

Company Layoffs

February 20, 2025

1 Company Layoffs Data Analysis

```
[103]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
plt.style.use('ggplot')
from matplotlib.pyplot import figure

%matplotlib inline
matplotlib.rcParams['figure.figsize'] = (15, 6) # resizing the plot
```

```
[104]: df = pd.read_csv(r"D:\Analyst materials\projects\files\layoffs_raw.csv")
df.head()
```

```
[104]:
```

	company	location	industry	total_laid_off	percentage_laid_off \
0	Atlassian	Sydney	Other	500.0	0.05
1	SiriusXM	New York City	Media	475.0	0.08
2	Alerzo	Ibadan	Retail	400.0	NaN
3	UpGrad	Mumbai	Education	120.0	NaN
4	Loft	Sao Paulo	Real Estate	340.0	0.15

	date	stage	country	funds_raised_millions
0	3/6/2023	Post-IPO	Australia	210.0
1	3/6/2023	Post-IPO	United States	525.0
2	3/6/2023	Series B	Nigeria	16.0
3	3/6/2023	Unknown	India	631.0
4	3/3/2023	Unknown	Brazil	788.0

```
[105]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2361 entries, 0 to 2360
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   company                2361 non-null   object
```

```

1  location                2361 non-null  object
2  industry                2357 non-null  object
3  total_laid_off          1621 non-null  float64
4  percentage_laid_off     1576 non-null  float64
5  date                   2360 non-null  object
6  stage                   2355 non-null  object
7  country                 2361 non-null  object
8  funds_raised_millions  2152 non-null  float64
dtypes: float64(3), object(6)
memory usage: 166.1+ KB

```

1.1 First, we do some data cleaning

1.1.1 Check and remove duplicates

```
[106]: df[df.duplicated()]
```

```

[106]:
      company  location  industry  total_laid_off  \
1492    Cazoo    London  Transportation      750.0
2357    Yahoo  SF Bay Area    Consumer     1600.0
2358    Hibob    Tel Aviv         HR         70.0
2359    Casper  New York City    Retail         NaN
2360  Wildlife Studios  Sao Paulo    Consumer      300.0

      percentage_laid_off  date  stage  country  \
1492                0.15  6/7/2022  Post-IP0  United Kingdom
2357                0.20  2/9/2023  Acquired  United States
2358                0.30  3/30/2020  Series A         Israel
2359                NaN  9/14/2021  Post-IP0  United States
2360                0.20  11/28/2022  Unknown         Brazil

      funds_raised_millions
1492                2000.0
2357                 6.0
2358                45.0
2359                339.0
2360                260.0

```

```
[107]: df = df.drop_duplicates()
```

1.1.2 Since I did the project in SQL first, I know there are some extra white spaces around company names, dots around country names, and the Crypto industry having multiple names. This is done by manually looking into the columns of the table in SQL

1.1.3 Stripping extra white spaces around company names

```
[108]: df.loc[:, 'company'] = df['company'].str.strip()
```

1.1.4 Stripping extra dots (.) around country names

```
[109]: df['country'] = df['country'].str.rstrip('.')
```

1.1.5 Collapsing all Crypto industries into Crypto

```
[110]: df.loc[df['industry'].str.startswith('Crypto', na=False), 'industry'] = 'Crypto'
```

1.1.6 Converting date data type to date

```
[111]: df['date'] = pd.to_datetime(df['date'])
```

1.1.7 Check null values

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

```
[112]: company                0
location                  0
industry                  4
total_laid_off            739
percentage_laid_off       784
date                      1
stage                     6
country                   0
funds_raised_millions     209
dtype: int64
```

1.1.8 Drop rows where both total_laid_off and percentage_laid_off are null (since the exploratory analysis will rely on these values)

```
[113]: df = df.dropna(subset=['total_laid_off', 'percentage_laid_off'], how='all')
```

1.1.9 Filling in industry if there are matching entries(same company and location) elsewhere with industry populated

```
[114]: pd.set_option('future.no_silent_downcasting', True)
df['industry'] = df.groupby(['company', 'location'])['industry'].
    ↪transform(lambda x: x.ffill())
```

```
df['industry'] = df.groupby(['company', 'location'])['industry'].
    ↪transform(lambda x: x.bfill())
```

1.1.10 Check for remaining null industries

```
[115]: df[df['industry'].isna()]
```

```
[115]:
```

	company	location	industry	total_laid_off	\
330	Bally's Interactive	Providence	NaN	NaN	

	percentage_laid_off	date	stage	country	\
330	0.15	2023-01-18	Post-IPO	United States	

	funds_raised_millions
330	946.0

1.1.11 There is still 1 company with null industry, we can fill it in

```
[116]: df['industry'] = df['industry'].fillna('Other')
```

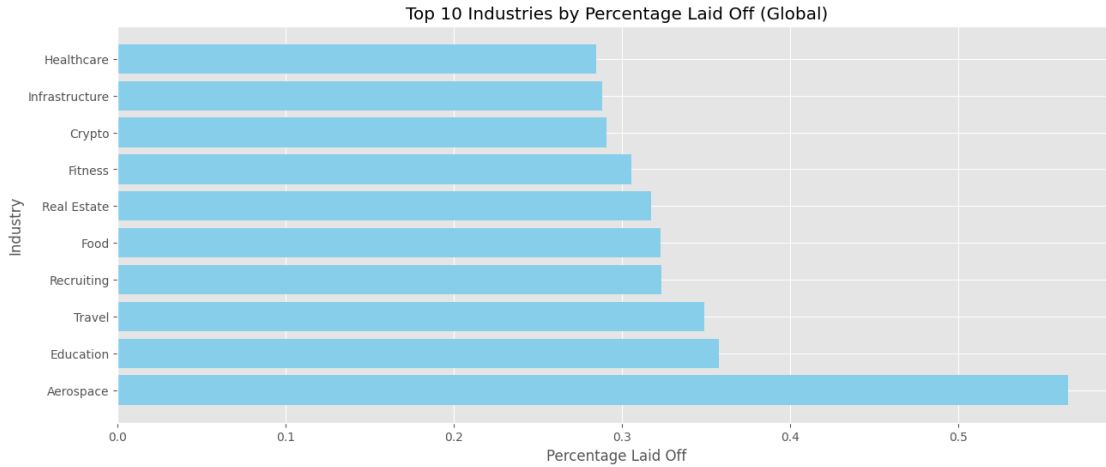
1.2 Now, we perform Exploratory Data Analysis with an emphasis on US impact

1.2.1 Top 10 industries by percentage laid off Globally

```
[117]: global_top10 = df.groupby('industry')['percentage_laid_off'].mean().
    ↪nlargest(10).reset_index()

plt.barh(global_top10['industry'], global_top10['percentage_laid_off'],
    ↪color='skyblue')
plt.xlabel('Percentage Laid Off')
plt.ylabel('Industry')
plt.title('Top 10 Industries by Percentage Laid Off (Global)')

plt.show()
```

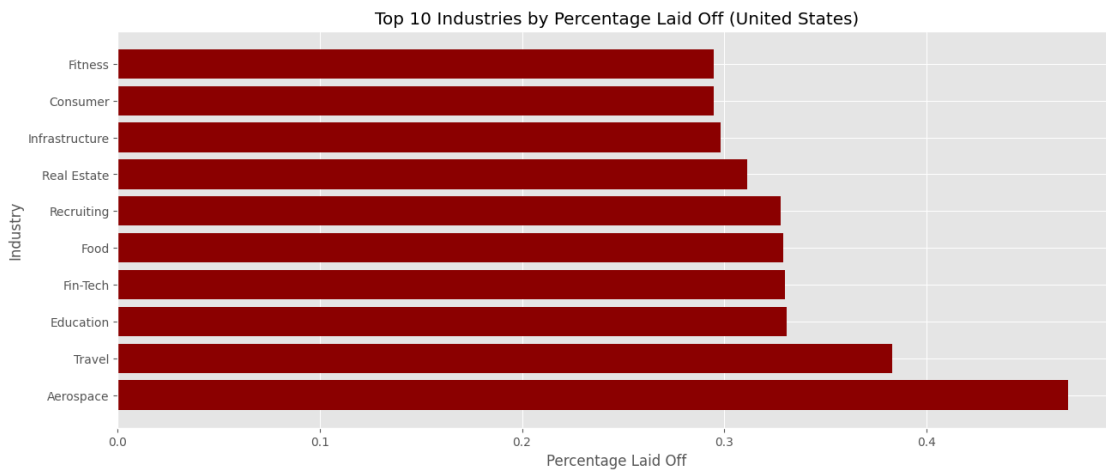


1.2.2 Top 10 industries by percentage laid off in United States

```
[118]: us_top10 = (df[df['country'] == 'United States'].
↳groupby('industry')['percentage_laid_off'].mean().nlargest(10).reset_index())

plt.barh(us_top10['industry'], us_top10['percentage_laid_off'], color='darkred')
plt.xlabel('Percentage Laid Off')
plt.ylabel('Industry')
plt.title('Top 10 Industries by Percentage Laid Off (United States)')

plt.show()
```



1.2.3 Top 10 industries with the largest gap in percentage laid off Global vs United States

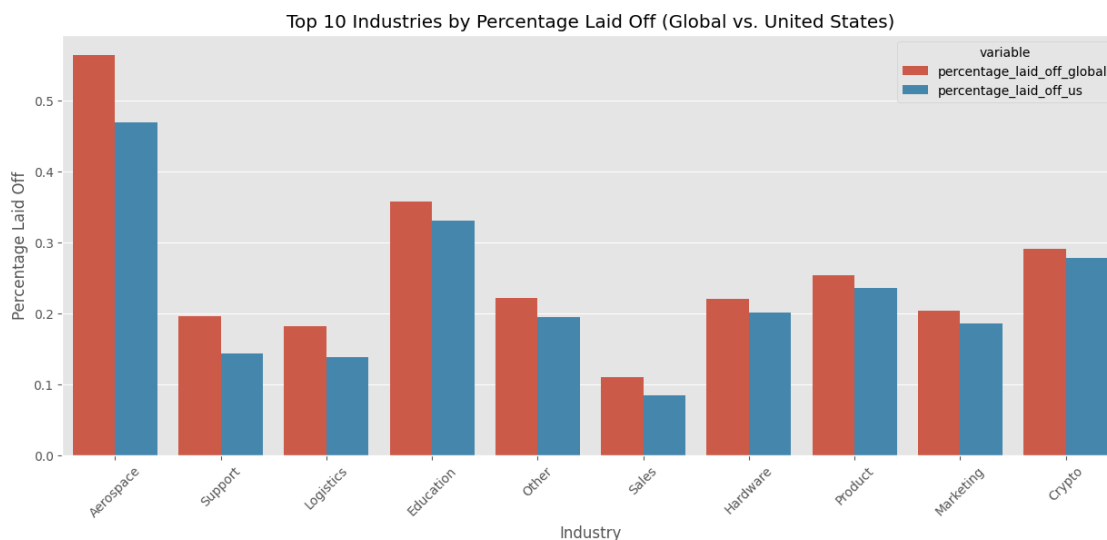
```
[119]: # getting percentage laid off for global and US
global_avg = df.groupby('industry')['percentage_laid_off'].mean().reset_index()
us_avg = df[df['country'] == 'United States'].
    ↪groupby('industry')['percentage_laid_off'].mean().reset_index()

# getting the top 10 percentage gaps between global and US
global_avg.rename(columns={'percentage_laid_off': 'percentage_laid_off_global'}, inplace=True)
us_avg.rename(columns={'percentage_laid_off': 'percentage_laid_off_us'}, inplace=True)
compare = pd.merge(global_avg, us_avg, on='industry', how='outer')
compare['gap'] = compare['percentage_laid_off_global'] -
    ↪compare['percentage_laid_off_us']
top_10_gap = compare.nlargest(10, 'gap').reset_index(drop=True)

sns.barplot(data=top_10_gap.melt(id_vars='industry',
    ↪value_vars=['percentage_laid_off_global', 'percentage_laid_off_us']),
    x='industry', y='value', hue='variable')

plt.title('Top 10 Industries by Percentage Laid Off (Global vs. United States)')
plt.xlabel('Industry')
plt.ylabel('Percentage Laid Off')
plt.xticks(rotation=45)

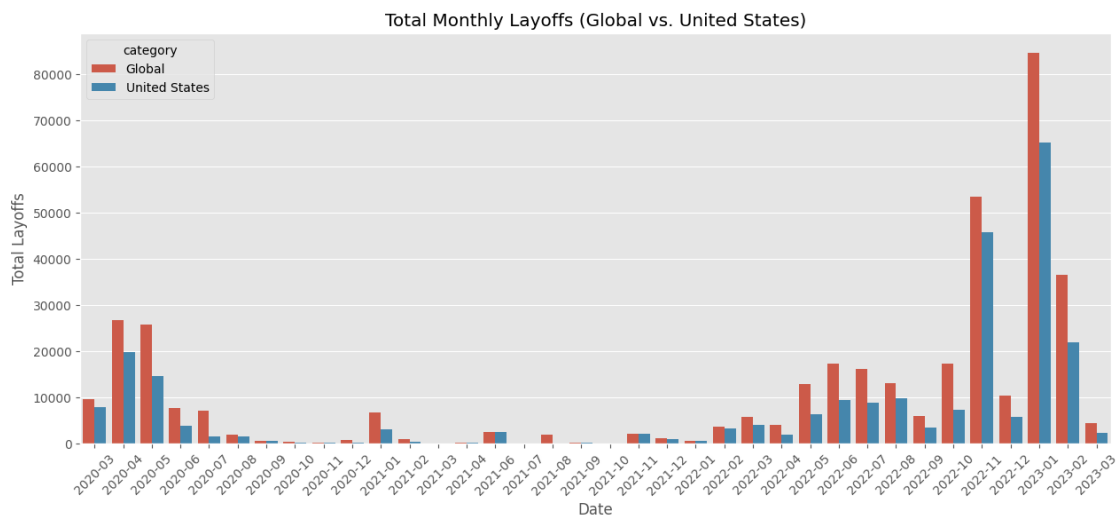
plt.show()
```



1.2.4 Total monthly United States vs Global layoffs

```
[123]: df['month_year'] = df['date'].dt.to_period('M')
global_layoffs = df.groupby('month_year')['total_laid_off'].sum().reset_index()
global_layoffs['category'] = 'Global'
us_layoffs = df[df['country'] == 'United States'].
    ↳groupby('month_year')['total_laid_off'].sum().reset_index()
us_layoffs['category'] = 'United States'
layoffs_combined = pd.concat([global_layoffs, us_layoffs])

sns.barplot(data=layoffs_combined, x='month_year', y='total_laid_off',
    ↳hue='category')
plt.title('Total Monthly Layoffs (Global vs. United States)')
plt.xlabel('Date')
plt.ylabel('Total Layoffs')
plt.xticks(rotation=45)
plt.show()
```

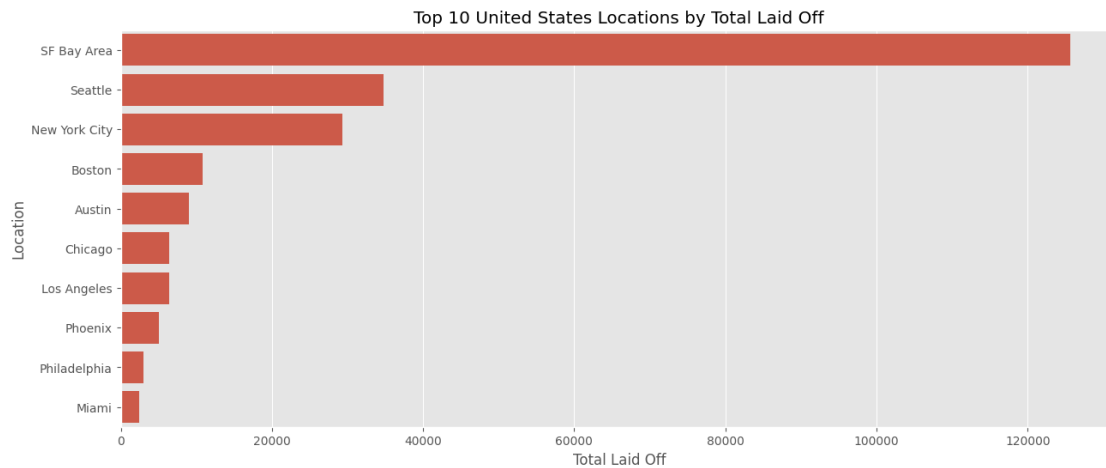


1.2.5 Top 10 United States Locations by Total Laid Off

```
[121]: us_grouped = df[df['country'] == 'United States'].
    ↳groupby('location')['total_laid_off'].sum().reset_index()
top_10_us_locations = us_grouped.sort_values(by='total_laid_off',
    ↳ascending=False).head(10)

sns.barplot(x='total_laid_off', y='location', data=top_10_us_locations)
plt.title('Top 10 United States Locations by Total Laid Off')
plt.xlabel('Total Laid Off')
plt.ylabel('Location')
```

```
plt.show()
```



1.2.6 Top 10 United States Companies by Total Laid Off

```
[122]: us_data = df[df['country'] == 'United States']

top_10_us_companies = us_data.groupby('company')['total_laid_off'].sum().
    ↪reset_index()
top_10_us_companies = top_10_us_companies.sort_values(by='total_laid_off',
    ↪ascending=False).head(10)

sns.barplot(x='total_laid_off', y='company', data=top_10_us_companies,
    ↪color='blue')
plt.title('Top 10 United States Companies by Total Laid Off')
plt.xlabel('Total Laid Off')
plt.ylabel('Company')

plt.show()
```