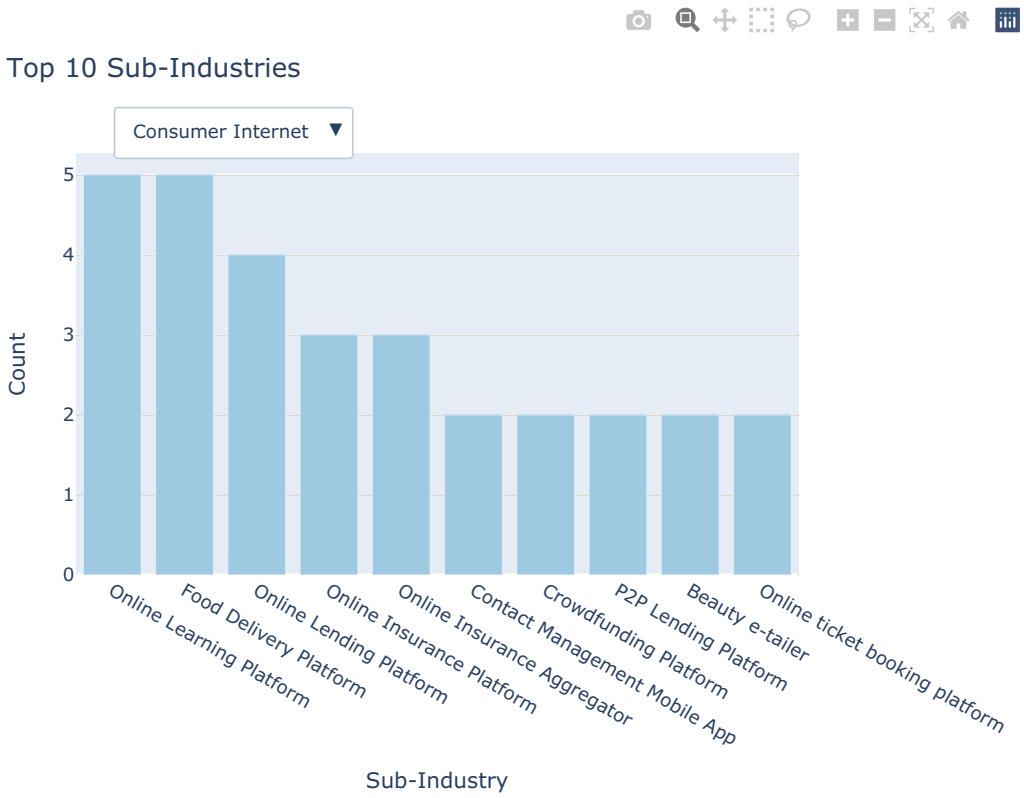
In [54]:
```
# Editing Industry column
values_to_replace = {'Bengaluru' : 'Bangalore'}

def replace_values(df):
    df['Location'] = df['Location'].replace(values_to_replace)
replace_values(final)

# Editing Industry column
values_to_replace = {'Undisclosed' : np.nan,
                     'Undisclosed Investors' : np.nan}

def replace_values(df):
    df['Investors'] = df['Investors'].replace(values_to_replace)
replace_values(final)

# Editing Investment_Type column
value_to_replace = {'Seed / Angel Funding':'Seed/Angel Funding','Angel / Seed Funding':'Seed/Angel Funding',
                    'Seed / Angle Funding':'Seed/Angel Funding','Seed/ Angel Funding':'Seed/Angel Funding',
                    'SeedFunding':'Seed Funding','Seed funding':'Seed Funding','Seed':'Seed Funding','Seed Round
                    'Seed Funding Round':'Seed Funding','More details':np.nan,'Valuation at $4M':np.nan,'3rd Rou
                    'More Details':np.nan,'To fund edu startups':np.nan,'At the 10 minute million event':np.nan}
def replace_value(df):
    df['Investment_Type'] = df['Investment_Type'].replace(value_to_replace)
replace_value(final)
```

Graphs

1) Top 10 Sub Industries in each Top 10 Industries

- The graph illustrates the top 10 sub-industries within each of the top 10 industries, offering insights into the diversified sectors driving startup activity. Each colored bar represents a sub-industry, showcasing the distribution of startups across various niche sectors within the broader industry landscape. Through this visualization, we gain a nuanced understanding of the specific areas of focus and innovation driving entrepreneurial endeavors across different industries.

In [56]:
```
import plotly.graph_objects as go

industry_counts = final['Industry'].value_counts().head(10)
top_sub_industries = {}
for industry in industry_counts.index:
    top_sub_industries[industry] = final[final['Industry'] == industry]['Sub_Industry'].value_counts().head(10)

fig = go.Figure()
colors = ['rgb(158,202,225)', 'rgb(206,162,225)', 'rgb(227,184,118)', 'rgb(168,227,118)', 'rgb(118,227,192)']

for i, industry in enumerate(industry_counts.index):
    fig.add_trace(go.Bar(
        x=top_sub_industries[industry].index,
        y=top_sub_industries[industry].values,
        name=industry,
        visible=(i == 0),   marker_color=colors[i % len(colors)],
    ))

buttons = []
for i, industry in enumerate(industry_counts.index):
    button = dict(
        label=industry,
        method="update",
        args=[{"visible": [i == j for j in range(len(industry_counts))]}, {"title": f"Top 10 Sub-Industries in
    )
    buttons.append(button)

fig.update_layout(
    updatemenus=[
        dict(
            buttons=buttons,
            direction="down",
            pad={"r": 10, "t": 10},
            showactive=True,
            x=0.05,
            xanchor="left",
            y=1.15,
            yanchor="top"
        ),
    ],
    title="Top 10 Sub-Industries",
    xaxis=dict(title="Sub-Industry"),
    yaxis=dict(title="Count"),
)

fig.show()
```



2) Top Industries & Sub -Industries startup count wise

- The visualization presents the top industries alongside their respective top sub-industries, providing a comprehensive overview of the dominant sectors and their corresponding niche areas within the startup ecosystem. Through this table, one can quickly discern the primary focus areas within each industry, shedding light on the varied landscape of entrepreneurial endeavors across different sectors.

In [57]:
```
import pandas as pd
import plotly.express as px

top_industries = final['Industry'].value_counts().head(10).index.tolist()
top_sub_industries = {}
for industry in top_industries:
    sub_industry_counts = final[final['Industry'] == industry]['Sub_Industry'].value_counts()
    top_sub_industries[industry] = {'Sub_Industry': sub_industry_counts.index[0], 'Sub_Industry_Count': sub_ind

table_data = []
for industry in top_industries:
    table_data.append([industry, top_sub_industries[industry]['Industry_Count'], top_sub_industries[industry]['

table_df = pd.DataFrame(table_data, columns=['Top Industry', 'Industry Count', 'Top Sub-Industry', 'Sub-Industr
fig = go.Figure(data=[go.Table(
    header=dict(values=list(table_df.columns),
                fill_color='lightgreen',
                align='center'),
    cells=dict(values=[table_df['Top Industry'], table_df['Industry Count'], table_df['Top Sub-Industry'], tabl
               fill_color='lavender',
               align='center'))
])

fig.update_layout(
    title='Top Industries and Top Sub-Industries',
    font=dict(size=12, family='Arial', color='black'),
    margin=dict(l=20, r=20, t=40, b=20),
)

fig.show()
```

3) Top 10 Locations by startup count

- The visualization showcases the top 10 locations with the highest concentration of startups, offering insight into the geographical distribution of entrepreneurial activity. Through a bar graph, it presents a clear representation of the startup counts in each location, providing valuable information for understanding regional trends and hotspots within the startup ecosystem.

In [58]:
```
import plotly.graph_objects as go

def top_locations(data):
    location_counts = data['Location'].value_counts().reset_index()
    location_counts.columns = ['Location', 'Startup_Count']
    top_10_locations = location_counts.head(10)

    fig = go.Figure()
    fig.add_trace(go.Bar(
        x=top_10_locations['Location'],
        y=top_10_locations['Startup_Count'],
        marker_color='lightseagreen'
    ))

    fig.update_layout(
        title='Top 10 Locations by Startup Count',
        xaxis_title='Location',
        yaxis_title='Startup Count',
        showlegend=False
    )

    fig.show()

top_locations(final)
```

4) Top 10 Investers

- The visualization highlights the top 10 investors based on their involvement with startups, presenting their respective counts of startup engagements in a horizontal bar graph. Each investor is represented by a bar, with colors adding visual distinction, allowing for easy comparison of their influence within the startup ecosystem. Through this representation, significant investors and their impact on startup ventures are brought to the forefront for analysis and insight.

In [59]:
```
import plotly.graph_objects as go

def top_investors(data):
    investor_counts = data['Investors'].value_counts().reset_index().head(10)
    investor_counts.columns = ['Investors', 'Startup_Count']
    investor_counts = investor_counts[::-1]
    light_colors = ['lightblue', 'lightgreen', 'lightcoral', 'lightsalmon', 'lightpink',
                    'lightseagreen', 'lightsteelblue', 'grey', 'salmon', 'skyblue']

    fig = go.Figure(data=[go.Bar(
        y=investor_counts['Investors'],
        x=investor_counts['Startup_Count'],
        orientation='h',
        marker=dict(
            color=light_colors,
        )
    )])

    fig.update_layout(
        title='Top 10 Investors by Startup Count',
        xaxis_title='Startup Count',
        yaxis_title='Investors',
        showlegend=False
    )

    fig.show()

top_investors(final)
```

5) Total Investment & Startup count year wise

- The combined visualization depicts trends in the startup ecosystem over time, showcasing both total investment per year and the number of startups established each year. The bar charts provide insights into the growth and financial activity of startups over successive years, offering a comprehensive view of the evolving landscape of entrepreneurship. The distinctive colors aid in distinguishing between the two metrics, facilitating easy comparison and analysis of their respective trends.

In [60]:
```
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import pandas as pd

def amount_yearwise(df):
    df = df.dropna(subset=['Year'])
    df = df[~df['Year'].isin([float('inf'), float('-inf')])]
    df['Year'] = df['Year'].astype(int)
    investment_sum = df.groupby('Year')['Amount($)'].sum()

    fig = go.Figure(data=[go.Bar(
        y=investment_sum.index,
        x=investment_sum.values,
        orientation='h',
        marker=dict(color='salmon')
    )])

    fig.update_layout(
        title='Total Investment per Year',
        yaxis_title='Year',
        xaxis_title='Total Investment'
    )

    return fig

def startup_count_year(df):
    df = df.dropna(subset=['Year'])
    df = df[~df['Year'].isin([float('inf'), float('-inf')])]
    df['Year'] = df['Year'].astype(int)
    startup_count = df.groupby('Year').size()
    fig = go.Figure(data=[go.Bar(
        y=startup_count.index,
        x=startup_count.values,
        orientation='h',
        marker=dict(color='skyblue')
    )])

    fig.update_layout(
        title='Number of Startups Each Year',
        yaxis_title='Year',
        xaxis_title='Number of Startups'
    )

    return fig
```

```
fig = make_subplots(rows=1, cols=2, subplot_titles=('Total Investment per Year', 'Number of Startups Each Year'
fig.add_trace(amount_yearwise(final)['data'][0], row=1, col=1)
fig.add_trace(startup_count_year(final)['data'][0], row=1, col=2)
investment_colors = ['lightblue', 'lightgreen', 'lightcoral', 'lightsalmon', 'lightpink',
                     'lightseagreen', 'lightsteelblue', 'lightgrey', 'lightcyan', 'lightgoldenrodyellow']
fig.update_traces(marker=dict(color=investment_colors), row=1, col=1)
startup_colors = ['lightblue', 'lightgreen', 'lightcoral', 'lightsalmon', 'lightpink',
                  'lightseagreen', 'lightsteelblue', 'lightgrey', 'lightcyan', 'lightgoldenrodyellow']
fig.update_traces(marker=dict(color=startup_colors), row=1, col=2)
fig.update_layout(height=600, width=1000, title_text="Total Investment and Startup Count Each Year")
fig.show()
```

6) Count of Startups Investment Type wise

- The visualization presents the distribution of startups according to their investment types, focusing on the top 10 investment categories within the dataset. Through a horizontal bar chart, it provides a clear overview of the prevalence of various investment types in funding startup ventures. This insight aids in understanding the preferred investment strategies and trends within the entrepreneurial landscape.

In [61]:
```
import plotly.graph_objects as go
import pandas as pd

def investment_type_count(df):
    investment_count = df['Investment_Type'].value_counts().reset_index()
    investment_count.columns = ['Investment_Type', 'Startup_Count']
    investment_count = investment_count.sort_values(by='Startup_Count', ascending=False).head(10)
    investment_count = investment_count[::-1]
    fig = go.Figure(data=[go.Bar(
        x=investment_count['Startup_Count'],
        y=investment_count['Investment_Type'],
        orientation='h',
        marker=dict(color='lightseagreen')
    )])

    fig.update_layout(
        title='Count of Startups Investment Type wise(Top 10)',
        xaxis_title='Startup Count',
        yaxis_title='Investment Type',
        width=800,
        height=600
    )

    fig.show()

investment_type_count(final)
```

In [ ]:

Insights -

- 1) Top 10 Industries and Sub-Industries:
  - The analysis identifies the dominant industries driving startup activity, with technology, healthcare, and finance emerging as prominent sectors.
  - Within each industry, specific sub-industries such as e-online learning platforms, online pharmacy,etc online lending platforms stand out as key areas of focus, highlighting the diversified nature of entrepreneurial endeavors.
- 2) Geographical Distribution:
  - Examination of startup locations unveils hotspots of entrepreneurial activity, with cities like Bangalore, New Delhi, and Pune.
  - Regional variations in startup concentration shed light on the global landscape of innovation and entrepreneurship, showcasing clusters of startup ecosystems across different continents.
- 3) Top Investors by Startup Count:
  - Examination of investor engagement reveals notable players with significant involvement in startup ventures.
  - Investors such as Ratan Tata, Indian Angel Network, and Accel Partners emerge as top contributors to the startup ecosystem, based on the frequency of their engagements with startups.
- 4) Top Investment Types by Startup Count:
  - Analysis of investment types showcases the preferred strategies employed by investors in funding startup ventures.
  - Common investment types include seed funding, private equity, and angel investments, reflecting diverse approaches to financing startups and driving innovation.
- 5) Year-Wise Startup Trends and Investment Activity:
  - Examination of startup trends reveals fluctuations in the number of startups established each year, reflecting shifts in entrepreneurial activity and innovation.
  - Concurrently, investment trends exhibit corresponding patterns, with fluctuations in total investment amounts mirroring changes in startup establishment rates across different years.
  - Analysis of startup and investment trends in tandem offers insights into the correlation between entrepreneurial activity and investment levels, highlighting the dynamic interplay between startup formation and funding availability over time.

In [ ]: