python-analysis-5

September 5, 2025

[1]:
```
from IPython.display import IFrame
import datetime
print("Notebook was last executed on:", datetime.date.today().
↪strftime("%Y-%b-%d"), "with Python version")
!python --version
```

Notebook was last executed on: 2025-sep-05 with Python version

```
Python 3.13.0
```

[2]:
```
IFrame('https://cdn.finshots.app/images/2023/07/design-119-shark-tank.jpg',␣
↪width=1000, height=700)
```

[2]: <IPython.lib.display.IFrame at 0x107fcfb60>

1 Project Objective

This project investigates patterns in startup funding pitches from Shark Tank India across multiple seasons. Using comprehensive pitch-level data, it uncovers the factors that influence funding outcomes, analyzes industry trends, and explores investor behavior.

2 Table of Contents

3 Importing Libraries and Setup

[3]:
```
!pip install geopandas
```

```
Requirement already satisfied: geopandas in
/Users/satarupabanik/myenv/lib/python3.13/site-packages (1.0.1)
Requirement already satisfied: numpy>=1.22 in
/Users/satarupabanik/myenv/lib/python3.13/site-packages (from geopandas) (2.1.3)
Requirement already satisfied: pyogrio>=0.7.2 in
/Users/satarupabanik/myenv/lib/python3.13/site-packages (from geopandas)
(0.10.0)
Requirement already satisfied: packaging in
/Users/satarupabanik/myenv/lib/python3.13/site-packages (from geopandas) (24.2)
Requirement already satisfied: pandas>=1.4.0 in
/Users/satarupabanik/myenv/lib/python3.13/site-packages (from geopandas) (2.2.3)
Requirement already satisfied: pyproj>=3.3.0 in
/Users/satarupabanik/myenv/lib/python3.13/site-packages (from geopandas) (3.7.1)
Requirement already satisfied: shapely>=2.0.0 in
/Users/satarupabanik/myenv/lib/python3.13/site-packages (from geopandas) (2.1.0)
Requirement already satisfied: python-dateutil>=2.8.2 in
/Users/satarupabanik/myenv/lib/python3.13/site-packages (from
pandas>=1.4.0->geopandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/Users/satarupabanik/myenv/lib/python3.13/site-packages (from
pandas>=1.4.0->geopandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in
/Users/satarupabanik/myenv/lib/python3.13/site-packages (from
pandas>=1.4.0->geopandas) (2024.2)
Requirement already satisfied: certifi in
/Users/satarupabanik/myenv/lib/python3.13/site-packages (from
pyogrio>=0.7.2->geopandas) (2024.8.30)
Requirement already satisfied: six>=1.5 in
/Users/satarupabanik/myenv/lib/python3.13/site-packages (from python-
dateutil>=2.8.2->pandas>=1.4.0->geopandas) (1.16.0)

[notice] A new release of pip is
available: 24.3.1 -> 25.1.1
[notice] To update, run:
pip install --upgrade pip
```

[4]:
```
import numpy as np
import pandas as pd
pd.set_option('display.max_columns', None)

import matplotlib.pyplot as plt
import seaborn as sns
from babel.numbers import format_currency
```

```python
from wordcloud import WordCloud, STOPWORDS
import geopandas as gpd
import plotly.express as px
import plotly.io as pio
```

4 Importing Dataset

[5]:
```python
df = pd.read_csv("Shark Tank India.csv")
nRow, nCol = df.shape
print(f'\nThere are {nRow} rows and {nCol} columns in the dataset')
```

```
There are 634 rows and 80 columns in the dataset
```

4.1 Data Cleaning

[6]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 634 entries, 0 to 633
Data columns (total 80 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Season Number               634 non-null    int64
 1   Startup Name                634 non-null    object
 2   Episode Number              634 non-null    int64
 3   Pitch Number                634 non-null    int64
 4   Season Start                634 non-null    object
 5   Season End                  478 non-null    object
 6   Original Air Date           603 non-null    object
 7   Episode Title               634 non-null    object
 8   Anchor                      634 non-null    object
 9   Industry                    634 non-null    object
 10  Business Description        634 non-null    object
 11  Company Website             618 non-null    object
 12  Started in                  456 non-null    float64
 13  Number of Presenters        634 non-null    int64
 14  Male Presenters             568 non-null    float64
 15  Female Presenters           382 non-null    float64
 16  Transgender Presenters      3 non-null      float64
 17  Couple Presenters           634 non-null    int64
 18  Pitchers Average Age        634 non-null    object
 19  Pitchers City               626 non-null    object
 20  Pitchers State              629 non-null    object
 21  Yearly Revenue              343 non-null    float64
 22  Monthly Sales               287 non-null    float64
```

23	Gross Margin	162 non-null	float64
24	Net Margin	93 non-null	float64
25	EBITDA	81 non-null	float64
26	Cash Burn	97 non-null	object
27	SKUs	43 non-null	float64
28	Has Patents	62 non-null	object
29	Bootstrapped	114 non-null	object
30	Part of Match off	6 non-null	object
31	Original Ask Amount	634 non-null	float64
32	Original Offered Equity	634 non-null	float64
33	Valuation Requested	634 non-null	float64
34	Received Offer	634 non-null	int64
35	Accepted Offer	423 non-null	float64
36	Total Deal Amount	360 non-null	float64
37	Total Deal Equity	360 non-null	float64
38	Total Deal Debt	75 non-null	float64
39	Debt Interest	58 non-null	float64
40	Deal Valuation	359 non-null	float64
41	Number of Sharks in Deal	360 non-null	float64
42	Deal Has Conditions	37 non-null	object
43	Royalty Percentage	36 non-null	float64
44	Royalty Recouped Amount	36 non-null	float64
45	Advisory Shares Equity	11 non-null	float64
46	Namita Investment Amount	114 non-null	float64
47	Namita Investment Equity	114 non-null	float64
48	Namita Debt Amount	21 non-null	float64
49	Vineeta Investment Amount	89 non-null	float64
50	Vineeta Investment Equity	89 non-null	float64
51	Vineeta Debt Amount	15 non-null	float64
52	Anupam Investment Amount	102 non-null	float64
53	Anupam Investment Equity	102 non-null	float64
54	Anupam Debt Amount	9 non-null	float64
55	Aman Investment Amount	141 non-null	float64
56	Aman Investment Equity	141 non-null	float64
57	Aman Debt Amount	19 non-null	float64
58	Peyush Investment Amount	104 non-null	float64
59	Peyush Investment Equity	104 non-null	float64
60	Peyush Debt Amount	13 non-null	float64
61	Ritesh Investment Amount	52 non-null	float64
62	Ritesh Investment Equity	52 non-null	float64
63	Ritesh Debt Amount	14 non-null	float64
64	Amit Investment Amount	35 non-null	float64
65	Amit Investment Equity	35 non-null	float64
66	Amit Debt Amount	7 non-null	float64
67	Guest Investment Amount	63 non-null	float64
68	Guest Investment Equity	63 non-null	float64
69	Guest Debt Amount	6 non-null	float64
70	Invested Guest Name	63 non-null	object

```
 71  All Guest Names            311 non-null    object
 72  Namita Present             495 non-null    float64
 73  Vineeta Present            428 non-null    float64
 74  Anupam Present             548 non-null    float64
 75  Aman Present               556 non-null    float64
 76  Peyush Present             388 non-null    float64
 77  Ritesh Present             138 non-null    float64
 78  Amit Present               137 non-null    float64
 79  Guest Present              311 non-null    float64
dtypes: float64(55), int64(6), object(19)
memory usage: 396.4+ KB
```

[7]:
```
df.describe()
```

[7]:
```
       Season Number  Episode Number  Pitch Number   Started in  \
count     634.000000      634.000000    634.000000   456.000000
mean        2.500000       24.055205    317.500000  2019.280702
std         1.106137       15.102549    183.164316     3.082767
min         1.000000        0.000000      1.000000  1995.000000
25%         2.000000       11.000000    159.250000  2018.000000
50%         2.000000       24.000000    317.500000  2020.000000
75%         3.000000       36.000000    475.750000  2021.000000
max         4.000000       52.000000    634.000000  2024.000000

       Number of Presenters  Male Presenters  Female Presenters  \
count            634.000000       568.000000         382.000000
mean               2.000000         1.593310           0.942408
std                0.796913         0.849374           0.624800
min                1.000000         0.000000           0.000000
25%                1.000000         1.000000           1.000000
50%                2.000000         1.000000           1.000000
75%                2.000000         2.000000           1.000000
max                6.000000         6.000000           3.000000

       Transgender Presenters  Couple Presenters  Yearly Revenue  \
count                     3.0         634.000000      343.000000
mean                      1.0           0.176656      665.909621
std                       0.0           0.381679     1619.021667
min                       1.0           0.000000        0.000000
25%                       1.0           0.000000       90.000000
50%                       1.0           0.000000      230.000000
75%                       1.0           0.000000      633.000000
max                       1.0           1.000000    18700.000000

       Monthly Sales  Gross Margin  Net Margin     EBITDA        SKUs  \
count     287.000000    162.000000   93.000000  81.000000   43.000000
mean       76.120251     55.179012   20.559140  12.597531  265.441860
```

```
std       233.310816    20.495388   12.930804  15.323968   923.926592
min         0.000000     3.000000    1.000000 -39.000000     1.000000
25%         7.750000    40.500000   10.000000   5.000000    12.000000
50%        25.000000    56.000000   20.000000  12.000000    38.000000
75%        69.500000    70.000000   30.000000  19.000000   145.000000
max      3500.000000   150.000000   62.000000  80.000000  6000.000000

       Original Ask Amount  Original Offered Equity  Valuation Requested  \
count           634.000000               634.000000           634.000000
mean            128.724785                 3.580331          5299.153323
std            1190.155243                 3.595335          8253.239217
min               0.000000                 0.200000             0.000000
25%              50.000000                 1.000000          1250.000000
50%              70.000000                 2.500000          3000.000000
75%             100.000000                 5.000000          6000.000000
max           30000.000000                30.000000        120000.000000

       Received Offer  Accepted Offer  Total Deal Amount  Total Deal Equity  \
count      634.000000      423.000000         360.000000         360.000000
mean         0.667192        0.851064          73.013649           7.628667
std          0.471590        0.356447          55.329270           8.818134
min          0.000000        0.000000           0.000000           0.500000
25%          0.000000        1.000000          40.000000           2.322500
50%          1.000000        1.000000          60.000000           5.000000
75%          1.000000        1.000000         100.000000          10.000000
max          1.000000        1.000000         500.000000          75.000000

       Total Deal Debt  Debt Interest  Deal Valuation  \
count        75.000000      58.000000      359.000000
mean         49.946667      10.258621     2479.379763
std          34.787344       2.970973     3342.992929
min          20.000000       0.000000        0.000000
25%          27.500000      10.000000      500.000000
50%          41.000000      10.000000     1250.000000
75%          50.000000      12.000000     2992.537313
max         200.000000      18.000000    25000.000000

       Number of Sharks in Deal  Royalty Percentage  Royalty Recouped Amount  \
count                360.000000           36.000000                36.000000
mean                   1.955556            1.569444               106.729167
std                    1.123674            1.056631                60.199946
min                    1.000000            0.500000                25.000000
25%                    1.000000            1.000000                70.000000
50%                    2.000000            1.000000               100.000000
75%                    2.000000            2.000000               142.500000
max                    5.000000            5.000000               300.000000
```

	Advisory Shares Equity	Namita Investment Amount	\
count	11.000000	114.000000	
mean	1.480000	35.630169	
std	0.862844	22.446143	
min	0.250000	0.000016	
25%	0.800000	20.000000	
50%	1.500000	30.000000	
75%	2.350000	50.000000	
max	2.630000	100.000000	

	Namita Investment Equity	Namita Debt Amount	\
count	114.000000	21.000000	
mean	3.508172	42.825048	
std	4.697263	41.373381	
min	0.200000	10.000000	
25%	1.000000	20.000000	
50%	2.000000	35.000000	
75%	4.000000	50.000000	
max	25.000000	200.000000	

	Vineeta Investment Amount	Vineeta Investment Equity	\
count	89.000000	89.000000	
mean	33.565107	3.646326	
std	21.806716	4.256047	
min	0.002500	0.200000	
25%	20.000000	1.000000	
50%	26.660000	2.500000	
75%	50.000000	5.000000	
max	100.000000	25.000000	

	Vineeta Debt Amount	Anupam Investment Amount	\
count	15.000000	102.000000	
mean	27.388400	33.307164	
std	15.170918	22.400433	
min	12.500000	0.000000	
25%	15.830000	20.000000	
50%	20.000000	25.000000	
75%	40.000000	50.000000	
max	50.000000	100.000000	

	Anupam Investment Equity	Anupam Debt Amount	Aman Investment Amount	\
count	102.000000	9.000000	141.000000	
mean	4.673245	25.000000	38.678848	
std	4.987685	15.051993	33.987022	
min	0.166000	10.000000	0.000000	
25%	1.212500	15.000000	20.000000	
50%	2.750000	20.000000	33.330000	

75%	6.000000	25.000000	50.000000
max	25.000000	50.000000	300.000000

	Aman Investment Equity	Aman Debt Amount	Peyush Investment Amount \
count	141.000000	19.000000	104.000000
mean	3.066061	38.043158	39.162642
std	4.009137	21.038993	53.971928
min	0.166000	10.000000	0.000000
25%	1.000000	22.500000	20.000000
50%	2.000000	35.000000	30.000000
75%	3.750000	47.500000	50.000000
max	40.000000	80.000000	500.000000

	Peyush Investment Equity	Peyush Debt Amount	Ritesh Investment Amount \
count	104.00000	13.000000	52.000000
mean	5.68201	29.000000	42.278548
std	11.06271	15.209646	26.557992
min	0.16600	10.000000	0.002500
25%	1.00000	22.000000	25.000000
50%	2.00000	25.000000	36.000000
75%	5.00000	30.000000	50.000000
max	75.00000	60.000000	150.000000

	Ritesh Investment Equity	Ritesh Debt Amount	Amit Investment Amount \
count	52.000000	14.000000	35.000000
mean	2.260394	45.594714	35.268571
std	2.280346	47.238364	26.540778
min	0.200000	15.000000	3.500000
25%	0.787500	25.000000	15.830000
50%	2.000000	27.500000	25.000000
75%	2.500000	47.915000	50.000000
max	12.500000	200.000000	100.000000

	Amit Investment Equity	Amit Debt Amount	Guest Investment Amount \
count	35.000000	7.000000	63.000000
mean	4.287046	35.000000	45.693369
std	4.713601	18.257419	45.483364
min	0.330000	10.000000	0.000253
25%	1.375000	22.500000	20.000000
50%	2.500000	40.000000	30.000000
75%	5.000000	42.500000	50.000000
max	20.000000	65.000000	250.000000

	Guest Investment Equity	Guest Debt Amount	Namita Present \
count	63.000000	6.000000	495.0
mean	4.035135	37.943333	1.0
std	4.331948	31.921392	0.0

```
min                   0.500000          12.500000             1.0
25%                   1.365000          17.500000             1.0
50%                   2.500000          29.750000             1.0
75%                   5.000000          39.870000             1.0
max                  25.000000          99.000000             1.0

       Vineeta Present  Anupam Present  Aman Present  Peyush Present  \
count            428.0           548.0         556.0           388.0
mean               1.0             1.0           1.0             1.0
std                0.0             0.0           0.0             0.0
min                1.0             1.0           1.0             1.0
25%                1.0             1.0           1.0             1.0
50%                1.0             1.0           1.0             1.0
75%                1.0             1.0           1.0             1.0
max                1.0             1.0           1.0             1.0

       Ritesh Present  Amit Present  Guest Present
count           138.0         137.0     311.000000
mean              1.0           1.0       1.131833
std               0.0           0.0       0.338854
min               1.0           1.0       1.000000
25%               1.0           1.0       1.000000
50%               1.0           1.0       1.000000
75%               1.0           1.0       1.000000
max               1.0           1.0       2.000000
```

[8]:
```
df.isnull().sum().to_frame().T
```

[8]:
```
   Season Number  Startup Name  Episode Number  Pitch Number  Season Start  \
0              0             0               0             0             0

   Season End  Original Air Date  Episode Title  Anchor  Industry  \
0         156                 31              0       0         0

   Business Description  Company Website  Started in  Number of Presenters  \
0                     0               16         178                     0

   Male Presenters  Female Presenters  Transgender Presenters  \
0               66                252                     631

   Couple Presenters  Pitchers Average Age  Pitchers City  Pitchers State  \
0                  0                     0              8               5

   Yearly Revenue  Monthly Sales  Gross Margin  Net Margin  EBITDA  Cash Burn  \
0             291            347           472         541     553        537

   SKUs  Has Patents  Bootstrapped  Part of Match off  Original Ask Amount  \
```

```
0  591         572          520                628                      0

   Original Offered Equity  Valuation Requested  Received Offer  \
0                        0                    0               0

   Accepted Offer  Total Deal Amount  Total Deal Equity  Total Deal Debt  \
0             211                274                274              559

   Debt Interest  Deal Valuation  Number of Sharks in Deal  \
0            576             275                       274

   Deal Has Conditions  Royalty Percentage  Royalty Recouped Amount  \
0                  597                 598                      598

   Advisory Shares Equity  Namita Investment Amount  Namita Investment Equity  \
0                     623                       520                       520

   Namita Debt Amount  Vineeta Investment Amount  Vineeta Investment Equity  \
0                 613                        545                        545

   Vineeta Debt Amount  Anupam Investment Amount  Anupam Investment Equity  \
0                  619                       532                       532

   Anupam Debt Amount  Aman Investment Amount  Aman Investment Equity  \
0                 625                     493                     493

   Aman Debt Amount  Peyush Investment Amount  Peyush Investment Equity  \
0               615                       530                       530

   Peyush Debt Amount  Ritesh Investment Amount  Ritesh Investment Equity  \
0                 621                       582                       582

   Ritesh Debt Amount  Amit Investment Amount  Amit Investment Equity  \
0                 620                     599                     599

   Amit Debt Amount  Guest Investment Amount  Guest Investment Equity  \
0               627                      571                      571

   Guest Debt Amount  Invested Guest Name  All Guest Names  Namita Present  \
0                628                  571              323             139

   Vineeta Present  Anupam Present  Aman Present  Peyush Present  \
0              206              86            78             246

   Ritesh Present  Amit Present  Guest Present
0             496           497            323
```

Checking Missing value percentage for each column

[9]:
```
for i in df.columns:
    if df[i].isnull().sum()>0:
        print(i,"-->",df[i].isnull().sum()*100/df.shape[0],"%")
```

```
Season End --> 24.605678233438486 %
Original Air Date --> 4.889589905362776 %
Company Website --> 2.5236593059936907 %
Started in --> 28.07570977917981 %
Male Presenters --> 10.410094637223974 %
Female Presenters --> 39.74763406940063 %
Transgender Presenters --> 99.52681388012618 %
Pitchers City --> 1.2618296529968454 %
Pitchers State --> 0.7886435331230284 %
Yearly Revenue --> 45.89905362776025 %
Monthly Sales --> 54.73186119873817 %
Gross Margin --> 74.44794952681389 %
Net Margin --> 85.33123028391167 %
EBITDA --> 87.22397476340694 %
Cash Burn --> 84.70031545741325 %
SKUs --> 93.21766561514195 %
Has Patents --> 90.22082018927445 %
Bootstrapped --> 82.01892744479495 %
Part of Match off --> 99.05362776025237 %
Accepted Offer --> 33.2807570977918 %
Total Deal Amount --> 43.217665615141954 %
Total Deal Equity --> 43.217665615141954 %
Total Deal Debt --> 88.17034700315457 %
Debt Interest --> 90.85173501577287 %
Deal Valuation --> 43.375394321766564 %
Number of Sharks in Deal --> 43.217665615141954 %
Deal Has Conditions --> 94.16403785488959 %
Royalty Percentage --> 94.3217665615142 %
Royalty Recouped Amount --> 94.3217665615142 %
Advisory Shares Equity --> 98.26498422712933 %
Namita Investment Amount --> 82.01892744479495 %
Namita Investment Equity --> 82.01892744479495 %
Namita Debt Amount --> 96.68769716088327 %
Vineeta Investment Amount --> 85.96214511041009 %
Vineeta Investment Equity --> 85.96214511041009 %
Vineeta Debt Amount --> 97.63406940063092 %
Anupam Investment Amount --> 83.91167192429022 %
Anupam Investment Equity --> 83.91167192429022 %
Anupam Debt Amount --> 98.58044164037855 %
Aman Investment Amount --> 77.7602523659306 %
Aman Investment Equity --> 77.7602523659306 %
Aman Debt Amount --> 97.0031545741325 %
```

```
Peyush Investment Amount --> 83.59621451104101 %
Peyush Investment Equity --> 83.59621451104101 %
Peyush Debt Amount --> 97.94952681388013 %
Ritesh Investment Amount --> 91.7981072555205 %
Ritesh Investment Equity --> 91.7981072555205 %
Ritesh Debt Amount --> 97.79179810725552 %
Amit Investment Amount --> 94.4794952681388 %
Amit Investment Equity --> 94.4794952681388 %
Amit Debt Amount --> 98.89589905362776 %
Guest Investment Amount --> 90.06309148264984 %
Guest Investment Equity --> 90.06309148264984 %
Guest Debt Amount --> 99.05362776025237 %
Invested Guest Name --> 90.06309148264984 %
All Guest Names --> 50.94637223974763 %
Namita Present --> 21.92429022082019 %
Vineeta Present --> 32.49211356466877 %
Anupam Present --> 13.564668769716087 %
Aman Present --> 12.302839116719243 %
Peyush Present --> 38.801261829652994 %
Ritesh Present --> 78.23343848580441 %
Amit Present --> 78.39116719242902 %
Guest Present --> 50.94637223974763 %
```

[10]:
```
df.duplicated().sum()
```

[10]:
```
np.int64(0)
```

Since company website, SKUs, Bootstrapped and Invested Guest Name are not relevant with respect to analysis so we drop it

[11]:
```
df = df.drop(columns=['Company Website','SKUs', 'Bootstrapped', 'Invested Guest␣
↪Name'])
```

[12]:
```
df.head(5)
```

[12]:
```
   Season Number      Startup Name  Episode Number  Pitch Number Season Start  \
0              1      BluePineFoods               1             1    20-Dec-21
1              1       BoozScooters               1             2    20-Dec-21
2              1  HeartUpMySleeves               1             3    20-Dec-21
3              1          TagzFoods               2             4    20-Dec-21
4              1       HeadAndHeart               2             5    20-Dec-21

  Season End Original Air Date                Episode Title           Anchor  \
0   4-Feb-22         20-Dec-21  Badlegi Business Ki Tasveer  Rannvijay Singh
1   4-Feb-22         20-Dec-21  Badlegi Business Ki Tasveer  Rannvijay Singh
2   4-Feb-22         20-Dec-21  Badlegi Business Ki Tasveer  Rannvijay Singh
3   4-Feb-22         21-Dec-21      Insaan, Ideas Aur Sapne  Rannvijay Singh
4   4-Feb-22         21-Dec-21      Insaan, Ideas Aur Sapne  Rannvijay Singh
```

```
                    Industry  \
0            Food and Beverage
1  Vehicles/Electrical Vehicles
2               Beauty/Fashion
3            Food and Beverage
4           Children/Education

                         Business Description  Started in  \
0                                Frozen Momos      2016.0
1  Renting e-bike for mobility in private spaces      2017.0
2                          Detachable Sleeves      2021.0
3                 Healthy Potato Chips Snacks      2019.0
4                    Brain Development Course      2015.0

   Number of Presenters  Male Presenters  Female Presenters  \
0                     3              2.0                1.0
1                     1              1.0                NaN
2                     1              NaN                1.0
3                     2              2.0                NaN
4                     4              1.0                3.0

   Transgender Presenters  Couple Presenters Pitchers Average Age  \
0                     NaN                  0               Middle
1                     NaN                  0                Young
2                     NaN                  0                Young
3                     NaN                  0               Middle
4                     NaN                  1               Middle

  Pitchers City Pitchers State  Yearly Revenue  Monthly Sales  Gross Margin  \
0         Delhi          Delhi            95.0            8.0           NaN
1     Ahmedabad        Gujarat             4.0            0.4           NaN
2         Delhi          Delhi             NaN            2.0           NaN
3     Bangalore      Karnataka           700.0            NaN          48.0
4       Patiala         Punjab            30.0            NaN           NaN

   Net Margin  EBITDA Cash Burn Has Patents Part of Match off  \
0         NaN     NaN       NaN         NaN               NaN
1         NaN     NaN       NaN         NaN               NaN
2         NaN     NaN       NaN         NaN               NaN
3         NaN     NaN       NaN         NaN               NaN
4         NaN     NaN       NaN         NaN               NaN

   Original Ask Amount  Original Offered Equity  Valuation Requested  \
0                 50.0                      5.0               1000.0
1                 40.0                     15.0                267.0
2                 25.0                     10.0                250.0
```

```
3                70.0                       1.0               7000.0
4                50.0                       5.0               1000.0

   Received Offer  Accepted Offer  Total Deal Amount  Total Deal Equity  \
0               1             1.0               75.0              16.00
1               1             1.0               40.0              50.00
2               1             1.0               25.0              30.00
3               1             1.0               70.0               2.75
4               0             NaN                NaN                NaN

   Total Deal Debt  Debt Interest  Deal Valuation  Number of Sharks in Deal  \
0              NaN            NaN           469.0                       3.0
1              NaN            NaN            80.0                       2.0
2              NaN            NaN            83.0                       2.0
3              NaN            NaN          2545.0                       1.0
4              NaN            NaN             NaN                       NaN

  Deal Has Conditions  Royalty Percentage  Royalty Recouped Amount  \
0                 NaN                 NaN                      NaN
1                 NaN                 NaN                      NaN
2                 NaN                 NaN                      NaN
3                 NaN                 NaN                      NaN
4                 NaN                 NaN                      NaN

   Advisory Shares Equity  Namita Investment Amount  Namita Investment Equity  \
0                     NaN                       NaN                       NaN
1                     NaN                       NaN                       NaN
2                     NaN                       NaN                       NaN
3                     NaN                       NaN                       NaN
4                     NaN                       NaN                       NaN

   Namita Debt Amount  Vineeta Investment Amount  Vineeta Investment Equity  \
0                 NaN                       25.0                       5.33
1                 NaN                       20.0                      25.00
2                 NaN                       12.5                      15.00
3                 NaN                        NaN                        NaN
4                 NaN                        NaN                        NaN

   Vineeta Debt Amount  Anupam Investment Amount  Anupam Investment Equity  \
0                  NaN                       NaN                       NaN
1                  NaN                       NaN                       NaN
2                  NaN                      12.5                      15.0
3                  NaN                       NaN                       NaN
4                  NaN                       NaN                       NaN

   Anupam Debt Amount  Aman Investment Amount  Aman Investment Equity  \
0                 NaN                    25.0                    5.33
```

```
1                 NaN                      NaN                      NaN
2                 NaN                      NaN                      NaN
3                 NaN                      NaN                      NaN
4                 NaN                      NaN                      NaN

   Aman Debt Amount  Peyush Investment Amount  Peyush Investment Equity  \
0               NaN                       NaN                       NaN
1               NaN                       NaN                       NaN
2               NaN                       NaN                       NaN
3               NaN                       NaN                       NaN
4               NaN                       NaN                       NaN

   Peyush Debt Amount  Ritesh Investment Amount  Ritesh Investment Equity  \
0                 NaN                       NaN                       NaN
1                 NaN                       NaN                       NaN
2                 NaN                       NaN                       NaN
3                 NaN                       NaN                       NaN
4                 NaN                       NaN                       NaN

   Ritesh Debt Amount  Amit Investment Amount  Amit Investment Equity  \
0                 NaN                     NaN                     NaN
1                 NaN                     NaN                     NaN
2                 NaN                     NaN                     NaN
3                 NaN                     NaN                     NaN
4                 NaN                     NaN                     NaN

   Amit Debt Amount  Guest Investment Amount  Guest Investment Equity  \
0               NaN                     25.0                     5.33
1               NaN                     20.0                    25.00
2               NaN                      NaN                      NaN
3               NaN                     70.0                     2.75
4               NaN                      NaN                      NaN

   Guest Debt Amount All Guest Names  Namita Present  Vineeta Present  \
0                NaN  Ashneer Grover             1.0              1.0
1                NaN  Ashneer Grover             1.0              1.0
2                NaN  Ashneer Grover             1.0              1.0
3                NaN  Ashneer Grover             1.0              1.0
4                NaN  Ashneer Grover             1.0              1.0

   Anupam Present  Aman Present  Peyush Present  Ritesh Present  Amit Present  \
0             1.0           1.0             NaN             NaN           NaN
1             1.0           1.0             NaN             NaN           NaN
2             1.0           1.0             NaN             NaN           NaN
3             1.0           1.0             NaN             NaN           NaN
4             1.0           1.0             NaN             NaN           NaN
```

```
   Guest Present
0            1.0
1            1.0
2            1.0
3            1.0
4            1.0
```

[13]:
```
df["Deal Has Conditions"].unique()
```

[13]:
```
array([nan, 'yes'], dtype=object)
```

Here nan refers to no condition, so replacing it with no

[14]:
```
df["Deal Has Conditions"]=df["Deal Has Conditions"].fillna("no")
```

[15]:
```
df["Deal Has Conditions"].unique()
```

[15]:
```
array(['no', 'yes'], dtype=object)
```

[16]:
```
df["All Guest Names"].unique()
```

[16]:
```
array(['Ashneer Grover', 'Ghazal Alagh', 'Ghazal Alagh,Ashneer Grover',
       nan, 'Vikas D Nahar', 'Deepinder Goyal', 'Azhar Iqubal',
       'Radhika Gupta', 'Ronnie Screwvala,Radhika Gupta',
       'Varun Dua,Radhika Gupta', 'Azhar Iqubal,Radhika Gupta',
       'Kunal Bahl', 'Varun Dua', 'Kunal Bahl,Azhar Iqubal',
       'Kunal Bahl,Viraj Bahl', 'Chirag Nakrani',
       'Srikanth Bolla,Chirag Nakrani'], dtype=object)
```

[17]:
```
df["All Guest Names"]=df["All Guest Names"].fillna("not present")
```

[18]:
```
df[['Pitchers City','Pitchers State']]=df[['Pitchers City','Pitchers State']].
 ↪fillna("not present")
```

[19]:
```
df[['Pitchers City','Pitchers State']].isnull().sum()
```

[19]:
```
Pitchers City     0
Pitchers State    0
dtype: int64
```

5 Missing Value Percenatge of numerical columns

[20]:
```
for i in df.columns:
    if df[i].isnull().sum()>0 and (df[i].dtype=="int32" or df[i].
 ↪dtype=="float64"):
        print(i,"-->",df[i].isnull().sum()*100/df.shape[0],"%")
```

```
Started in --> 28.07570977917981 %
Male Presenters --> 10.410094637223974 %
Female Presenters --> 39.74763406940063 %
Transgender Presenters --> 99.52681388012618 %
Yearly Revenue --> 45.89905362776025 %
Monthly Sales --> 54.73186119873817 %
Gross Margin --> 74.44794952681389 %
Net Margin --> 85.33123028391167 %
EBITDA --> 87.22397476340694 %
Accepted Offer --> 33.2807570977918 %
Total Deal Amount --> 43.217665615141954 %
Total Deal Equity --> 43.217665615141954 %
Total Deal Debt --> 88.17034700315457 %
Debt Interest --> 90.85173501577287 %
Deal Valuation --> 43.375394321766564 %
Number of Sharks in Deal --> 43.217665615141954 %
Royalty Percentage --> 94.3217665615142 %
Royalty Recouped Amount --> 94.3217665615142 %
Advisory Shares Equity --> 98.26498422712933 %
Namita Investment Amount --> 82.01892744479495 %
Namita Investment Equity --> 82.01892744479495 %
Namita Debt Amount --> 96.68769716088327 %
Vineeta Investment Amount --> 85.96214511041009 %
Vineeta Investment Equity --> 85.96214511041009 %
Vineeta Debt Amount --> 97.63406940063092 %
Anupam Investment Amount --> 83.91167192429022 %
Anupam Investment Equity --> 83.91167192429022 %
Anupam Debt Amount --> 98.58044164037855 %
Aman Investment Amount --> 77.7602523659306 %
Aman Investment Equity --> 77.7602523659306 %
Aman Debt Amount --> 97.0031545741325 %
Peyush Investment Amount --> 83.59621451104101 %
Peyush Investment Equity --> 83.59621451104101 %
Peyush Debt Amount --> 97.94952681388013 %
Ritesh Investment Amount --> 91.7981072555205 %
Ritesh Investment Equity --> 91.7981072555205 %
Ritesh Debt Amount --> 97.79179810725552 %
Amit Investment Amount --> 94.4794952681388 %
Amit Investment Equity --> 94.4794952681388 %
Amit Debt Amount --> 98.89589905362776 %
Guest Investment Amount --> 90.06309148264984 %
Guest Investment Equity --> 90.06309148264984 %
Guest Debt Amount --> 99.05362776025237 %
Namita Present --> 21.92429022082019 %
Vineeta Present --> 32.49211356466877 %
Anupam Present --> 13.564668769716087 %
Aman Present --> 12.302839116719243 %
Peyush Present --> 38.801261829652994 %
```

```
Ritesh Present --> 78.23343848580441 %
Amit Present --> 78.39116719242902 %
Guest Present --> 50.94637223974763 %
```

Although some numerical columns have over 70% missing values, we will retain them due to their significance in the analysis. Instead of dropping these columns, we will impute the missing values in numerical fields using domain knowledge. However we will eliminate 'Royalty Percentage', 'Royalty Recouped Amount', 'Advisory Shares Equity' because of the very high percentage of missing values (over 94% for all three)

[21]:
```
df.drop(['Royalty Percentage', 'Royalty Recouped Amount', 'Advisory Shares␣
↪Equity'], axis=1, inplace=True)
```

[22]:
```
df[["Number of Presenters","Male Presenters","Female Presenters","Transgender␣
↪Presenters","Couple Presenters"]]
```

[22]:
```
     Number of Presenters  Male Presenters  Female Presenters  \
0                       3              2.0                1.0
1                       1              1.0                NaN
2                       1              NaN                1.0
3                       2              2.0                NaN
4                       4              1.0                3.0
..                    …                …                  …
629                     2              1.0                1.0
630                     2              2.0                0.0
631                     3              0.0                3.0
632                     2              2.0                0.0
633                     2              1.0                1.0

     Transgender Presenters  Couple Presenters
0                       NaN                  0
1                       NaN                  0
2                       NaN                  0
3                       NaN                  0
4                       NaN                  1
..                        …                …
629                     NaN                  1
630                     NaN                  0
631                     NaN                  0
632                     NaN                  0
633                     NaN                  1

[634 rows x 5 columns]
```

Here, Nan represents zero. So will replace it with 0

[23]:
```
presenter = ['Male Presenters', 'Female Presenters', 'Transgender Presenters']
df[presenter] = df[presenter].fillna(0).astype(int)
```

[24]:
```
df[presenter].dtypes
```

[24]:
```
Male Presenters            int64
Female Presenters          int64
Transgender Presenters     int64
dtype: object
```

[25]:
```
df[presenter].isnull().sum()
```

[25]:
```
Male Presenters            0
Female Presenters          0
Transgender Presenters     0
dtype: int64
```

[26]:
```
df["Started in"].unique()
```

[26]:
```
array([2016., 2017., 2021., 2019., 2015., 2005., 2020., 2013., 2012.,
       2018., 1998., 2014., 2022., 2010.,   nan, 2006., 2023., 2024.,
       1995.])
```

[27]:
```
df["Started in"]=df["Started in"].fillna(0)
df["Started in"].unique()
```

[27]:
```
array([2016., 2017., 2021., 2019., 2015., 2005., 2020., 2013., 2012.,
       2018., 1998., 2014., 2022., 2010.,    0., 2006., 2023., 2024.,
       1995.])
```

[28]:
```
df["Started in"]=df["Started in"].astype(int)
```

[29]:
```
df["Started in"].dtypes
```

[29]:
```
dtype('int64')
```

Checking if some Correlation exists

[30]:
```
df[["Yearly Revenue","Monthly Sales","Gross Margin","Net Margin","EBITDA"]].
 ↪corr()
```

[30]:
```
                Yearly Revenue  Monthly Sales  Gross Margin  Net Margin  \
Yearly Revenue        1.000000       0.928754     -0.136249   -0.153632
Monthly Sales         0.928754       1.000000     -0.178171   -0.064078
Gross Margin         -0.136249      -0.178171      1.000000    0.415473
Net Margin           -0.153632      -0.064078      0.415473    1.000000
EBITDA               -0.008528      -0.086403      0.150085    0.628517

                  EBITDA
Yearly Revenue -0.008528
```

```
Monthly Sales  -0.086403
Gross Margin    0.150085
Net Margin      0.628517
EBITDA          1.000000
```

We can see there a moderate positive correlation between monthly sales and yearly revenue and with rest columns there is no relation

[31]:
```
# Impute Yearly Revenue using Monthly Sales (if both are not null)

mask = df['Yearly Revenue'].isnull() & df['Monthly Sales'].notnull()
df.loc[mask, 'Yearly Revenue'] = df.loc[mask, 'Monthly Sales'] * 12

# Impute Monthly Sales using Yearly Revenue (if both are not null)

mask = df['Monthly Sales'].isnull() & df['Yearly Revenue'].notnull()
df.loc[mask, 'Monthly Sales'] = df.loc[mask, 'Yearly Revenue'] / 12

# For remaining nulls in these columns, we will use median
for col in ['Yearly Revenue', 'Monthly Sales', 'Gross Margin', 'Net Margin',␣
 ↪'EBITDA']:
    df[col] = df[col].fillna(df[col].median())
```

[32]:
```
df[["Yearly Revenue","Monthly Sales","Gross Margin","Net Margin","EBITDA"]].
 ↪isnull().sum()
```

[32]:
```
Yearly Revenue    0
Monthly Sales     0
Gross Margin      0
Net Margin        0
EBITDA            0
dtype: int64
```

[33]:
```
df["Cash Burn"].unique()
```

[33]:
```
array([nan, 'yes'], dtype=object)
```

[34]:
```
df["Cash Burn"]=df["Cash Burn"].fillna("no")
```

[35]:
```
df["Accepted Offer"].unique()
```

[35]:
```
array([ 1., nan,  0.])
```

[36]:
```
df[["Received Offer","Accepted Offer"]]
```

[36]:
```
     Received Offer  Accepted Offer
0                 1             1.0
```

```
1                   1              1.0
2                   1              1.0
3                   1              1.0
4                   0              NaN
..                ...            ...
629                 1              1.0
630                 0              NaN
631                 1              1.0
632                 1              1.0
633                 0              NaN

[634 rows x 2 columns]
```

Nan in accepted offers indicates deals were not finalized so we fill NaN with 0

[37]:
```
df["Accepted Offer"]=df["Accepted Offer"].fillna(0)
```

[38]:
```
df["Accepted Offer"].dtypes
```

[38]:
```
dtype('float64')
```

[39]:
```
df["Accepted Offer"].astype(int)
```

[39]:
```
0      1
1      1
2      1
3      1
4      0
      ..
629    1
630    0
631    1
632    1
633    0
Name: Accepted Offer, Length: 634, dtype: int64
```

[40]:
```
df["Has Patents"].unique()
```

[40]:
```
array([nan, 'yes'], dtype=object)
```

[41]:
```
df["Has Patents"]=df["Has Patents"].fillna("no")
```

[42]:
```
df[["Original Ask Amount","Total Deal Amount","Original Offered Equity","Total␣
 ↪Deal Equity","Total Deal Debt","Debt Interest","Deal Valuation"]]
```

[42]:
```
     Original Ask Amount  Total Deal Amount  Original Offered Equity  \
0                   50.0               75.0                     5.00
```

```
1                  40.0               40.0                    15.00
2                  25.0               25.0                    10.00
3                  70.0               70.0                     1.00
4                  50.0                NaN                     5.00
..                  …                  …                       …
629               100.0               50.0                     1.25
630               100.0                NaN                     3.33
631                21.3               21.3                     7.00
632                80.0               40.0                     2.00
633               150.0                NaN                     1.50

     Total Deal Equity  Total Deal Debt  Debt Interest  Deal Valuation
0                16.00              NaN            NaN      469.000000
1                50.00              NaN            NaN       80.000000
2                30.00              NaN            NaN       83.000000
3                 2.75              NaN            NaN     2545.000000
4                  NaN              NaN            NaN             NaN
..                 …               …             …             …
629               1.00             50.0            9.0     5000.000000
630                NaN              NaN            NaN             NaN
631               7.00              NaN            NaN      304.285714
632               1.00             40.0           10.0     4000.000000
633                NaN              NaN            NaN             NaN

[634 rows x 7 columns]
```

[43]:
```
df[["Total Deal Amount","Original Offered Equity","Total Deal Equity","Total␣
↪Deal Debt","Debt Interest","Deal Valuation"]]=df[["Total Deal␣
↪Amount","Original Offered Equity","Total Deal Equity","Total Deal␣
↪Debt","Debt Interest","Deal Valuation"]].fillna(0)
```

[44]:
```
df["Number of Sharks in Deal"]=df["Number of Sharks in Deal"].fillna(0)
```

[45]:
```
df.columns[38:-9]
```

[45]:
```
Index(['Number of Sharks in Deal', 'Deal Has Conditions',
       'Namita Investment Amount', 'Namita Investment Equity',
       'Namita Debt Amount', 'Vineeta Investment Amount',
       'Vineeta Investment Equity', 'Vineeta Debt Amount',
       'Anupam Investment Amount', 'Anupam Investment Equity',
       'Anupam Debt Amount', 'Aman Investment Amount',
       'Aman Investment Equity', 'Aman Debt Amount',
       'Peyush Investment Amount', 'Peyush Investment Equity',
       'Peyush Debt Amount', 'Ritesh Investment Amount',
       'Ritesh Investment Equity', 'Ritesh Debt Amount',
       'Amit Investment Amount', 'Amit Investment Equity', 'Amit Debt Amount',
       'Guest Investment Amount', 'Guest Investment Equity',
```

```
        'Guest Debt Amount'],
       dtype='object')
```

[46]:
```
df[df.columns[38:-9]]=df[df.columns[38:-9]].fillna(0)
```

[47]:
```
df.columns[-8:]
```

[47]:
```
Index(['Namita Present', 'Vineeta Present', 'Anupam Present', 'Aman Present',
        'Peyush Present', 'Ritesh Present', 'Amit Present', 'Guest Present'],
       dtype='object')
```

[48]:
```
df[df.columns[-8:]]=df[df.columns[-8:]].fillna(0)
```

[49]:
```
df[df.columns[-8:]].isnull().sum().sum()
```

[49]:
```
np.int64(0)
```

[50]:
```
df.isnull().sum().to_frame().T
```

[50]:
```
   Season Number  Startup Name  Episode Number  Pitch Number  Season Start  \
0              0             0               0             0             0

   Season End  Original Air Date  Episode Title  Anchor  Industry  \
0         156                 31              0       0         0

   Business Description  Started in  Number of Presenters  Male Presenters  \
0                     0           0                     0                0

   Female Presenters  Transgender Presenters  Couple Presenters  \
0                  0                       0                  0

   Pitchers Average Age  Pitchers City  Pitchers State  Yearly Revenue  \
0                     0              0               0               0

   Monthly Sales  Gross Margin  Net Margin  EBITDA  Cash Burn  Has Patents  \
0              0             0           0       0          0            0

   Part of Match off  Original Ask Amount  Original Offered Equity  \
0                628                    0                        0

   Valuation Requested  Received Offer  Accepted Offer  Total Deal Amount  \
0                    0               0               0                  0

   Total Deal Equity  Total Deal Debt  Debt Interest  Deal Valuation  \
0                  0                0              0               0

   Number of Sharks in Deal  Deal Has Conditions  Namita Investment Amount  \
```

```
0                          0                    0                          0

   Namita Investment Equity  Namita Debt Amount  Vineeta Investment Amount  \
0                          0                   0                          0

   Vineeta Investment Equity  Vineeta Debt Amount  Anupam Investment Amount  \
0                           0                    0                         0

   Anupam Investment Equity  Anupam Debt Amount  Aman Investment Amount  \
0                          0                   0                       0

   Aman Investment Equity  Aman Debt Amount  Peyush Investment Amount  \
0                        0                 0                         0

   Peyush Investment Equity  Peyush Debt Amount  Ritesh Investment Amount  \
0                          0                   0                         0

   Ritesh Investment Equity  Ritesh Debt Amount  Amit Investment Amount  \
0                          0                   0                       0

   Amit Investment Equity  Amit Debt Amount  Guest Investment Amount  \
0                        0                 0                        0

   Guest Investment Equity  Guest Debt Amount  All Guest Names  \
0                         0                  0                0

   Namita Present  Vineeta Present  Anupam Present  Aman Present  \
0               0                0               0             0

   Peyush Present  Ritesh Present  Amit Present  Guest Present
0               0               0             0              0
```

[51]:
```
df["Season End"]
```

[51]:
```
0      4-Feb-22
1      4-Feb-22
2      4-Feb-22
3      4-Feb-22
4      4-Feb-22
         …
629         NaN
630         NaN
631         NaN
632         NaN
633         NaN
Name: Season End, Length: 634, dtype: object
```

[52]:
```
df['Season End'] = df['Season End'].fillna('Unknown')
df['Original Air Date'] = df['Original Air Date'].fillna('Unknown')
```

[53]:
```
df["Part of Match off"]
```

[53]:
```
0      NaN
1      NaN
2      NaN
3      NaN
4      NaN
      …
629    NaN
630    NaN
631    NaN
632    NaN
633    NaN
Name: Part of Match off, Length: 634, dtype: object
```

[54]:
```
df["Part of Match off"].unique()
```

[54]:
```
array([nan, 'yes'], dtype=object)
```

[55]:
```
df["Part of Match off"]=df["Part of Match off"].fillna("no")
```

[56]:
```
df.isnull().sum().sum()
```

[56]:
```
np.int64(0)
```

Now, there is no null in the dataset

[57]:
```
df.duplicated().sum()
```

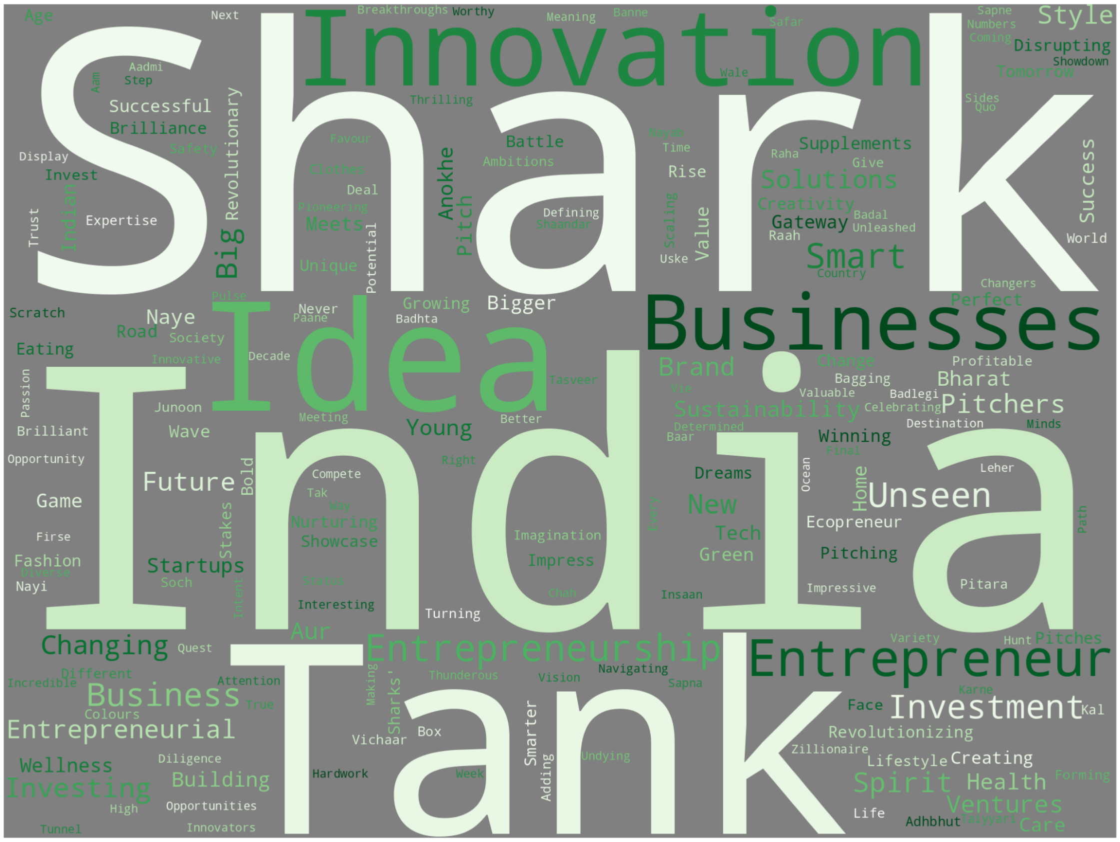
[57]:
```
np.int64(0)
```

No Duplicate in the dataset

6 Exploratory Data Analysis

[58]:
```
text = " Shark Tank India ".join(cat for cat in df.loc[df['Episode Title'].
  ↪notnull()]['Episode Title'])
stop_words = list(STOPWORDS) + ["Ka", "Ki", "Ke", "Ko", "Se", "Hai", "Ek"]
wordcloud = WordCloud(width=2000, height=1500, stopwords=stop_words,␣
  ↪background_color='Grey', colormap='Greens', collocations=False,␣
  ↪random_state=2025).generate(text)
plt.figure(figsize=(25,20))
plt.imshow(wordcloud)
plt.axis("off")
```

```
plt.show()
```



7 All seasons of SHARK TANK INDIA was broadcasted in SonyLiv OTT and Sony TV

[59]:
```
print(df['Season Number'].max(), "Total seasons in Indian SharkTank \n")
print(df['Pitch Number'].max(), "Startups came for pitching \n")
```

```
4 Total seasons in Indian SharkTank

634 Startups came for pitching
```

[60]:
```
shark_tank_season1 = df.loc[df['Season Number']==1]
shark_tank_season1_without_unseen = df.loc[(df['Season Number']==1) &␣
↪(df['Episode Number']!=0)]
shark_tank_season2 = df.loc[df['Season Number']==2]
shark_tank_season3 = df.loc[(df['Season Number']==3) | (df['Season Number'].
↪isnull())]
```

```
shark_tank_season4 = df.loc[df['Season Number']==4]
```

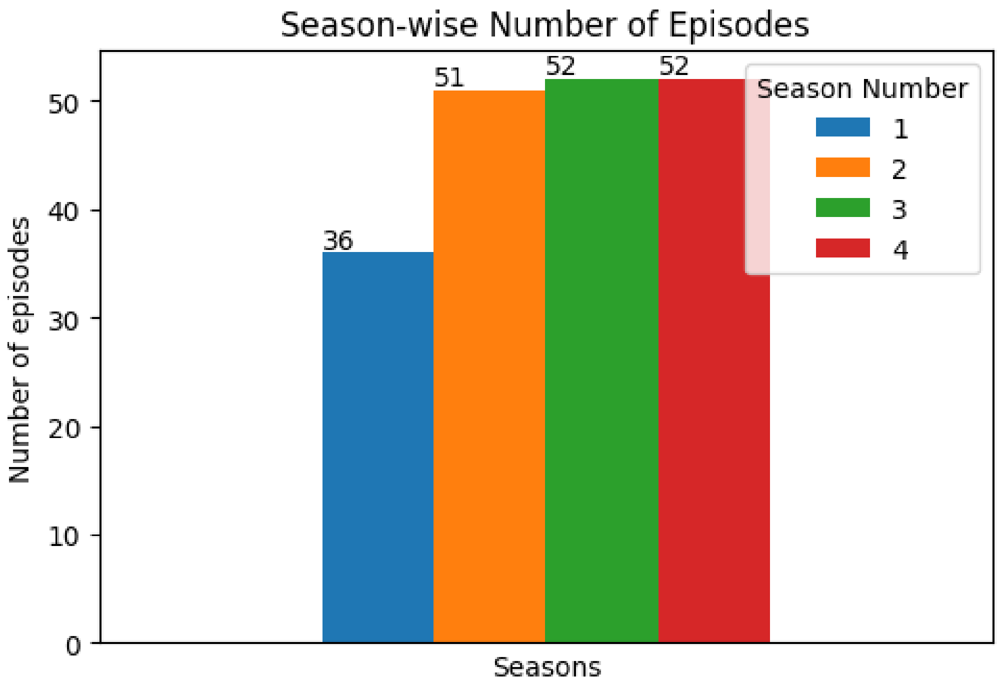
[61]:
```
print("In Season 1, in", shark_tank_season1['Episode Number'].max(), "episodes,␣
↪there were", shark_tank_season1.loc[shark_tank_season1['Episode Number']!
↪=0]['Startup Name'].count(), "actual pitches and", shark_tank_season1.
↪loc[shark_tank_season1['Episode Number']==0]['Startup Name'].count(),"unseen␣
↪pitches\n")
print("In Season 2, in", shark_tank_season2['Episode Number'].max(), "episodes,␣
↪there were", shark_tank_season2.loc[shark_tank_season2['Episode Number']!
↪=0]['Startup Name'].count(), "actual pitches and", shark_tank_season2.
↪loc[shark_tank_season2['Episode Number']==0]['Startup Name'].count(),"unseen␣
↪pitch\n")
print("In Season 3, in", shark_tank_season3['Episode Number'].max(), "episodes,␣
↪there were", shark_tank_season3.loc[shark_tank_season3['Episode Number']!
↪=0]['Startup Name'].count(), "actual pitches\n")
print("In Season 4, in", shark_tank_season4['Episode Number'].max(), "episodes,␣
↪there were", shark_tank_season4.loc[shark_tank_season4['Episode Number']!
↪=0]['Startup Name'].count(), "actual pitches\n")
```

```
In Season 1, in 36 episodes, there were 122 actual pitches and 30 unseen pitches

In Season 2, in 51 episodes, there were 168 actual pitches and 1 unseen pitch

In Season 3, in 52 episodes, there were 157 actual pitches

In Season 4, in 52 episodes, there were 156 actual pitches
```

8 Pitch Statistics

[62]:
```
# Season-wise number of episodes
tmp = pd.pivot_table(df, values='Episode Number', columns='Season Number',␣
↪aggfunc='max')
print(tmp)
ax = tmp.plot.bar(figsize=(6,4), title="Season-wise Number of Episodes")
plt.xlabel("Seasons")
plt.ylabel("Number of episodes")
plt.xticks([])
for t in ax.patches:
    if (np.isnan(float(t.get_height()))):
        ax.annotate(0, (t.get_x(), 0))
    else:
        ax.annotate(str(format(int(t.get_height()), ',d')), (t.get_x(), t.
↪get_height()*1.01))
```

```
Season Number    1   2   3   4
Episode Number  36  51  52  52
```



[63]:
```
# Season-wise number of pitches
tmp = df['Season Number'].value_counts().sort_values()
fig = px.bar(tmp, x=tmp.values, title="<b> Season-wise number of pitches</b>",␣
↪template='ggplot2', text=tmp, width=520, height=400)
fig.update_yaxes(tickvals=list(range(6)))
fig.update_xaxes(visible=False)
fig.show(renderer="iframe")
```

[105]:
```
import pandas as pd
import plotly.express as px

# Mark pitches as funded if Total Deal Amount > 0
df['Got Funded'] = df['Total Deal Amount'] > 0

# Count funded pitches per season
funded_pitches = df[df['Got Funded']]
tmp = funded_pitches['Season Number'].value_counts().sort_index()
tmp = tmp.reset_index()
tmp.columns = ['Season', 'Funded Pitches']

# Convert Season to string for clean x-axis
tmp['Season'] = tmp['Season'].astype(str)

# Plot
```

```
fig = px.bar(
    tmp,
    x='Season',
    y='Funded Pitches',
    text='Funded Pitches',
    title='<b>Season-wise Number of Funded Pitches</b>',
    template='plotly_white',
)

fig.update_traces(textposition='outside')
fig.update_layout(
    xaxis_title='Season',
    yaxis_title='Number of Funded Pitches',
    title_x=0.5,
    bargap=0.3
)

fig.show(renderer='iframe')
```

Among all seasons, Season 2 saw the most pitches getting funded, whereas Season 1 had the lowest funding activity.

[65]:
```
print("Number of pitches per episode was:\n")
print(df.loc[df['Episode Number']!=0][['Season Number','Episode Number']].
 ↪value_counts().sort_values(ascending=True).unique())
```

```
Number of pitches per episode was:

[2 3 4]
```

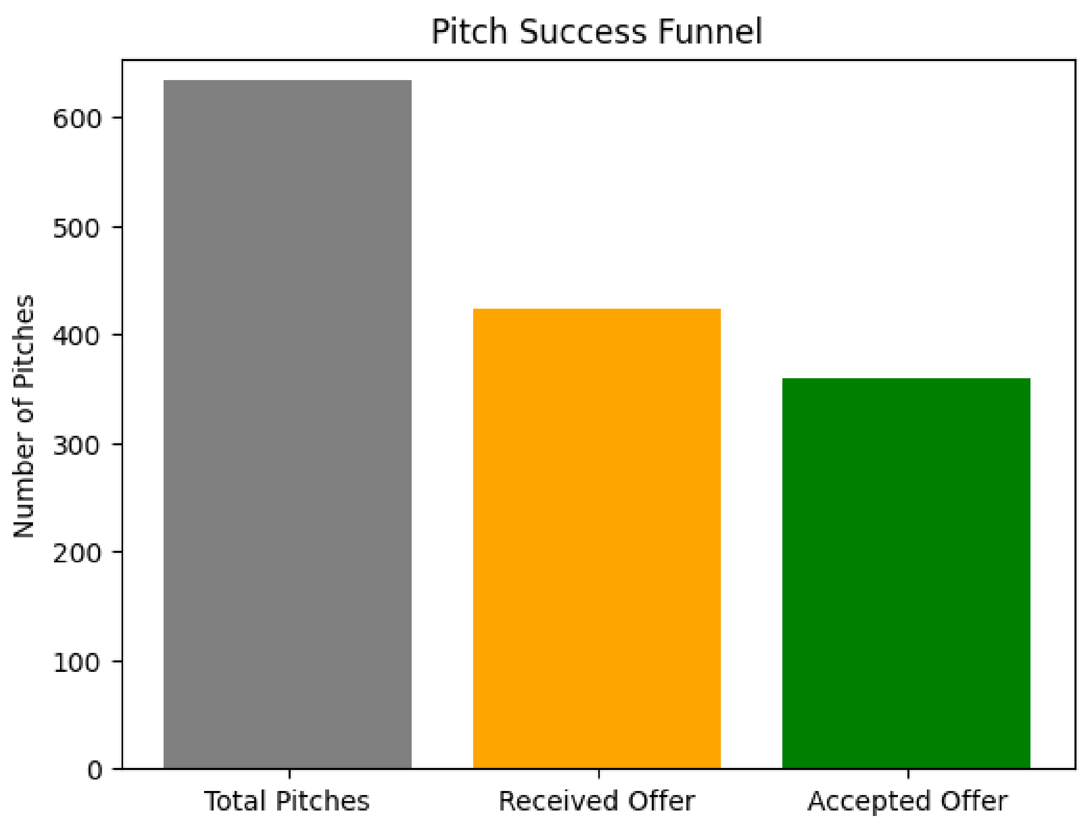
[101]:
```
# Types of industries, came for investment, in current/latest season (4th␣
 ↪season)
tmp = shark_tank_season4['Industry'].value_counts().sort_values(ascending=True)
fig = px.bar(tmp, x=tmp.values, title="<b> Indian Shark Tank in 4th season -␣
 ↪Industry wise startups</b>", template='simple_white', text=tmp, width=630,␣
 ↪height=600)
fig.update_yaxes(title_text="")
fig.update_xaxes(visible=False)
fig.show(renderer="iframe")
```

[106]:
```
#Pitches Analysis
total_pitches = len(df)
offers_received = df['Received Offer'].sum()
offers_accepted = df['Accepted Offer'].sum()

print(f"Total Pitches: {total_pitches}")
print(f"Received Offers: {offers_received}")
```

```
print(f"Accepted Offers: {offers_accepted}")
```

```
Total Pitches: 402
Received Offers: 271
Accepted Offers: 227.0
```

[67]:
```
import matplotlib.pyplot as plt

labels = ['Total Pitches', 'Received Offer', 'Accepted Offer']
values = [total_pitches, offers_received, offers_accepted]

plt.bar(labels, values, color=['gray', 'orange', 'green'])
plt.title("Pitch Success Funnel")
plt.ylabel("Number of Pitches")
plt.ylim(0, max(values) + 20)
plt.show()
```



[68]:
```
import pandas as pd

season_summary = df.groupby('Season Number').agg(
    total_pitches=('Startup Name', 'count'),
```

```
    received_offers=('Received Offer', 'sum'),
    accepted_offers=('Accepted Offer', 'sum')
).reset_index()

season_summary['acceptance_rate'] = (season_summary['accepted_offers'] /␣
↪season_summary['received_offers']).fillna(0)
season_summary['rejection_rate'] = 1 - season_summary['acceptance_rate']

season_summary['acceptance_rate'] = season_summary['acceptance_rate'].round(2)
season_summary['rejection_rate'] = season_summary['rejection_rate'].round(2)

season_summary
```

[68]:

```
   Season Number  total_pitches  received_offers  accepted_offers  \
0              1            152               96             70.0
1              2            169              121            106.0
2              3            157              104             92.0
3              4            156              102             92.0

   acceptance_rate  rejection_rate
0             0.73            0.27
1             0.88            0.12
2             0.88            0.12
3             0.90            0.10
```

[107]:

```
import plotly.graph_objects as go

fig = go.Figure()

# Add Acceptance Rate bar (light green)
fig.add_trace(go.Bar(
    x=season_summary['Season Number'].astype(str),
    y=season_summary['acceptance_rate'],
    name='Acceptance Rate',
    marker_color='lightgreen',
    text=season_summary['acceptance_rate'].apply(lambda x: f"{x:.0%}"),
    textposition='outside'
))

# Add Rejection Rate bar (red)
fig.add_trace(go.Bar(
    x=season_summary['Season Number'].astype(str),
    y=season_summary['rejection_rate'],
    name='Rejection Rate',
    marker_color='red',
    text=season_summary['rejection_rate'].apply(lambda x: f"{x:.0%}"),
    textposition='outside'
```

```
))

# Update layout
fig.update_layout(
    title='<b>Acceptance vs Rejection Rate by Season</b>',
    xaxis_title='Season',
    yaxis_title='Rate',
    barmode='group',
    template='plotly_white',
    title_x=0.5,
    yaxis=dict(tickformat=".0%"),
    uniformtext_minsize=8,
    uniformtext_mode='hide'
)

fig.show(renderer='iframe')
```

Acceptance rates have increased consistently across seasons — from 73% in Season 1 to 90% in Season 4.

This suggests a growing openness among sharks to invest in a broader range of startups or an improvement in the quality of pitches over time.

[109]:
```
# All seasons averages
pivot = pd.pivot_table(df, values=['Original Ask Amount','Total Deal␣
↪Amount','Valuation Requested','Deal Valuation','Original Offered␣
↪Equity','Total Deal Equity'], columns='Season Number', aggfunc=np.mean,␣
↪sort=False)
pivot.style.format('{:.0f}')
```

```
/var/folders/1v/g6bcxzl16s587g__qbw3d0z00000gp/T/ipykernel_55717/1488082707.py:2
: FutureWarning:

The provided callable <function mean at 0x108333880> is currently using
DataFrameGroupBy.mean. In a future version of pandas, the provided callable will
be used directly. To keep current behavior pass the string "mean" instead.
```

[109]:
```
<pandas.io.formats.style.Styler at 0x1135a2a50>
```

9 Presenter Demographics

[70]:
```
import plotly.express as px

# Categorize pitch type
df['Pitch Type'] = df.apply(
    lambda row: 'Couple' if row['Couple Presenters'] > 0 else
```

```
                ('Solo' if row['Number of Presenters'] == 1 else 'Group'),␣
 ↪axis=1
)

pitch_type_counts = df['Pitch Type'].value_counts().reset_index()
pitch_type_counts.columns = ['Pitch Type', 'Count']

fig = px.pie(
    pitch_type_counts,
    names='Pitch Type',
    values='Count',
    title='<b>Distribution of Pitch Types</b>',
    color_discrete_sequence=px.colors.sequential.RdBu
)
fig.update_traces(textposition='inside', textinfo='percent+label')
fig.show(renderer='iframe')
```

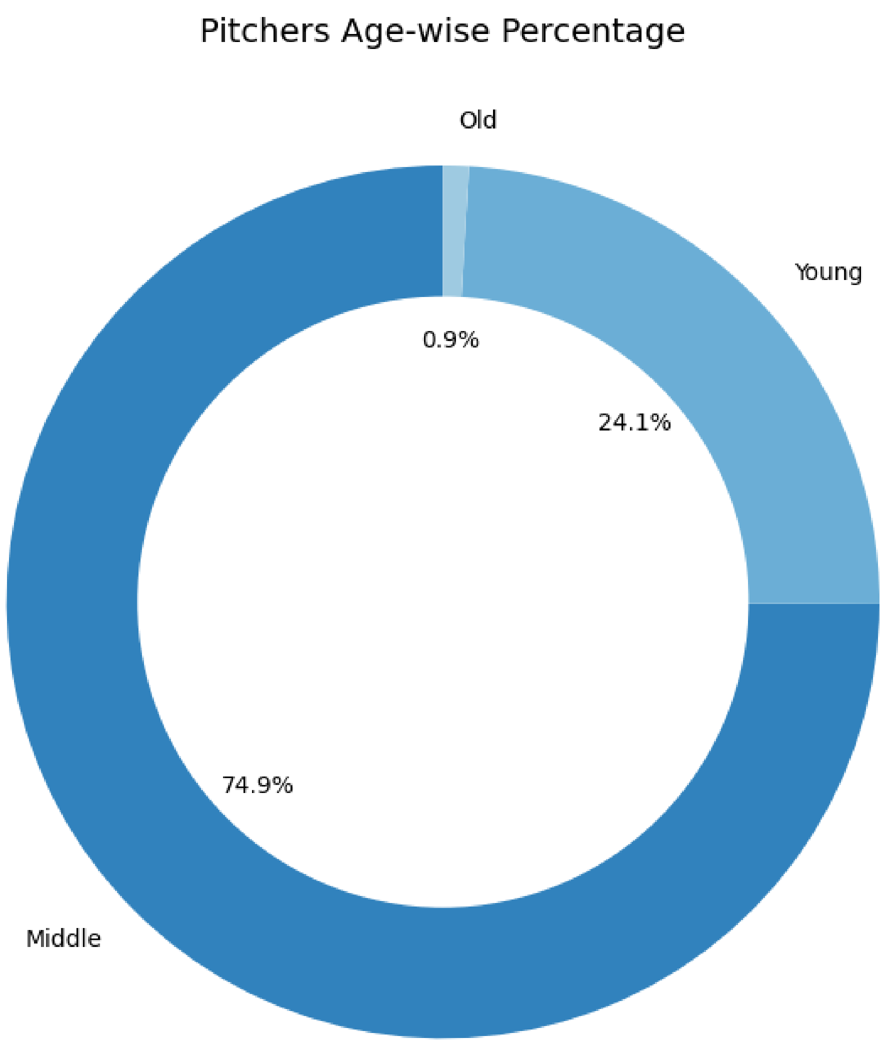
[104]:
```
gender_data = {
    'Male': df['Male Presenters'].sum(),
    'Female': df['Female Presenters'].sum(),
    'Transgender': df['Transgender Presenters'].sum()
}
gender_df = pd.DataFrame(gender_data.items(), columns=['Gender', 'Count'])

fig = px.pie(
    gender_df,
    names='Gender',
    values='Count',
    title='<b>Gender Breakdown of Presenters</b>',
    color_discrete_sequence=['lightblue', 'pink', 'purple']
)
fig.update_traces(textposition='inside', textinfo='percent+label')
fig.show(renderer='iframe')
```

While male entrepreneurs dominate the pitch floor, it's encouraging to see nearly 30% representation by female founders, reflecting growing diversity. However, transgender representation remains minimal, highlighting an opportunity for greater inclusivity in future seasons.

[72]:
```
import matplotlib.pyplot as plt

# Count and percentage
print(df['Pitchers Average Age'].value_counts(), "\n")
print(round(df['Pitchers Average Age'].value_counts(normalize=True)*100).
 ↪astype(str).str.replace('.0', '%', regex=False), "\n")

# Data for plotting
age_counts = df["Pitchers Average Age"].value_counts()
```

```python
# Plot
fig, ax = plt.subplots(figsize=(7, 7))
wedges, texts, autotexts = ax.pie(
    age_counts,
    labels=age_counts.index,
    autopct='%.1f%%',
    startangle=90,
    textprops=dict(color="black"),
    colors=plt.cm.tab20c.colors
)

# Draw white circle to make it a doughnut
centre_circle = plt.Circle((0, 0), 0.70, fc='white')
fig.gca().add_artist(centre_circle)

plt.title("Pitchers Age-wise Percentage", fontsize=14)
plt.tight_layout()
plt.show()
```

```
Pitchers Average Age
Middle    475
Young     153
Old         6
Name: count, dtype: int64

Pitchers Average Age
Middle    75%
Young     24%
Old        1%
Name: proportion, dtype: object
```



[102]:
```
import plotly.express as px

# Prepare Top 10 Cities
top_cities = df['Pitchers City'].value_counts().head(10).reset_index()
top_cities.columns = ['City', 'Count']

# Plot with unique colors per city
fig = px.bar(
    top_cities,
    x='City',
    y='Count',
    color='City',  # Different color per city
```

```
    title='<b>Top 10 Pitcher Cities</b>',
    text='Count',
    template='plotly_white'
)

fig.update_traces(textposition='outside')
fig.update_layout(
    title_x=0.5,
    showlegend=False  # Hide legend if not needed
)

fig.show(renderer='iframe')
```

Mumbai, Delhi, and Bangalore emerge as the leading startup hubs, consistently featuring among the top 10 cities with the highest number of entrepreneurs pitching on Shark Tank India.

10 Business Characteristics

[74]:
```
import pandas as pd
import plotly.express as px

# Group by industry and calculate success rate
industry_success = df.groupby('Industry').agg(
    Total_Pitches=('Startup Name', 'count'),
    Funded_Pitches=('Got Funded', 'sum')
).sort_values(by='Total_Pitches', ascending=False).head(10)

industry_success['Success_Rate'] = round((industry_success['Funded_Pitches'] /
 ↪industry_success['Total_Pitches']) * 100, 1)

# Plot using plotly
fig = px.bar(
    industry_success.reset_index(),
    x='Industry',
    y='Success_Rate',
    text='Success_Rate',
    title='<b>Top 10 Industries and Their Success Rates</b>',
    color='Industry',
    template='plotly_white'
)

fig.update_traces(textposition='outside')
fig.update_layout(showlegend=False, title_x=0.5)
fig.show()
```

The Fitness, Sports & Outdoor sector leads with a remarkable 78.9% success rate, indicating strong investor interest. Meanwhile, traditional sectors like Food & Beverage and Beauty & Fashion still perform well, but face stiffer competition.

[103]:
```
import pandas as pd
import plotly.express as px

# Filter rows where margins are not null
df_filtered = df.dropna(subset=['Gross Margin', 'Net Margin'])

# Plot using plotly for Gross Margin vs Net Margin by Industry
fig = px.scatter(
    df_filtered,
    x='Gross Margin',
    y='Net Margin',
    color='Industry',
    title="<b>Gross Margin vs Net Margin by Industry</b>",
    template='plotly_white',
    labels={'Gross Margin': 'Gross Margin (%)', 'Net Margin': 'Net Margin (%)'}
)

fig.update_layout(title_x=0.5)
fig.show()
```

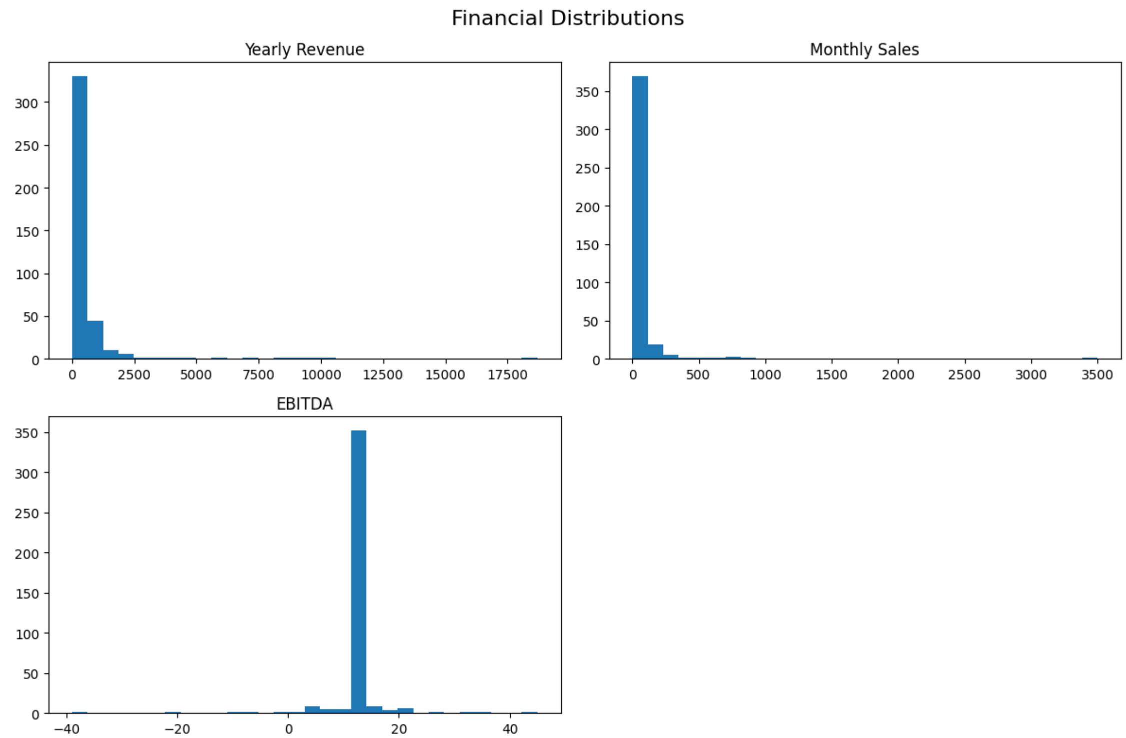
[112]:
```
import pandas as pd
import plotly.express as px

# Ensure 'Started in' contains valid numeric values
df['Started in'] = pd.to_numeric(df['Started in'], errors='coerce')

# Filter out rows where 'Started in' is before 2017
df = df[df['Started in'] >= 2017]

# Clean the 'Season Number' to ensure it's an integer
df['Season Number'] = df['Season Number'].apply(lambda x: int(x) if pd.notna(x)␣
 ↪else None)

# Group by 'Started in' (the year the business was started) and 'Season Number'␣
 ↪to analyze deal success rate
started_vs_season = df.groupby(['Started in', 'Season Number']).agg(
    Total_Pitches=('Startup Name', 'count'),
    Funded_Pitches=('Got Funded', 'sum')
).reset_index()

# Calculate success rate
started_vs_season['Success_Rate'] = round((started_vs_season['Funded_Pitches'] /
 ↪ started_vs_season['Total_Pitches']) * 100, 1)

# Plot using scatter plot
```

```
fig = px.scatter(
    started_vs_season,
    x='Started in',
    y='Success_Rate',
    color='Season Number',
    size='Total_Pitches',
    title="<b>Started In vs Season Year: Are Newer Businesses More Likely to␣
 ↪Get a Deal?</b>",
    template='plotly_white',
    labels={'Started in': 'Year Started', 'Success_Rate': 'Success Rate (%)'},
    hover_data=['Total_Pitches', 'Funded_Pitches']
)

fig.update_layout(
    title_x=0.5,
    xaxis_title="Year Started",
    yaxis_title="Success Rate (%)",
    xaxis=dict(tickmode='linear', tick0=min(started_vs_season['Started in']),␣
 ↪dtick=1),  # Set ticks for years
    coloraxis_showscale=False
)

fig.update_xaxes(showgrid=True)
fig.update_yaxes(showgrid=True)

fig.show()
```

There's no clear evidence that newer startups are significantly more likely to get a deal.

11 Financial Analysis

[113]:
```
# Drop NaNs for distribution plot
df_dist = df[['Yearly Revenue', 'Monthly Sales', 'EBITDA', 'Cash Burn']].
 ↪dropna(how='all')

df_dist.hist(bins=30, figsize=(12, 8), grid=False, color='#1f77b4')
plt.suptitle("Financial Distributions", fontsize=16)
plt.tight_layout()
plt.show()
```



1. Yearly Revenue Observation: Most startups have yearly revenues clustered at the lower end (0–2000), with a steep drop-off as revenue increases.

Interpretation: A large majority of startups have relatively low annual revenues, indicating early-stage or small-scale operations. A few outliers report very high revenue (up to 18,000+), but these are rare.

2. Monthly Sales Observation: Similarly skewed to the left — most startups report monthly sales below 500, with only a few going above 1000.

Interpretation: Like yearly revenue, this shows most startups are in early revenue stages, with limited monthly income. A handful of high-performing startups skew the scale.

3. EBITDA (Earnings Before Interest, Taxes, Depreciation, and Amortization) Observation: A dense cluster near a certain positive EBITDA value (likely around 15–20), but with long tails in both directions, including negative values.

Interpretation:

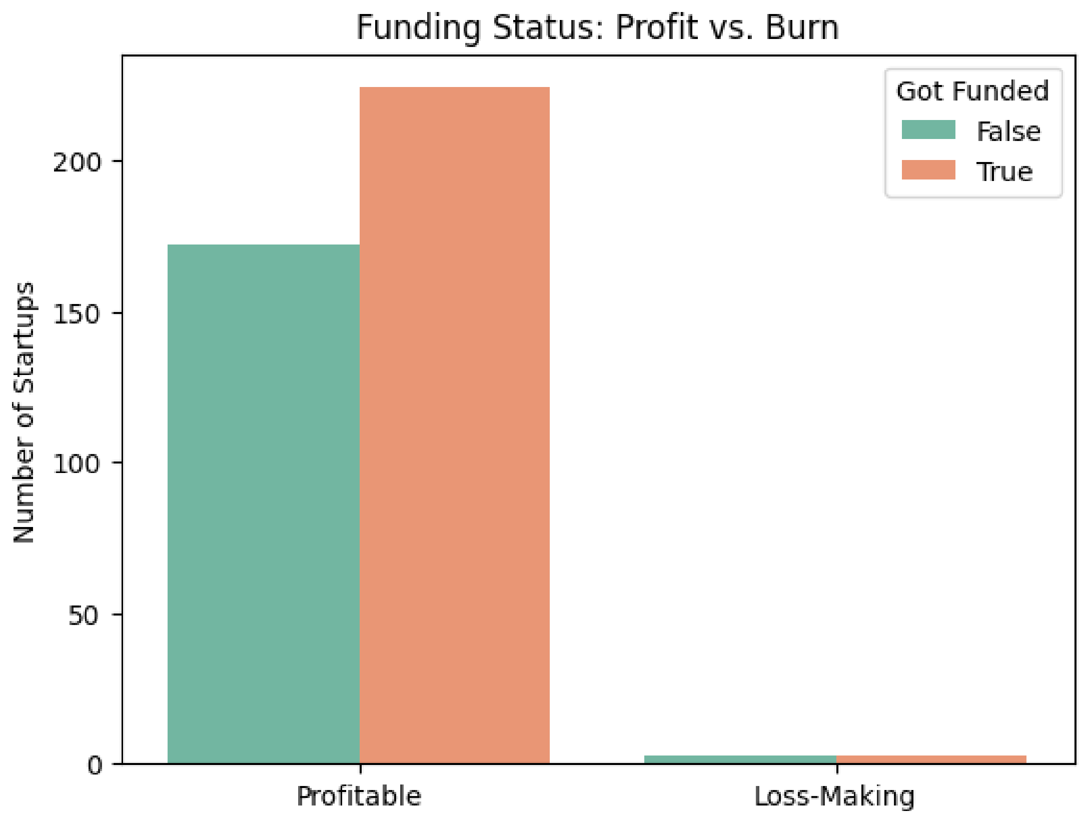
The positive mode shows that many startups are aiming for or reporting small profits.

The left tail shows several startups are operating at a loss.

A few outliers also report high EBITDA.

[78]:
```
df_burn = df[['Cash Burn', 'EBITDA', 'Got Funded']].dropna()

# Classify as 'Profitable' or 'Burning Cash'
```

```
df_burn['Status'] = df_burn.apply(lambda x: 'Profitable' if x['EBITDA'] > 0␣
↪else 'Loss-Making', axis=1)

sns.countplot(data=df_burn, x='Status', hue='Got Funded', palette='Set2')
plt.title('Funding Status: Profit vs. Burn')
plt.xlabel('')
plt.ylabel('Number of Startups')
plt.legend(title='Got Funded')
plt.show()
```



Startups that are already profitable have a substantially higher chance of receiving funding.

[79]:
```
df_val = df[['Valuation Requested', 'Deal Valuation']].dropna()

fig = px.scatter(
    df_val,
    x='Valuation Requested',
    y='Deal Valuation',
    title='Requested vs. Actual Deal Valuation',
    trendline='ols',
```

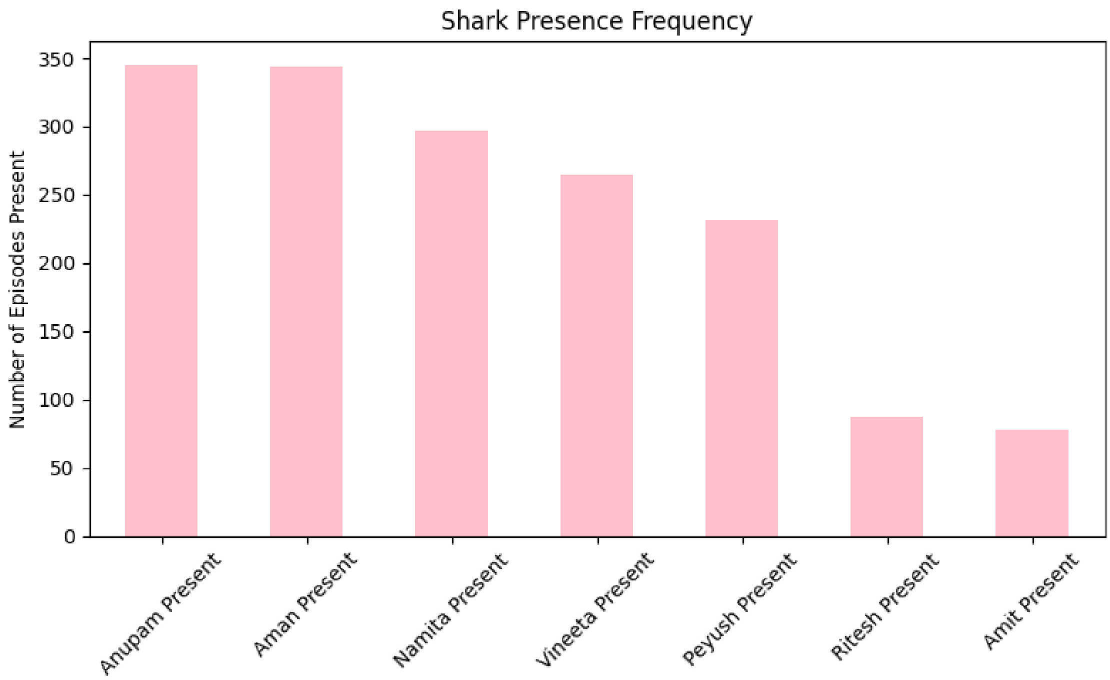
```
    labels={'Valuation Requested': 'Valuation Asked (in Cr)', 'Deal Valuation':␣
 ↪'Final Deal Valuation (in Cr)'},
    template='plotly_white',
    color_discrete_sequence=['#17becf']
)

fig.update_layout(title_x=0.5)
fig.show()
```

12 Shark Participation and Investment Trends

[121]:
```
sharks_names = []

for col in df.columns[41:-13:3]:
    shark = col.split(maxsplit=1)[0]
    sharks_names.append(shark)

print(len(sharks_names), "sharks participated.\n")
print("Following are the names:\n")
print(sharks_names)
```

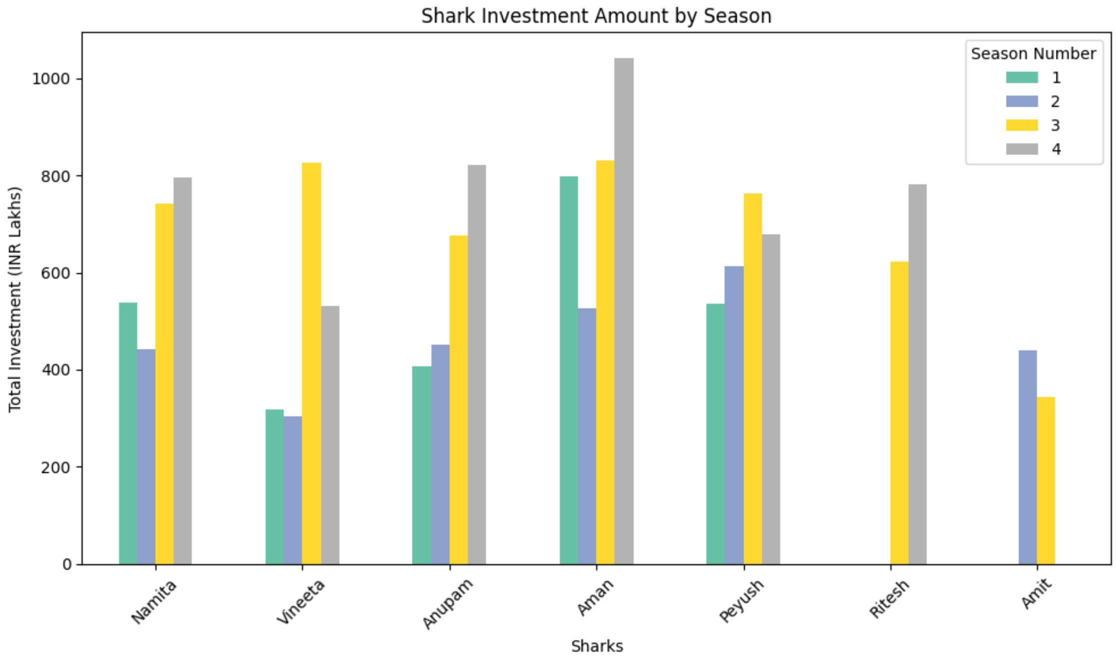
```
8 sharks participated.

Following are the names:

['Namita', 'Vineeta', 'Anupam', 'Aman', 'Peyush', 'Ritesh', 'Amit', 'Guest']
```

[80]:
```
import matplotlib.pyplot as plt

sharks = ['Namita Present', 'Vineeta Present', 'Anupam Present', 'Aman␣
 ↪Present', 'Peyush Present', 'Ritesh Present', 'Amit Present']

presence_counts = df[sharks].sum().sort_values(ascending=False)

plt.figure(figsize=(8, 5))
presence_counts.plot(kind='bar', color='pink')
plt.title("Shark Presence Frequency")
plt.ylabel("Number of Episodes Present")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Anupam and Aman are the most frequently present sharks, each appearing in nearly all episodes, highlighting their consistent participation across seasons.

[92]:
```
shark_cols = {
    'Namita': 'Namita Investment Amount',
    'Vineeta': 'Vineeta Investment Amount',
    'Anupam': 'Anupam Investment Amount',
    'Aman': 'Aman Investment Amount',
    'Peyush': 'Peyush Investment Amount',
    'Ritesh': 'Ritesh Investment Amount',
    'Amit': 'Amit Investment Amount'
}

season_wise = df.groupby('Season Number')[[col for col in shark_cols.values()]].
  ↪sum()
season_wise.columns = shark_cols.keys()
season_wise = season_wise.T

season_wise.plot(kind='bar', figsize=(10, 6), title="Shark Investment Amount by␣
  ↪Season", colormap='Set2')
plt.ylabel("Total Investment (INR Lakhs)")
plt.xlabel("Sharks")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



12.1 INSIGHTS BY SHARK

12.2 Aman Gupta

Consistently the top investor across all seasons.

His investment peaked in Season 4, indicating growing confidence or presence.

12.3 Anupam Mittal

Steady rise from Season 1 to Season 4.

Significantly high jump in Season 4, showing increased involvement.

12.4 Namita Thapar

Shows consistent and growing investment across seasons.

Season 4 marks her highest contribution, after a sharp rise since Season 2.

12.5 Peyush Bansal

Gradual increase till Season 3.

Slight dip or plateau in Season 4.

12.6 Vineeta Singh

Moderate and fairly stable investor.

Season 3 was her peak investment, with a small decline in Season 4.

12.7 Ritesh Agarwal

Appears only in Season 3 and 4.

Strong presence in both, especially Season 4.

12.8 Amit Jain

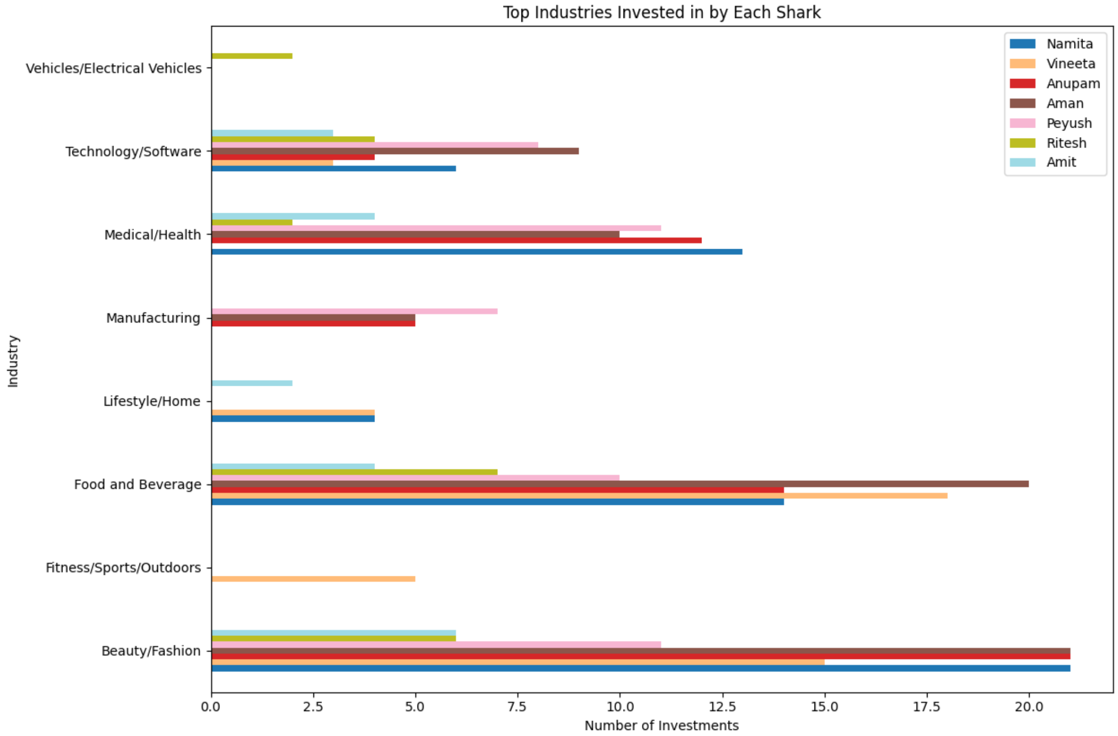
Participated only in Seasons 2 and 3.

Lesser investment compared to others, possibly a guest or late-entrant.

12.9 Overall Analysis

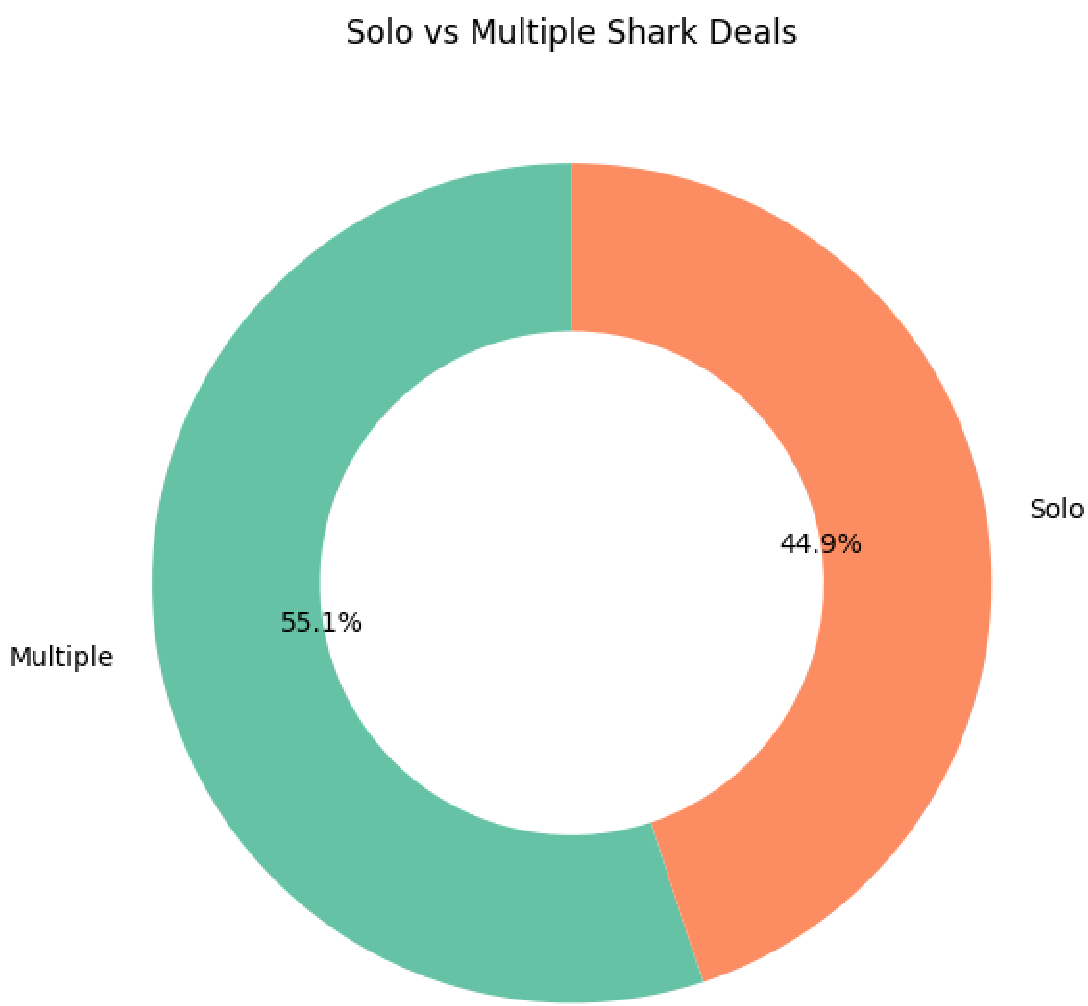
Season 4 had the highest total investments overall, marking it as a high-engagement season.

[82]:
```python
import matplotlib.pyplot as plt

# Shark columns and corresponding names
shark_invest_cols = {
    'Namita': 'Namita Investment Amount',
    'Vineeta': 'Vineeta Investment Amount',
    'Anupam': 'Anupam Investment Amount',
    'Aman': 'Aman Investment Amount',
    'Peyush': 'Peyush Investment Amount',
    'Ritesh': 'Ritesh Investment Amount',
    'Amit': 'Amit Investment Amount'
}

# Dictionary to store industry investment count per shark
shark_industry = {}

for shark, col in shark_invest_cols.items():
    invested_df = df[df[col] > 0]  # Filter where this shark invested
    top_industries = invested_df['Industry'].value_counts().head(5)
    shark_industry[shark] = top_industries

# Convert to a single DataFrame for plotting
shark_industry_df = pd.DataFrame(shark_industry).fillna(0).astype(int)

# Plotting (horizontal)
shark_industry_df.plot(kind='barh', figsize=(12, 8), colormap='tab20',␣
 ↪title='Top Industries Invested in by Each Shark')
plt.xlabel("Number of Investments")
plt.ylabel("Industry")
plt.tight_layout()
plt.show()
```



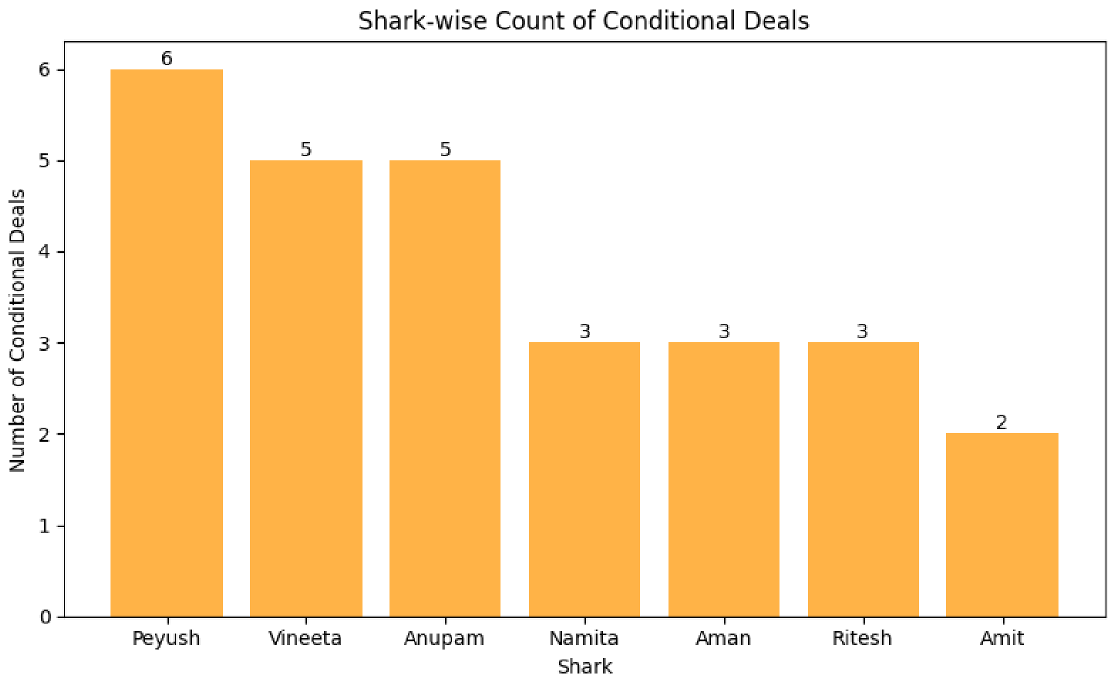
[93]:
```python
# Filter funded deals
funded_df = df[df['Total Deal Amount'] > 0].copy()

# Count number of sharks involved in each deal
funded_df['Sharks Involved'] = (
    (funded_df['Namita Investment Amount'] > 0).astype(int) +
    (funded_df['Vineeta Investment Amount'] > 0).astype(int) +
    (funded_df['Anupam Investment Amount'] > 0).astype(int) +
    (funded_df['Aman Investment Amount'] > 0).astype(int) +
    (funded_df['Peyush Investment Amount'] > 0).astype(int) +
    (funded_df['Ritesh Investment Amount'] > 0).astype(int) +
    (funded_df['Amit Investment Amount'] > 0).astype(int)
)

# Classify deals
funded_df['Deal Type'] = funded_df['Sharks Involved'].apply(lambda x: 'Solo' if␣
↪x == 1 else 'Multiple')

# Count solo vs multiple deals
deal_counts = funded_df['Deal Type'].value_counts()

# Plot
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(6, 6))
colors = ['#66c2a5', '#fc8d62']
deal_counts.plot(kind='pie', autopct='%1.1f%%', colors=colors, startangle=90,␣
 ↪wedgeprops=dict(width=0.4))
plt.title("Solo vs Multiple Shark Deals")
plt.ylabel('')
plt.tight_layout()
plt.show()
```



[85]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

```python
# Filter only funded deals with conditions
conditional_deals = df[(df['Deal Has Conditions'] == 'yes') & (df['Total Deal␣
↪Amount'] > 0)]

# Total number of conditional deals
total_conditional = len(conditional_deals)

# Shark-wise conditional deal counts
sharks = ['Namita', 'Vineeta', 'Anupam', 'Aman', 'Peyush', 'Ritesh', 'Amit']

shark_cond_counts = {}

for shark in sharks:
    shark_col = f"{shark} Investment Amount"
    count = conditional_deals[conditional_deals[shark_col] > 0].shape[0]
    shark_cond_counts[shark] = count

# Convert to DataFrame for plotting
shark_cond_df = pd.DataFrame.from_dict(shark_cond_counts, orient='index',␣
↪columns=['Conditional Deals'])
shark_cond_df = shark_cond_df.sort_values('Conditional Deals', ascending=False)

# Plot
plt.figure(figsize=(8, 5))
bars = plt.bar(shark_cond_df.index, shark_cond_df['Conditional Deals'],␣
↪color='#ffb347')
plt.title("Shark-wise Count of Conditional Deals")
plt.xlabel("Shark")
plt.ylabel("Number of Conditional Deals")

# Add count annotations
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height, int(height), ha='center',␣
↪va='bottom')

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

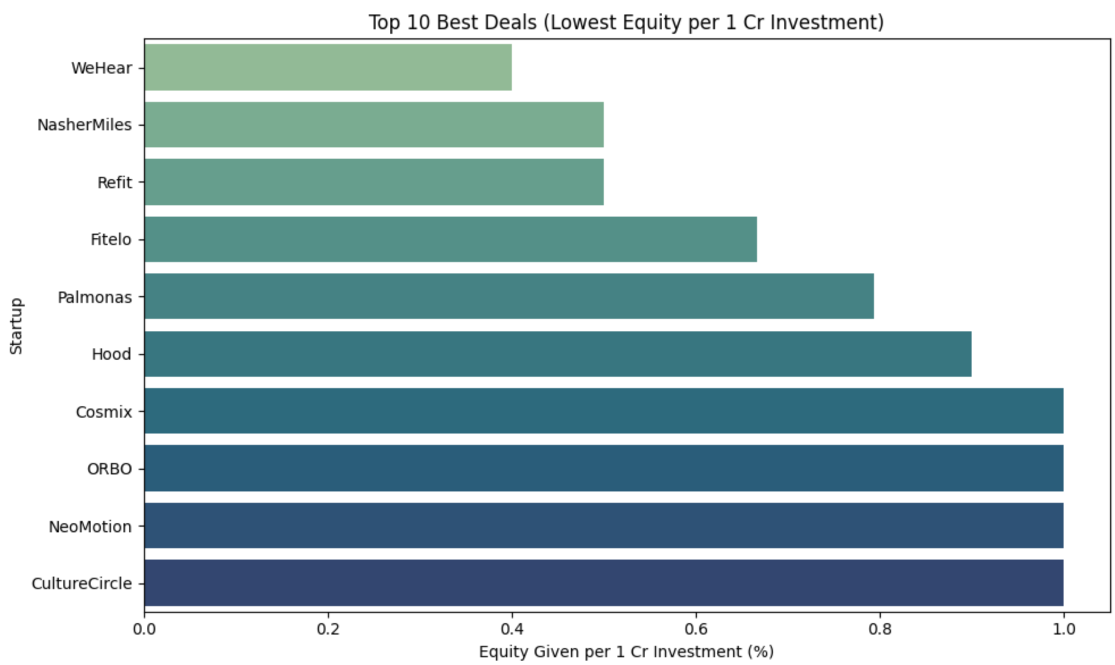


13 Deal Outcomes and Patterns

[86]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Make sure these columns exist and are numeric
df['Total Deal Amount'] = pd.to_numeric(df['Total Deal Amount'],␣
↪errors='coerce')
df['Total Deal Equity'] = pd.to_numeric(df['Total Deal Equity'],␣
↪errors='coerce')

# Create metric: equity given per 1 Cr investment
df['Equity per Crore'] = df['Total Deal Equity'] / (df['Total Deal Amount'] /␣
↪100)

# Filter out invalid values
best_deals = df[(df['Total Deal Amount'] > 0) & (df['Total Deal Equity'] > 0)]
best_deals = best_deals.sort_values(by='Equity per Crore').head(10)

# Plotting
plt.figure(figsize=(10, 6))
sns.barplot(
    data=best_deals,
```

```
    x='Equity per Crore',
    y='Startup Name',
    palette='crest'
)
plt.title('Top 10 Best Deals (Lowest Equity per 1 Cr Investment)')
plt.xlabel('Equity Given per 1 Cr Investment (%)')
plt.ylabel('Startup')
plt.tight_layout()
plt.show()
```

```
/var/folders/1v/g6bcxzl16s587g__qbw3d0z00000gp/T/ipykernel_55717/1345446805.py:1
8: FutureWarning:



Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.
```

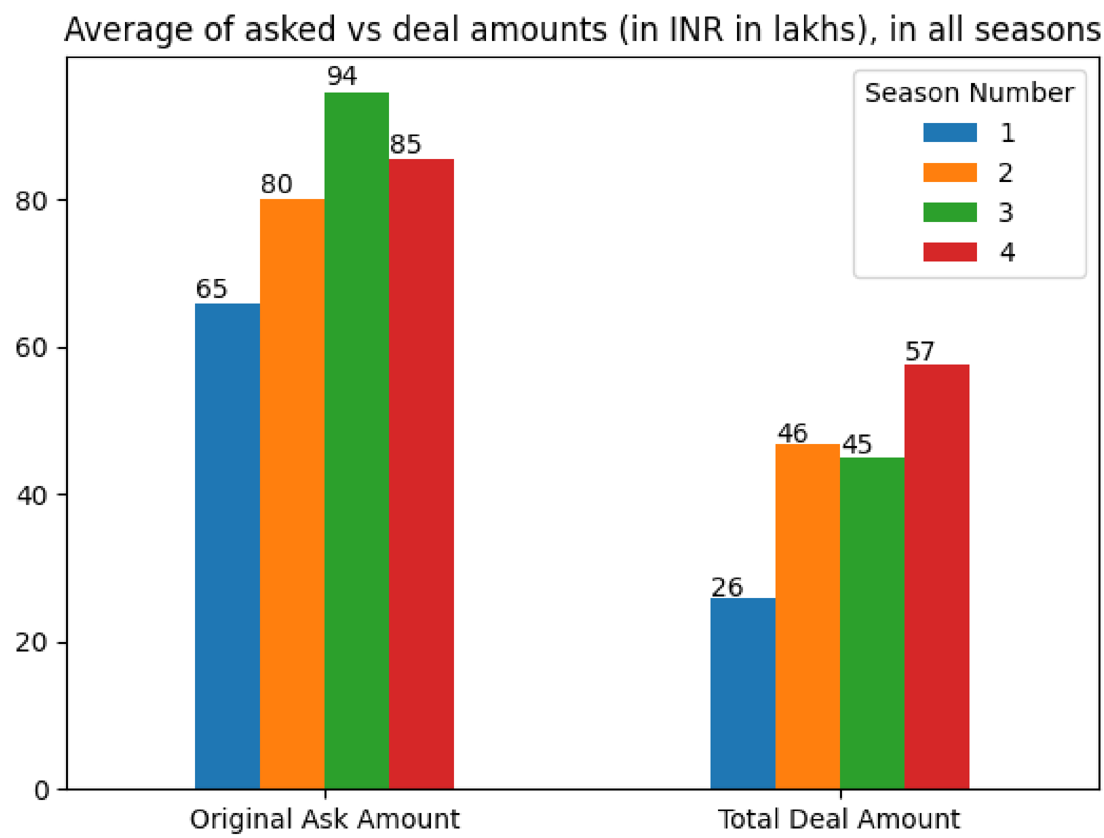


his bar chart visualizes the Top 10 startup deals from Shark Tank India that offered the **least amount of equity per 1 crore investment**, indicating the **best deals from an investor's point of view**

Key Observations: 1. **WeHear** gave away the least equity (~0.36%) for 1 Cr, indicating the highest valuation among the deals. 2. **NasherMiles** and **Refit** follow closely, also showing high investor-favorable terms. 3. Startups like **Fitelo**, **Palmonas**, and **Hood** gave away slightly more but still remained within a favorable range. 4. The bottom three startups — **Cosmix**, **ORBO**, and **NeoMotion** — gave away around 1% equity, still making the list of top 10 favorable deals.

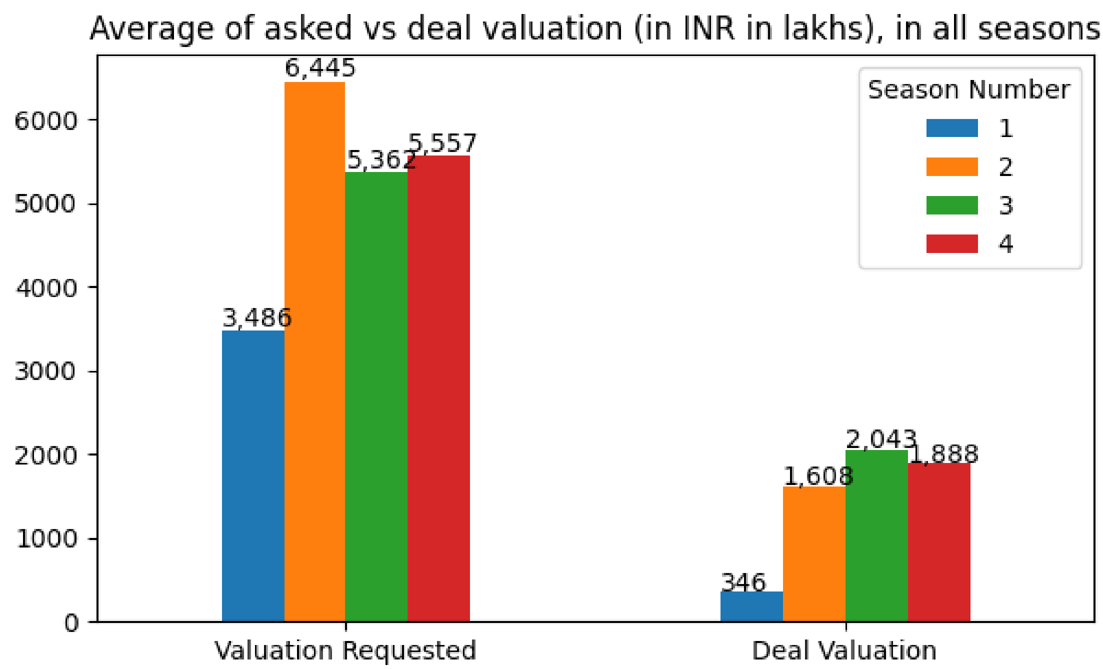
[87]:
```
# All seasons average of offered/deal amounts
ax = pd.pivot_table(df, values=['Original Ask Amount','Total Deal Amount'],␣
↪columns='Season Number', aggfunc=np.mean, sort=False).plot.
↪bar(figsize=(7,5), title="Average of asked vs deal amounts (in INR in␣
↪lakhs), in all seasons")
plt.xticks(rotation='horizontal')
for t in ax.patches:
    if (np.isnan(float(t.get_height()))):
        ax.annotate(0, (t.get_x(), 0))
    else:
        ax.annotate(str(format(int(t.get_height()), ',d')), (t.get_x(), t.
↪get_height()*1.01))
```

```
/var/folders/1v/g6bcxzl16s587g__qbw3d0z00000gp/T/ipykernel_55717/520255084.py:2:
FutureWarning:

The provided callable <function mean at 0x108333880> is currently using
DataFrameGroupBy.mean. In a future version of pandas, the provided callable will
be used directly. To keep current behavior pass the string "mean" instead.
```



[88]:
```
#All seasons average of offered/deal valuation
ax = pd.pivot_table(df, values=['Valuation Requested','Deal Valuation'],␣
↪columns='Season Number', aggfunc=np.mean, sort=False).plot.
↪bar(figsize=(7,4), title="Average of asked vs deal valuation (in INR in␣
↪lakhs), in all seasons")
plt.xticks(rotation='horizontal')
for t in ax.patches:
    if (np.isnan(float(t.get_height()))):
        ax.annotate(0, (t.get_x(), 0))
    else:
        ax.annotate(str(format(int(t.get_height()), ',d')), (t.get_x(), t.
↪get_height()*1.01))
```

```
/var/folders/1v/g6bcxzl16s587g__qbw3d0z00000gp/T/ipykernel_55717/4157631541.py:2
: FutureWarning:

The provided callable <function mean at 0x108333880> is currently using
DataFrameGroupBy.mean. In a future version of pandas, the provided callable will
be used directly. To keep current behavior pass the string "mean" instead.
```



[89]:
```
# Offers rejected by pitchers/startup companies
print(df[df['Accepted Offer']==0]["Startup Name"].count())
df.loc[df['Accepted Offer']==0, ["Season Number","Startup␣
↪Name","Industry","Original Ask Amount","Original Offered Equity"]]
```

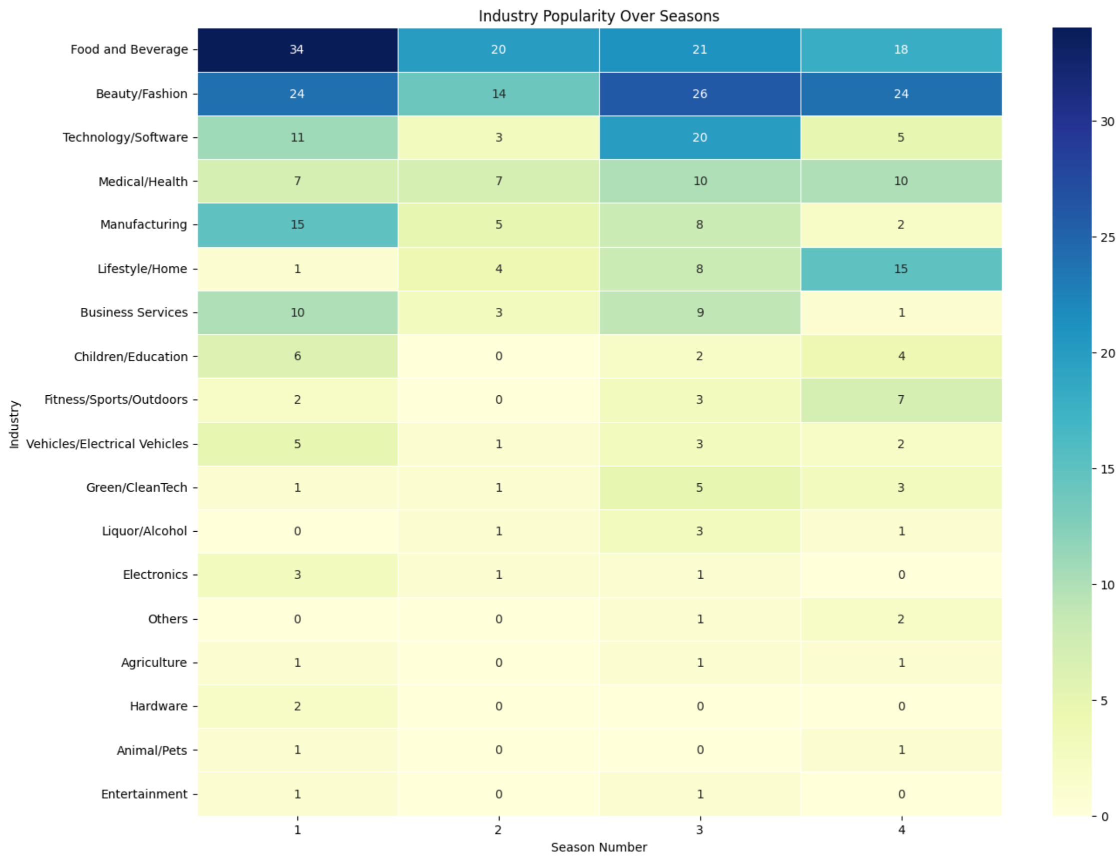
```
175
```

[89]:
```
     Season Number        Startup Name            Industry  \
6                1           qZenseLabs   Food and Beverage
14               1   ShrawaniEngineers       Beauty/Fashion
17               1               Hecoll      Beauty/Fashion
19               1             Torch-it  Children/Education
21               1          LaKheerDeli   Food and Beverage
..             …                   …                  …
624              4          UrbanAnimal          Animal/Pets
625              4                Nooky      Lifestyle/Home
626              4           Subculture      Beauty/Fashion
627              4             Woodsmen     Liquor/Alcohol
630              4             Rescript     Green/CleanTech

     Original Ask Amount  Original Offered Equity
6                  100.0                     0.25
14                  20.0                    10.00
17                 100.0                     1.00
19                  75.0                     1.00
```

```
21                   50.0                     7.50
..                    …                       …
624                  45.0                     5.00
625                  60.0                     1.00
626                  50.0                     7.00
627                 150.0                     0.50
630                 100.0                     3.33

[175 rows x 5 columns]
```

[97]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Group by Industry and Season Number
industry_season = df.groupby(['Industry', 'Season Number']).size().
 ↪unstack(fill_value=0)

# Sort industries by total pitches across seasons
industry_season = industry_season.loc[industry_season.sum(axis=1).
 ↪sort_values(ascending=False).index]

# Plot heatmap
plt.figure(figsize=(14, 10))
sns.heatmap(industry_season, annot=True, fmt='d', cmap='YlGnBu', linewidths=0.5)
plt.title("Industry Popularity Over Seasons")
plt.xlabel("Season Number")
plt.ylabel("Industry")
plt.tight_layout()
plt.show()
```

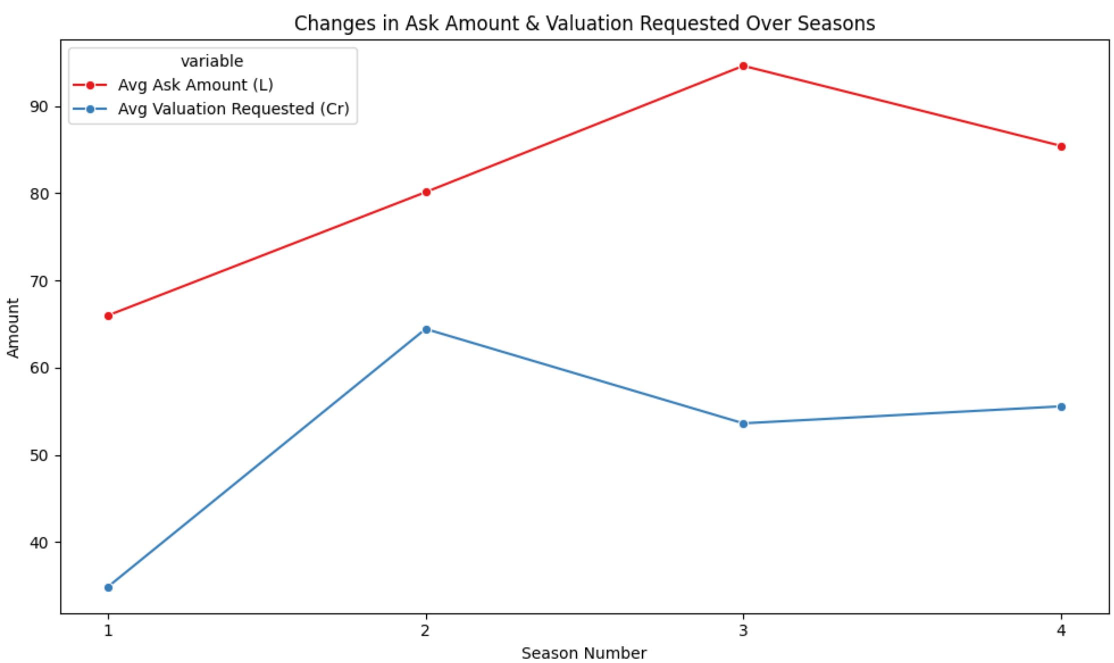


14 Insights:

The heatmap reveals an evolving startup ecosystem where traditional sectors like Food & Beverage are giving way to rising interest in tech, lifestyle, and sustainability domains. This reflects changing consumer behavior, innovation focus, and perhaps shifting shark preferences.

[100]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Group by Season Number
valuation_trends = df.groupby('Season Number').agg({
    'Original Ask Amount': 'mean',
    'Valuation Requested': 'mean'
}).reset_index()

# Rename for clarity
valuation_trends.columns = ['Season', 'Avg Ask Amount (L)', 'Avg Valuation␣
↪Requested (Cr)']

# Convert valuation to crores if needed
```

```python
valuation_trends['Avg Valuation Requested (Cr)'] = valuation_trends['Avg␣
↪Valuation Requested (Cr)'] / 100

# Ensure Season is integer
valuation_trends['Season'] = valuation_trends['Season'].astype(int)

# Melt for plotting
melted_valuation = valuation_trends.melt(id_vars='Season',
                                          value_vars=['Avg Ask Amount (L)', 'Avg␣
↪Valuation Requested (Cr)'])

# Plot
plt.figure(figsize=(10, 6))
sns.lineplot(data=melted_valuation, x='Season', y='value', hue='variable',␣
↪marker='o', palette='Set1')
plt.title("Changes in Ask Amount & Valuation Requested Over Seasons")
plt.xlabel("Season Number")
plt.ylabel("Amount")
plt.xticks(valuation_trends['Season'].unique())
#plt.grid(True)
plt.tight_layout()
plt.show()
```



14.1 Insights:

1. Startups have generally increased their funding expectations over time.
2. A divergence appears after Season 2: while ask amounts continued to rise until Season 3, valuations dropped, signaling a shift in investor sentiment or startup strategy.
3. Season 4 shows signs of more balanced behavior.

15 Conclusion

This analysis of Shark Tank India reveals clear patterns behind successful startup funding. Investors tend to favor startups in rapidly growing sectors like Fitness & Sports, those with strong financial metrics, and pitches that demonstrate innovation, clarity, and team diversity. While high competition exists in traditional categories like Food & Beverage, standout businesses still secure deals by differentiating themselves. The data also highlights individual investor tendencies and underscores the role of business fundamentals like revenue, gross margin, and valuation alignment in attracting offers.

16 Key insights

1. Fitness sector had the highest success rate (78.9%).
2. Food & Beauty sectors faced high competition, lower conversions.
3. Stronger financials (high margin/EBITDA) boosted funding chances.
4. Mixed/couple teams pitched more successfully.
5. Newer startups (post-2020) got funded more often.
6. Equity deals dominated over debt or hybrid offers.
7. Aman & Peyush invested widely; others had sector focus.

[124]:
```
df.to_csv('Shark_tank_analysis.csv', index=False)
```

[129]:
```
df.columns
```

[129]:
```
Index(['Season Number', 'Startup Name', 'Episode Number', 'Pitch Number',
       'Season Start', 'Season End', 'Original Air Date', 'Episode Title',
       'Anchor', 'Industry', 'Business Description', 'Started in',
       'Number of Presenters', 'Male Presenters', 'Female Presenters',
       'Transgender Presenters', 'Couple Presenters', 'Pitchers Average Age',
       'Pitchers City', 'Pitchers State', 'Yearly Revenue', 'Monthly Sales',
       'Gross Margin', 'Net Margin', 'EBITDA', 'Cash Burn', 'Has Patents',
       'Part of Match off', 'Original Ask Amount', 'Original Offered Equity',
       'Valuation Requested', 'Received Offer', 'Accepted Offer',
       'Total Deal Amount', 'Total Deal Equity', 'Total Deal Debt',
       'Debt Interest', 'Deal Valuation', 'Number of Sharks in Deal',
       'Deal Has Conditions', 'Namita Investment Amount',
       'Namita Investment Equity', 'Namita Debt Amount',
```

```
 'Vineeta Investment Amount', 'Vineeta Investment Equity',
 'Vineeta Debt Amount', 'Anupam Investment Amount',
 'Anupam Investment Equity', 'Anupam Debt Amount',
 'Aman Investment Amount', 'Aman Investment Equity', 'Aman Debt Amount',
 'Peyush Investment Amount', 'Peyush Investment Equity',
 'Peyush Debt Amount', 'Ritesh Investment Amount',
 'Ritesh Investment Equity', 'Ritesh Debt Amount',
 'Amit Investment Amount', 'Amit Investment Equity', 'Amit Debt Amount',
 'Guest Investment Amount', 'Guest Investment Equity',
 'Guest Debt Amount', 'All Guest Names', 'Namita Present',
 'Vineeta Present', 'Anupam Present', 'Aman Present', 'Peyush Present',
 'Ritesh Present', 'Amit Present', 'Guest Present', 'Got Funded',
 'Pitch Type', 'Equity per Crore'],
dtype='object')
```

[ ]:

```

```