interview

June 28, 2024

Germany startups between 2023-01-01 and 2024-06-31

[12]:
```
import pandas as pd

#
csv_path = "GE_startup_2023-2024_modified_date.csv"
df = pd.read_csv(csv_path)

pd.set_option('display.max_colwidth',80)  #

#
print("First 15 rows")
df.head(10)
```

```
First 15 rows
```

[12]:
```
              name  \
0           LiveEO
1     Sanity Group
2             Hive
3       Deepsafety
4    Forward Earth
5            Tilta
6  ISS VOLL GESUND
7             Mika
8     Lillian Care
9          Restate

tagline  \
0  LiveEO uses satellite data and machine learning algorithms to monitor
infras…
1  Unlocking the potential of cannabinoids to develop innovative
pharmaceutical…
2  Service, software and operations out of one hand helps you manage your
brand…
3  We're building AiDAR , a safe and certifiable Spatial AI Sensor to replace
L…
4
NaN
```

```
5  Building the technology that brings frictionless payment terms to B2B
market…
6
NaN
7                                             The AI co-pilot for small
businesses
8  Building a nationwide network of health facilities in Germany, aligning the
…
9                                       Without worrying about distributed
systems

              investors_name          address  \
0               MMC Ventures  Berlin, Germany
1                 OrganiGram  Berlin, Germany
2  Earlybird Venture Capital  Berlin, Germany
3                        NaN  Berlin, Germany
4                Speedinvest  Berlin, Germany
5         ff Venture Capital  Berlin, Germany
6         Sünderhauf Holding  Berlin, Germany
7     Samen Slimmer Alliance  Berlin, Germany
8         Bjoern von Siemens  Berlin, Germany
9          Redpoint Ventures  Berlin, Germany

                            industries  valuation   last_round  \
0                     energy oil & gas  €100-150m     SERIES B
1                health pharmaceutical  €150-226m  CONVERTIBLE
2  transportation logistics & delivery      €145m     SERIES A
3                    robotics mobility        NaN          NaN
4                  enterprise software     €8-12m         SEED
5          fintech mortgages & lending    €16-24m          NaN
6                                  NaN        NaN          NaN
7                                  NaN      €3-5m         SEED
8                               health      €6-9m         SEED
9                                  NaN    €28-42m         SEED

  last_round_amount    date
0              €25m  2024.6
1           €12.33m  2024.6
2            €18.2m  2024.6
3               NaN  2024.6
4               €2m  2024.6
5               NaN  2024.6
6               NaN  2024.6
7             €0.8m  2024.6
8             €1.5m  2024.6
9               €7m  2024.6
```

[9]:

```
total_startups = df['name'].nunique()
print(f"Total Number of Startups: {total_startups}")
num_companies_without_last_round_amount = df['last_round_amount'].isna().sum()
print(f"Number of Companies Without a Known Last Round Amount:␣
 ↪{num_companies_without_last_round_amount}")
# Convert the last_round_amount column to numerical values
last_round_amount_cleaned = df['last_round_amount'].str.extract(r'(\d+\.?\d*)').
 ↪astype(float)

# Calculate the total amount of the last round
total_last_round_amount = last_round_amount_cleaned.sum()[0]
print(f"Total Amount of the Last Round: €{total_last_round_amount:.2f} million")
```

```
Total Number of Startups: 468
Number of Companies Without a Known Last Round Amount: 168
Total Amount of the Last Round: €3473.33 million
```

0.0.1 Company Valuation Statistics

[33]:

```
valuation_cleaned = df['valuation'].str.extract(r'(\d+\.?\d*)').astype(float)
valuation_stats = valuation_cleaned.describe()
print("Company Valuation Statistics:")
print(valuation_stats)
```

```
Company Valuation Statistics:
                 0
count   404.000000
mean     93.185644
std     304.286935
min       0.000000
25%       5.000000
50%      14.500000
75%      40.250000
max    2250.000000
```

[38]:

```
valuation_cleaned = df['valuation'].str.extract(r'(\d+\.?\d*)').astype(float)
df['valuation_cleaned'] = valuation_cleaned
non_zero_valuations = df[df['valuation_cleaned'] > 0]


top_10_valuations = non_zero_valuations.sort_values(by='valuation_cleaned',␣
 ↪ascending=False).head(10)
print("Top 10 Valuations:")
print(top_10_valuations[['name', 'valuation_cleaned']])
```

```
last_10_valuations = non_zero_valuations.sort_values(by='valuation_cleaned',␣
 ↪ascending=True).head(10)
print("Last 10 Valuations without 0.0m:")
print(last_10_valuations[['name', 'valuation_cleaned']])
```

```
Top 10 Valuations:
             name  valuation_cleaned
540         Enpal             2250.0
146         Enpal             2250.0
469          TIER             2000.0
335  GetYourGuide             2000.0
542         Razor             1700.0
395         Razor             1700.0
81          Razor             1700.0
74        Solaris             1600.0
293       Solaris             1600.0
476         Choco             1200.0
Last 10 Valuations without 0.0m:
                    name  valuation_cleaned
416                Amply                1.0
418         randevu.tech                1.0
356  Bildungsurlauber.de                1.0
347               Waanda                1.0
390            Viva Maia                1.0
161         randevu.tech                1.0
383              Jobreel                1.0
413              heyroom                1.0
350             Converta                1.0
249             Heystack                1.0
```

[52]:

```
import plotly.express as px


# Drop rows with missing valuations for the valuation distribution plot
valuation_df = df.dropna(subset=['valuation'])

# Plotting Valuation Distribution using Plotly
fig = px.histogram(valuation_df, x='valuation', nbins=50, title='Valuation␣
 ↪Distribution',
                   labels={'valuation': 'Valuation (€m)'}, marginal="box",
                   color_discrete_sequence=['#1f77b4'])

fig.update_layout(
    xaxis_title='Valuation (€m)',
    yaxis_title='Frequency',
    title={
        'x':0.5,
        'xanchor': 'center'
    }
)

# Show the plot
fig.show()

# Save the plot as an HTML file
fig.write_html("Company valuation_distribution.html")
```

0.0.2 Distribution of Startups by Industry

[54]:

```
# Calculate the top 10 industries
top_industries = df['industries'].value_counts().head(10)

# Convert the top industries to a DataFrame for plotting
top_industries_df = top_industries.reset_index()
top_industries_df.columns = ['Industry', 'Count']
print("Top Industries")
print(top_industries_df)
```

```
Top Industries
                         Industry  Count
0             enterprise software     66
1          health health platform     34
2                         fintech     19
3             energy clean energy     13
4                       education     12
5                          health     10
6       fintech wealth management     10
7            food innovative food     10
8          fintech crypto and defi     10
9   food food logistics & delivery     10
```

[57]:

```
import pandas as pd
import plotly.express as px


# Distribution of Startups by Industry using Plotly
industry_counts = df['industries'].value_counts().reset_index()
industry_counts.columns = ['industry', 'count']

fig = px.bar(industry_counts, x='industry', y='count', title='Distribution of␣
 ↪Startups by Industry',
             labels={'industry': 'Industry', 'count': 'Number of Startups'},␣
 ↪color='count',
            color_continuous_scale='Viridis')

fig.update_layout(
    xaxis_title='Industry',
    yaxis_title='Number of Startups',
    title={
        'x':0.5,
        'xanchor': 'center'
    },
    xaxis_tickangle=-45
)

# Show the plot
fig.show()

# Save the plot as an HTML file
fig.write_html("industry_distribution.html")
```

[ ]: