Section	Description
	Dimension: 923 rows × 13 columns **Descriptive statistics:**

Data Overview

	Unnamed: 0	latitude	longitude	labels	age_first_funding_year	age_last_funding_year	age_first_milestone_year	age_last_milestone_year
count	923.000000	923.000000	923.000000	923.000000	923.000000	923.000000	771.000000	771.000000
mean	572.297941	38.517442	-103.539212	0.646804	2.235630	3.931456	3.055353	4.754423
std	333.585431	3.741497	22.394167	0.478222	2.510449	2.967910	2.977057	3.212107
min	1.000000	25.752358	-122.756956	0.000000	-9.046600	-9.046600	-14.169900	-7.005500
25%	283.500000	37.388869	-122.198732	0.000000	0.576700	1.669850	1.000000	2.411000
50%	577.000000	37.779281	-118.374037	1.000000	1.446600	3.528800	2.520500	4.476700
75%	866.500000	40.730646	-77.214731	1.000000	3.575350	5.560250	4.686300	6.753400
max	1153.000000	59.335232	18.057121	1.000000	21.895900	21.895900	24.684900	24.684900

Univariate Analysis	

Data Collection and Preprocessing Phase

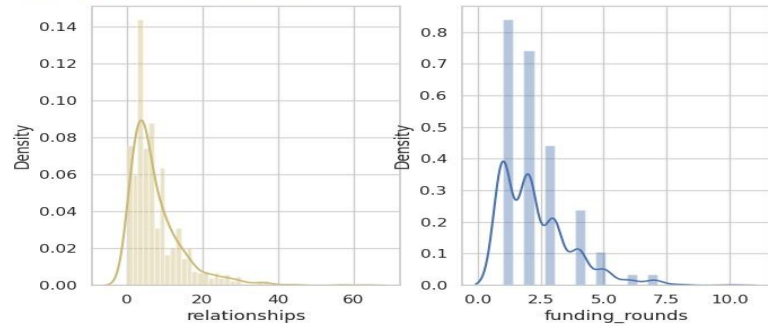| Date	21 June 2024
Team ID	TMID740029
Project Title	Startup Prophet
Maximum Marks	6 Marks

Data Exploration and Preprocessing Report

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will
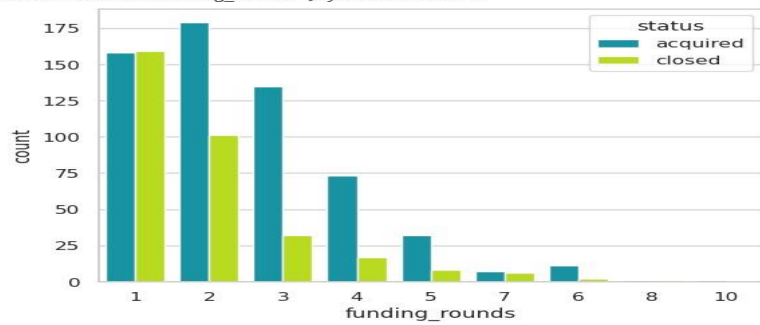address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

`sns.distplot(df['funding_rounds'])`


```
<Axes: xlabel='funding_rounds', ylabel='count'>
```
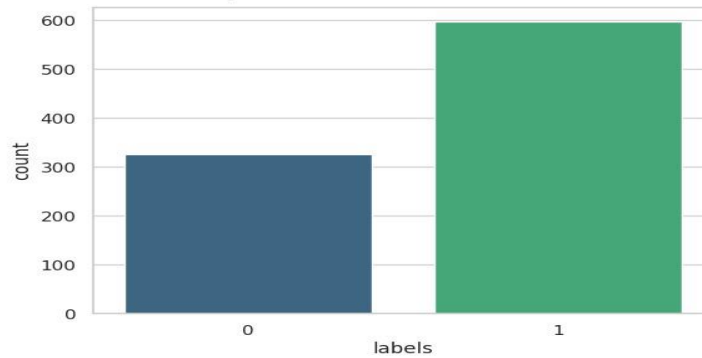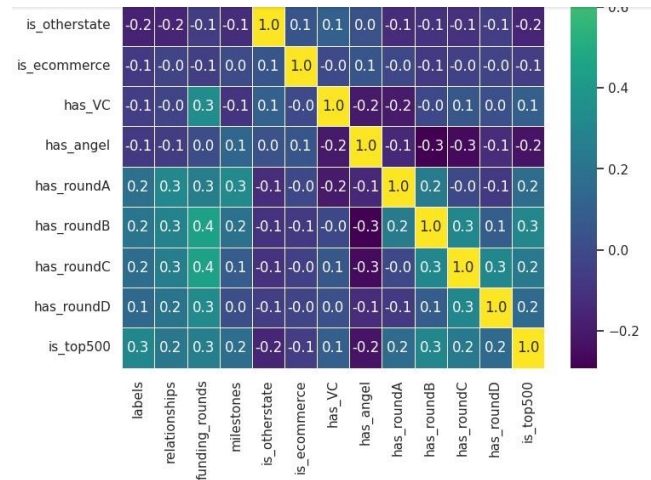


```
<ipython-input-16-8d78e83965e3>:2: FutureWarning:
Passing `palette` without assigning `hue` is deprecated and will be removed in
  sns.countplot(x=df['labels'],palette='viridis')
<Axes: xlabel='labels', ylabel='count'>
```


Bivariate Analysis

| Multivariate Analysis | |

| Outliers and Anomalies | - |

Data Preprocessing Code Screenshots

| | |
| Loading Data | |

Handling Missing Data	---

| Data Transformation | ```
[25]  #SEPARATING THE DATA
      x=df.drop(columns=['labels'],axis=1)
      y=df['labels']

      #STANDARD SCALAR
      from sklearn.preprocessing import StandardScaler
      sc=StandardScaler()
      x=sc.fit_transform(x)
      x

array([[-0.648696  ,  0.49566485,  0.87613768, ..., -0.55106471,
        -0.3327311 , -2.06017431],
       [ 0.17754099,  1.21500235, -0.6368185 , ...,  1.81466891,
         3.00542987,  0.48539582],
       [-0.37328367, -0.94301016,  0.11965959, ..., -0.55106471,
        -0.3327311 ,  0.48539582],
       ...,
       [-0.37328367, -0.94301016, -0.6368185 , ..., -0.55106471,
         3.00542987,  0.48539582],
       [ 0.59065949, -0.22367266,  0.11965959, ..., -0.55106471,
        -0.3327311 ,  0.48539582],
       [-0.51098983, -0.94301016, -0.6368185 , ..., -0.55106471,
``` |
| Feature Engineering | Attached the codes in final submission. |
| Save Processed Data | - |