

LAB 01: LAYOFFS 2023

Bảng nội dung

- LAB 01: LAYOFFS 2023
 - Thông tin người thực hiện
 - Mô tả đồ án:
 - A. Thu thập dữ liệu
 - Ngữ cảnh thực hiện tìm kiếm dữ liệu
 - Chủ đề
 - Nguồn
 - Người ta có cho phép sử dụng dữ liệu như thế này hay không?
 - Người ta đã thu thập dữ liệu này như thế nào? Phương pháp thực hiện là gì?
 - Đọc dữ liệu
 - B. Khám phá dữ liệu
 - Mỗi dòng có ý nghĩa gì? Có vấn đề các dòng có ý nghĩa khác nhau không?
 - Mỗi cột có ý nghĩa gì? Thông tin của mỗi cột ? Mỗi cột hiện đang có kiểu dữ liệu gì?
 - Phân bố dữ liệu ở các cột
 - Với mỗi cột có kiểu dữ liệu dạng numeric, các giá trị được phân bố như thế nào?
 - Với mỗi cột không có kiểu dữ liệu dạng categorical, các giá trị được phân bố như thế nào?
 - Tiền xử lý dữ liệu
 - C. Khám phá mối quan hệ trong dữ liệu
 - Tìm hiểu mối liên hệ giữa Stage và Percentage_laid_off:

- Xét từng quốc gia, thể hiện bằng biểu đồ cột cho thấy số lượng người bị sa thải của các ngành.
- Ngành nào đang có số lượng layoff cao nhất trên thế giới
- Xét tất cả các ngành, dùng biểu đồ cột thể hiện số lượng quốc gia có ngành nghề đó layoff top đầu / cuối.
- Khám phá dữ liệu thực tế
 - Địa điểm tiêu biểu nơi xảy ra layoff
 - Số lượng layoff theo thời gian
 - Số lượng layoff theo tháng
 - Liệu có đợt layoff này diễn ra vào holiday week?
 - Tình hình layoff ở Việt Nam
 - Với mỗi đợt sa thải, các công ty thường cho nghỉ bao nhiêu phần trăm nhân sự?
 - Top 20 công ty có tổng % số lượng nhân sự bị sa thải nhiều nhất
 - Tìm hiểu về các công ty sa thải hơn 50% nhân sự trong các đợt
 - Tìm hiểu quan hệ giữa
total_laid_off,percentage_laid_off,funds_raised,total_employees

Thông tin người thực hiện

MSSV	Họ và tên
20120095	Ngô Quang Hưng
20120519	Nguyễn Thị Thúy Liễu
20120540	Võ Hoàng Thảo Nguyên
20120633	Viên Hải Yến
20120634	Lê Minh Trí

Mô tả đồ án:

- Từ nguồn dữ liệu lấy được từ Kaggle về vấn đề sa thải nhân viên ở các công ty trên toàn thế giới, đồ án này nhằm mục đích phân tích dữ liệu đã có để có được cái nhìn khách quan hơn về vấn đề này.
-

Import các thư viện

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from textwrap import wrap
from wordcloud import WordCloud
import plotly.express as px
import matplotlib.colors as colors
import warnings
```

A. Thu thập dữ liệu

Ngữ cảnh thực hiện tìm kiếm dữ liệu

Sau dịch covid 19, các công ty công nghệ trên toàn cầu đang đối mặt với sự suy thoái kinh tế. Sự chậm lại của chi tiêu từ người tiêu dùng, lãi suất cao hơn từ các ngân hàng trung ương và sức mạnh của đồng đô la ngoại tệ đang gợi ý về khả năng xảy ra suy thoái kinh tế và các công ty công nghệ đã bắt đầu **cắt giảm nhân sự**.

Chủ đề

Phân tích tình trạng hỗn loạn công nghệ gần đây thông qua số liệu về việc sa thải nhân sự, từ đó rút ra được những thông tin hữu ích.

Nguồn

Dữ liệu được lấy trên Kaggle dưới file csv, đường link dẫn đến [Ở đây](#)

Người ta có cho phép sử dụng dữ liệu như thế này hay không?

Thông tin License của dữ liệu: Open Database.

Theo thông tin License của dữ liệu, người dùng được phép tự do chia sẻ, sửa đổi và sử dụng dữ liệu trong khi duy trì sự tự do tương tự cho người dùng khác.

Người ta đã thu thập dữ liệu này như thế nào? Phương pháp thực hiện là gì?

Dữ liệu được thu thập bằng cách theo dõi số liệu được báo cáo trên các nền tảng:

- Bloomberg
- San Francisco Business Times

- TechCrunch
- The New York Times

Phương pháp thu thập dữ liệu: Dữ liệu được tải xuống từ Layoffs.fyi Tracker và được biên soạn thành 1 file CSV.

Đọc dữ liệu

```
df=pd.read_csv('./datasets/layoffs.csv')
df.head()
```

	company	location	industry	total_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
0	2023-03-06	Post-IPO	Australia	210.0
1	2023-03-06	Post-IPO	United States	525.0
2	2023-03-06	Series B	Nigeria	16.0
3	2023-03-06	Unknown	India	631.0
4	2023-03-03	Unknown	Brazil	788.0

B. Khám phá dữ liệu

Mỗi dòng có ý nghĩa gì? Có vấn đề các dòng có ý nghĩa khác nhau không?

- Mỗi dòng chứa dữ liệu về một công ty và các thông tin liên quan đến công ty đó như: tên công ty, nơi đặt trụ sở chính, loại ngành nghề, số nhân viên bị sa thải, tỷ lệ sa thải, ngày công bố, giai đoạn của công ty, quốc gia công ty hoạt động, số vốn huy động bởi công ty.
- ví dụ: dòng 1: doanh nghiệp SiriusXM có trụ sở tại New York City và hoạt động trong ngành truyền thông, đã sa thải 475 nhân viên, chiếm 8% tổng số nhân viên của họ tính tới ngày 6 tháng 3 năm 2023. Công ty cũng đang ở giai đoạn Post-IPO và có trụ sở tại Hoa Kỳ. Họ đã huy động được số vốn 525 triệu đô la.
- Không có tình trạng các dòng có ý nghĩa khác nhau.

Mỗi cột có ý nghĩa gì?

```
warnings.filterwarnings('ignore')
```

```

column=2
print("column ", column,": ", df.columns[column])
print("size: ", len(df.loc[:, df.columns[column]]))
print("number of values: ", len(df.loc[:,
df.columns[column]].unique()))

arr= df.loc[:, df.columns[column]].unique()

nanIndex=-1
for index in range(len(arr)):
    if isinstance(arr[index], float) and np.isnan(arr[index]):
        nanIndex=index
        break
if nanIndex != -1:
    arr=np.delete(arr,nanIndex)

print("number of values (without nan): ", len(arr))

print("data (without nan) : ", ', '.join(arr))

```

```

column 2 : industry
size: 2357
number of values: 30
number of values (without nan): 29
data (without nan) : Other, Media, Retail, Education, Real Estate,
Transportation, Marketing, Travel, Healthcare, Security, Food,
Fitness, Consumer, Logistics, HR, Support, Crypto, Finance, Data,
Sales, Infrastructure, Hardware, Product, Construction, Legal, Energy,
Manufacturing, Recruiting, Aerospace

```

Thông tin của mỗi cột:

- Company: Tên của công ty
 - Kích thước: 2357
 - Số lượng dữ liệu không trùng lặp: 1893
 - Một số Company có trong dữ liệu: Atlassian, SiriusXM, Alerzo, UpGrad, Loft, Embark Trucks,...
- Location: Nơi đặt trụ sở chính của công ty
 - Kích thước: 2357
 - Số lượng dữ liệu không trùng lặp: 191
 - Một số Location có trong dữ liệu: Sydney, New York City, Ibadan, Mumbai, Sao Paulo,...
- Industry: Loại ngành mà công ty hoạt động
 - Kích thước: 2357

- Số lượng dữ liệu không trùng lặp: 30
- Một số Industry có trong dữ liệu: Other, Media, Retail, Education, Real Estate, Transportation,...
- Total_laid_off: Tổng số nhân viên bị sa thải bởi công ty
 - Kích thước: 2357
 - Số lượng dữ liệu không trùng lặp: 286
 - Một số Total_laid_off có trong dữ liệu: 475.0, 400.0, 120.0, 340.0, 230.0, 100.0, 63.0, 30.0...
- Percentage_laid_off: Tỷ lệ nhân viên bị sa thải bởi công ty
 - Kích thước: 2357
 - Số lượng dữ liệu không trùng lặp: 76
 - Một số Percentage_laid_off có trong dữ liệu: 0.05, 0.08, 0.15, 0.7, 0.03, 0.11, 0.1, 1.0...
- Date: Ngày công bố thông tin về việc sa thải
 - Kích thước: 2357
 - Số lượng dữ liệu không trùng lặp: 191
 - Một số Date có trong dữ liệu: 2023-03-06, 2023-03-03, 2023-03-02, 2023-03-01, 2023-02-28,...
- Stage: Giai đoạn của công ty
 - Kích thước: 2357
 - Số lượng dữ liệu không trùng lặp: 17
 - Một số Stage có trong dữ liệu: Post-IPO, Series B, Unknown, Acquired, Series F, Series E...
- Country: Quốc gia nơi công ty đặt trụ sở
 - Kích thước: 2357
 - Số lượng dữ liệu không trùng lặp: 59
 - Một số Country có trong dữ liệu: Australia, United States, Nigeria, India, Brazil,...
- Funds_raised: Tổng số vốn do công ty huy động (tính bằng triệu đô la)
 - Kích thước: 2357
 - Số lượng dữ liệu không trùng lặp: 639
 - Một số Funds_raised có trong dữ liệu: 525.0, 16.0, 631.0, 788.0, 317.0, 59.0, 152.0...

Mỗi cột hiện đang có kiểu dữ liệu gì?

df.dtypes

company	object
location	object
industry	object
total_laid_off	float64
percentage_laid_off	float64
date	object
stage	object

```
country          object
funds_raised     float64
dtype: object
```

```
object_columns = df.select_dtypes(include=['object']).columns
```

```
for col in object_columns:
    print(col, ":", set(df[col].apply(type)))
```

```
company : {<class 'str'>}
location : {<class 'str'>}
industry : {<class 'str'>, <class 'float'>}
date : {<class 'str'>, <class 'float'>}
stage : {<class 'str'>, <class 'float'>}
country : {<class 'str'>}
```

Chúng ta nhận thấy rằng ba cột industry, date và stage có dạng dữ liệu không đồng nhất, tuy nhiên đây chỉ là do sự thiếu hụt dữ liệu trong các dòng, được biểu thị bằng giá trị NaN (dạng số thực).

Có cột nào có kiểu dữ liệu chưa phù hợp để có thể xử lý tiếp hay không?

Các cột sau đang có kiểu dữ liệu là object, do đó cần được chuyển đổi sang kiểu chuỗi (string):

- company
- location
- industry
- stage
- country

Trong khi đó, cột date đang có kiểu dữ liệu là object, cần được chuyển đổi sang kiểu ngày tháng (datetime).

Phân bố dữ liệu ở các cột

Với mỗi cột có kiểu dữ liệu dạng numeric, các giá trị được phân bố như thế nào?

```
numerical_des = df.describe()
missing_ratio = pd.DataFrame({col : df[col].isnull().sum() / len(df)
    for col in numerical_des.columns}, index=["missing_ratio"])
pd.concat([numerical_des, missing_ratio])
```

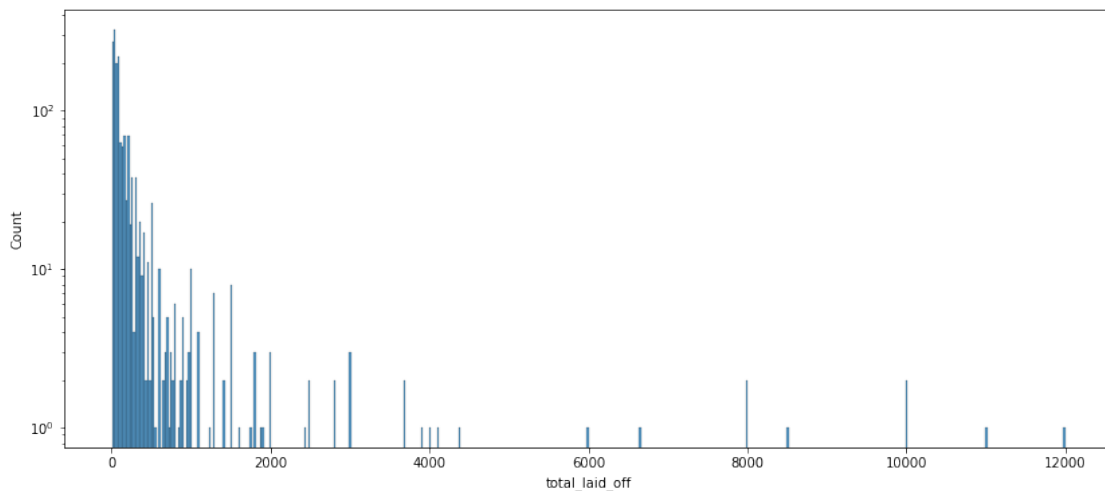
	total_laid_off	percentage_laid_off	funds_raised
count	1618.000000	1573.000000	2148.000000
mean	237.582818	0.257964	823.683098
std	769.674697	0.255808	5599.094277
min	3.000000	0.000000	0.000000
25%	36.000000	0.100000	50.000000
50%	80.000000	0.170000	156.500000
75%	180.000000	0.300000	445.000000

max	12000.000000	1.000000	121900.000000
missing_ratio	0.313534	0.332626	0.088672

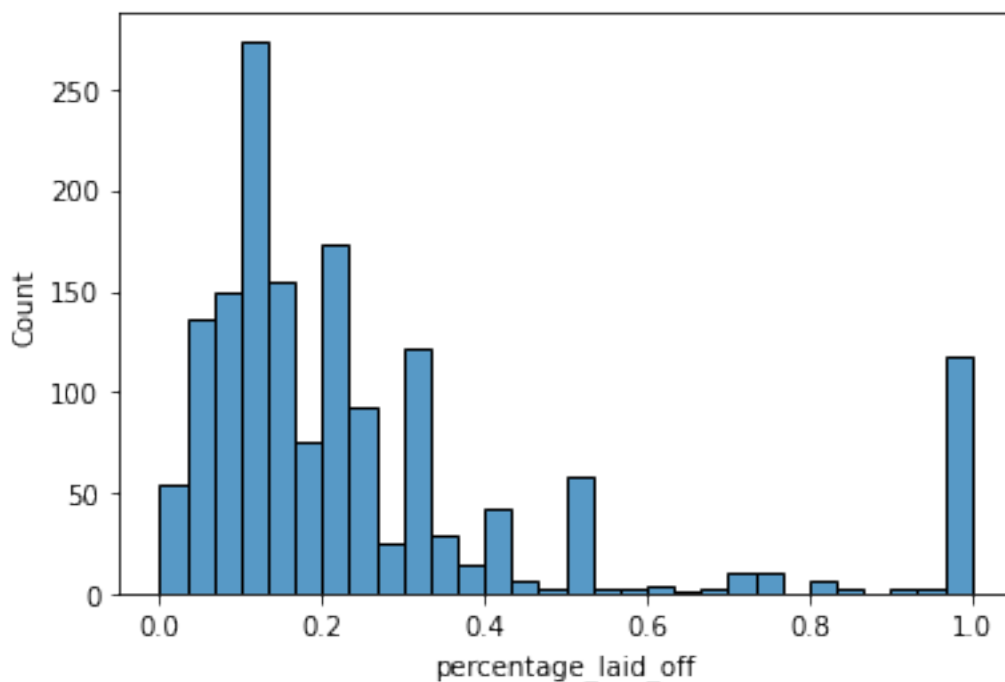
Từ bảng dữ liệu phân tích của kiểu dữ liệu numeric, tồn tại những trường dữ liệu với tỉ lệ thiếu lên tới hơn 30%.

Total Laid Off

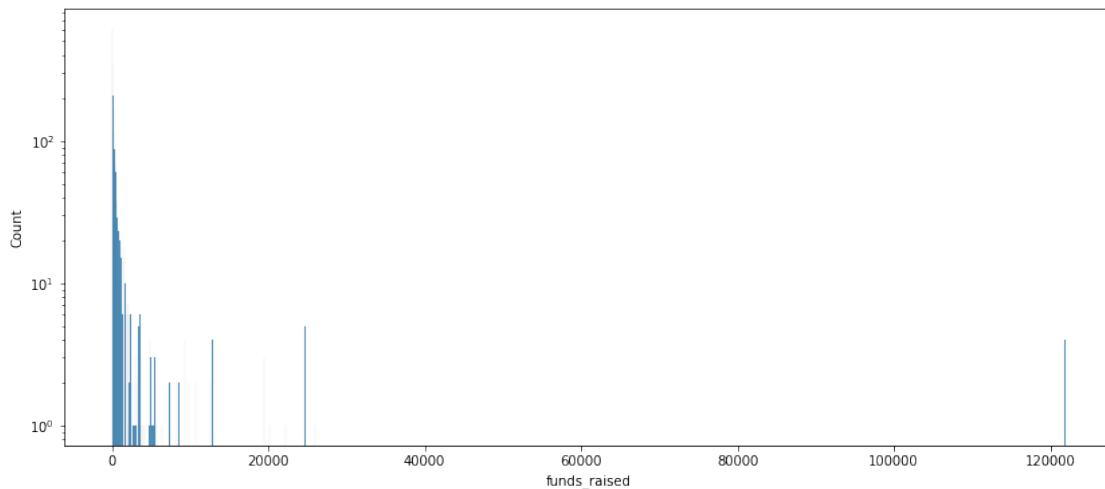
```
plt.figure(figsize=(14,6))
ax= sns.histplot(data=df, x='total_laid_off')
ax.set_yscale('log')
plt.show()
```



```
sns.histplot(data=df, x='percentage_laid_off')
plt.show()
```




```
plt.figure(figsize=(14,6))
ax=sns.histplot(data=df, x='funds_raised')
ax.set_yscale('log')
plt.show()
```



Với mỗi cột không có kiểu dữ liệu dạng categorical, các giá trị được phân bố như thế nào? ↑

```
cate_cols=['company', 'location', 'industry', 'stage', 'country']
cate_rows=["missing_ratio","num_diff_vals","diff_vals"]
data_cate={}
for col in cate_cols:
    row=[]
    row.append(round(df[col].isnull().sum()/len(df[col])*100,3))
    data_cate.update({f'{col}':row})

    row.append(len(df[col][df[col].isna()==False].unique()))
    row.append(np.array(df[col][df[col].isna()==False].unique()))
    data_cate.update({f'{col}':row})

cate_col_df=pd.DataFrame(data_cate,index=cate_rows)
display(cate_col_df)
```

	company \
missing_ratio	0.0
num_diff_vals	1893
diff_vals	[Atlassian, SiriusXM, Alerzo, UpGrad, Loft, Em...
	location \
missing_ratio	0.0
num_diff_vals	191
diff_vals	[Sydney, New York City, Ibadan, Mumbai, Sao Pa...
	industry \
missing_ratio	0.042
num_diff_vals	29
diff_vals	[Other, Media, Retail, Education, Real Estate,...

```

missing_ratio      stage \
num_diff_vals      0.255
diff_vals          16
diff_vals          [Post-IP0, Series B, Unknown, Acquired, Series...

```

```

missing_ratio      country
num_diff_vals      0.0
diff_vals          59
diff_vals          [Australia, United States, Nigeria, India, Bra...

```

Có cần phải tiền xử lý dữ liệu hay không và nếu có thì nhóm sinh viên cần phải xử lý như thế nào?

```
df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2357 entries, 0 to 2356
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   company               2357 non-null   object
 1   location              2357 non-null   object
 2   industry              2356 non-null   object
 3   total_laid_off        1618 non-null   float64
 4   percentage_laid_off   1573 non-null   float64
 5   date                  2356 non-null   object
 6   stage                 2351 non-null   object
 7   country               2357 non-null   object
 8   funds_raised          2148 non-null   float64
dtypes: float64(3), object(6)
memory usage: 165.9+ KB

```

Quan sát thấy có số lượng lớn dữ liệu bị thiếu và có những cột phải xử lý kiểu dữ liệu do vậy ta phải tiền xử lý dữ liệu.

Tiền xử lý dữ liệu

Kiểm tra missing data

```
have_Nan=df.columns[df.isna().any()]
```

Đếm số missing data của các cột có missing data

```
print("Missing data:")
```

```
print(df[have_Nan].isnull().sum())
```

```
df.isna().sum().plot(kind="barh",title='Missing data')
```

```
plt.show()
```

Phân trăm dữ liệu bị thiếu của những cột này

```
print("Percentage missing data:")
```

```
percentage_missing_data=df[have_Nan].isnull().sum() * 100 /
```

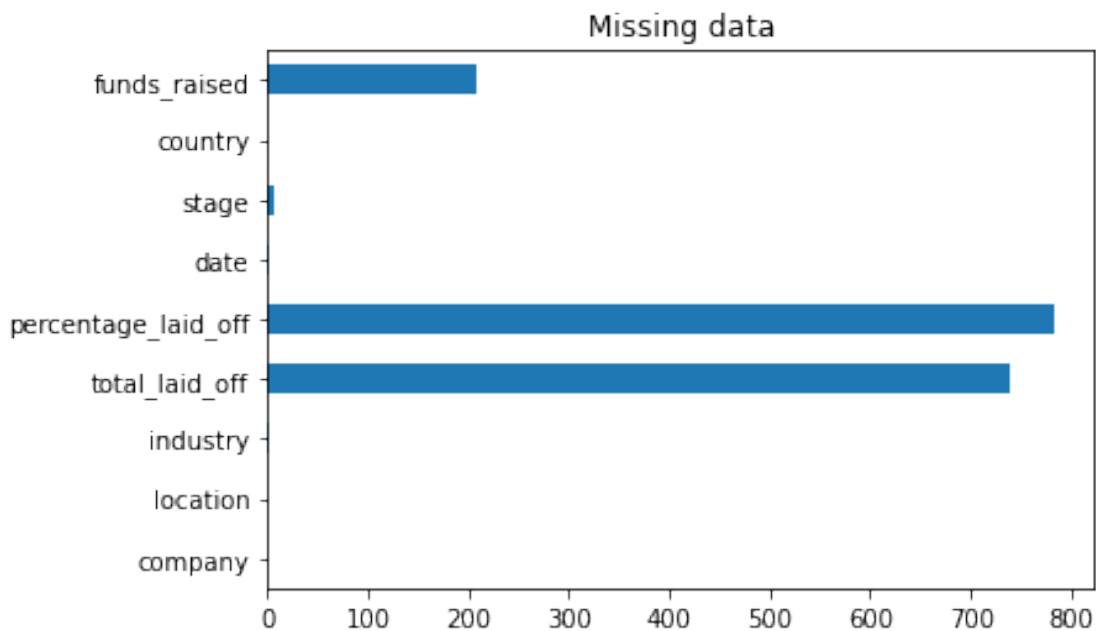
```
df.shape[0]
```

```
print(percentage_missing_data)
```

```
# Kiểm tra có dữ liệu bị duplicate không
print("Number of rows duplicated: "+str(df[df.duplicated()].shape[0]))
```

Missing data:

```
industry          1
total_laid_off    739
percentage_laid_off 784
date              1
stage             6
funds_raised      209
dtype: int64
```



Percentage missing data:

```
industry          0.042427
total_laid_off    31.353415
percentage_laid_off 33.262622
date              0.042427
stage             0.254561
funds_raised      8.867204
dtype: float64
```

Number of rows duplicated: 1

Ta thấy có một số lượng khá lớn dữ liệu bị thiếu ở cột

total_laid_off, percentage_laid_off và funds_raised. Vì có những cột dữ liệu bị thiếu lên đến hơn 30% nên ta không thể chọn phương án bỏ toàn bộ những dòng nào bị thiếu dữ liệu. Bên cạnh đó có tồn tại 1 dòng bị trùng ta cũng cần phải xử lý.

- Đầu tiên ta loại bỏ những dòng có dữ liệu bị thiếu ở cột industry, date và stage vì số lượng bị thiếu không đáng kể.

- Xóa dòng bị trùng dữ liệu.
- Loại bỏ những dòng mà bị thiếu cả hai trường quan trọng là `total_laid_off` và `percentage_laid_off` vì những dòng này không có ý nghĩa để ta phân tích.
- Tiếp theo tùy theo hướng phân tích ta sẽ chọn những cột dữ liệu phù hợp. tạm thời ta sẽ fill bằng 0.

Loại bỏ những dòng có dữ liệu bị thiếu ở cột 'industry', 'date' và 'stage'

```
df = df[~df['industry'].isna()]
df = df[~df['date'].isna()]
df = df[~df['stage'].isna()]
```

Xóa dòng bị trùng dữ liệu.

```
df = df[~df.duplicated()]
```

Loại bỏ những dòng mà bị thiếu cả hai trường quan trọng là 'total_laid_off' và 'percentage_laid_off'

```
df =
df[~(df['total_laid_off'].isna() & df['percentage_laid_off'].isna())]
```

#

```
df = df.fillna(0)
print("After fill null:")
print(df.isna().sum())
```

After fill null:

```
company          0
location         0
industry         0
total_laid_off   0
percentage_laid_off 0
date            0
stage           0
country         0
funds_raised     0
dtype: int64
```

Sau khi đã tạm thời xử lý các dữ liệu bị thiếu ta bắt đầu xử lý kiểu dữ liệu của các cột.

- Chuyển data type của cột `date` sang dạng `datetime`.
- Chuyển data type của các cột đang là dạng `object` ('str') sang dạng `category`

Chuyển data type của cột 'date' sang dạng 'datetime'.

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

Chuyển data type của các cột đang là dạng 'object' sang dạng 'category'

```
object_columns = df.select_dtypes(include=['object']).columns
```

```
for col in object_columns:
    df[col] = df[col].astype('category')
df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1988 entries, 0 to 2355
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   company               1988 non-null   category
1   location              1988 non-null   category
2   industry              1988 non-null   category
3   total_laid_off        1988 non-null   float64
4   percentage_laid_off   1988 non-null   float64
5   date                  1988 non-null   datetime64[ns]
6   stage                 1988 non-null   category
7   country               1988 non-null   category
8   funds_raised          1988 non-null   float64
dtypes: category(5), datetime64[ns](1), float64(3)
memory usage: 178.3 KB

```

Đã xử lý xong kiểu dữ liệu của các cột. Giờ ta quan sát đến giá trị dữ liệu bên trong các cột.

```

# Sắp xếp df theo tên company để quan sát cột 'company'
print(df.sort_values('company')['company'].unique())
a=df['company'].unique()
b=df['company'].str.upper().unique()
print(a.shape[0])
print(b.shape[0])

[' Included Health', '#Paid', '&Open', '100 Thieves', '10X
Genomics', ..., 'iFood', 'iPrice Group', 'iRobot', 'nCino', 'uShip']
Length: 1629
Categories (1629, object): [' Included Health', '#Paid', '&Open', '100
Thieves', ..., 'iPrice Group', 'iRobot', 'nCino', 'uShip']
1629
1624

```

Số lượng tên công ty:

- Data gốc: 1629
- Data sau khi đưa tất cả về uppercase: 1624

Ta thấy có sự khác nhau về số lượng tên công ty khi ta giữ nguyên data gốc và sau khi đưa tất cả về uppercase -> Có những tên của cùng một công ty nhưng chưa được chuẩn hóa cùng 1 quy tắc. -> Đưa tất cả về 1 loại (uppercase) và ta cũng phải xóa các khoảng trắng thừa trong tên công ty phòng trường hợp có tên công ty bị thừa khoảng trắng.

```

# Xử lý dữ liệu cột 'company'
df['company']=df['company'].str.upper()
df['company']=df['company'].str.split().str.join(" ")
df['company'].unique().shape[0]

```

1621

Xử lý tương tự cho cột location, industry, stage và country.

```
# Xử lý dữ liệu cột `location`
df['location']=df['location'].str.upper()
df['location']=df['location'].str.split().str.join(" ")
df['location'].unique().shape[0]
```

166

```
# Xử lý dữ liệu cột `industry`
df['industry']=df['industry'].str.upper()
df['industry']=df['industry'].str.split().str.join(" ")
df['industry'].unique().shape[0]
```

29

```
# Xử lý dữ liệu cột `stage`
df['stage']=df['stage'].str.upper()
df['stage']=df['stage'].str.split().str.join(" ")
df['stage'].unique().shape[0]
```

16

```
# Xử lý dữ liệu cột `country`
df['country']=df['country'].str.upper()
df['country']=df['country'].str.split().str.join(" ")
df['country'].unique().shape[0]
```

51

Tiếp theo ta kiểm tra những cột có kiểu dữ liệu là số để xem có xuất hiện dữ liệu không hợp lệ không.

- Có dòng nào chứa số âm không?
- percentage_laid_off có giá trị nào lớn hơn 1 không?

```
# Kiểm tra giá trị âm
print("Negative total_laid_off: "+str(sum(df.total_laid_off<0)))
print("Negative percentage_laid_off: "+str(sum(df.percentage_laid_off<0)))
print("Negative funds_raised: "+str(sum(df.funds_raised<0)))
```

```
# Kiểm tra giá trị `percentage_laid_off` không hợp lệ
print("percentage_laid_off greater than 1: "+str(sum(df.percentage_laid_off>1)))
```

```
Negative total_laid_off: 0
Negative percentage_laid_off: 0
Negative funds_raised: 0
percentage_laid_off greater than 1: 0
```

Sau khi đã xử lý xong tất cả ta sắp xếp và reset index của data.

```
df=df.sort_values('company')
df=df.reset_index(drop=True)
df.head()
```

	company	location	industry	total_laid_off \
0	#PAID	TORONTO	MARKETING	19.0
1	&OPEN	DUBLIN	MARKETING	9.0
2	100 THIEVES	LOS ANGELES	CONSUMER	12.0
3	10X GENOMICS	SF BAY AREA	HEALTHCARE	100.0
4	1STDIBS	NEW YORK CITY	RETAIL	70.0

	percentage_laid_off	date	stage	country
0	0.17	2023-01-27	SERIES B	CANADA
1	0.09	2022-11-17	SERIES A	IRELAND
2	0.00	2022-07-13	SERIES C	UNITED STATES
3	0.08	2022-08-04	POST-IPO	UNITED STATES
4	0.17	2020-04-02	SERIES D	UNITED STATES

Thêm cột total_employee để phục vụ phân tích sau này

```
df_with_total_employees=df.sort_values(by='total_laid_off',ascending=False)
df_with_total_employees=df_with_total_employees.reset_index(drop=True)
df_with_total_employees=df_with_total_employees[df_with_total_employees['percentage_laid_off']>0]
df_with_total_employees['total_employees']=round(df_with_total_employees['total_laid_off']/df_with_total_employees['percentage_laid_off'])
df_with_total_employees=df_with_total_employees[df_with_total_employees['total_employees']>0]
df_with_total_employees=df_with_total_employees.sort_values(by='total_employees',ascending=False)
df_with_total_employees=df_with_total_employees.reset_index(drop=True)

df2=df_with_total_employees.groupby('company')
['total_employees'].max()
df2.sort_values(ascending=False).head(10)
```

company	
AMAZON	400000.0
GOOGLE	200000.0
MICROSOFT	200000.0
IBM	195000.0
DELL	133000.0
ERICSSON	106250.0
SAP	100000.0
SALESFORCE	100000.0

```
META          84615.0
CISCO         82000.0
Name: total_employees, dtype: float64
```

C. Khám phá mối quan hệ trong dữ liệu

Tìm hiểu mối liên hệ giữa Stage và Percentage_laid_off:

```
df_clean_percentage=df[df['percentage_laid_off']>0]
df_clean_percentage['percentage_laid_off']=df_clean_percentage['percentage_laid_off']*100
df_clean_percentage['percentage_lo_range']=pd.cut(x=df_clean_percentage['percentage_laid_off'], bins=[0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100], labels=['0-10', '10-20', '20-30', '30-40', '40-50', '50-60', '60-70', '70-80', '80-90', '90-100'], include_lowest=True)

df_clean_percentage['stage']=df_clean_percentage['stage'].sort_values()
table = pd.pivot_table(df_clean_percentage,
values='percentage_laid_off',
index='percentage_lo_range', columns='stage', aggfunc='count')
table=table.sort_values(by=['percentage_lo_range'], ascending=False)

fig, (ax1,ax2) = plt.subplots(1,2,figsize=(15, 5), tight_layout=True)

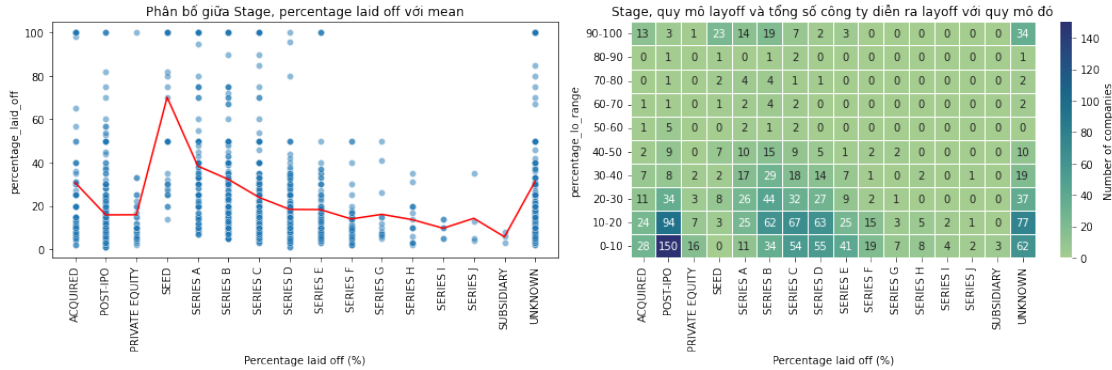
df_clean_percentage=df_clean_percentage.sort_values('stage', ascending=True)
# fig.tight_layout(pad=5.0)
ax1.set_xlabel('Percentage laid off (%)')

ax1.set_title('Phân bố giữa Stage, percentage laid off với mean', wrap=True)
ax1.set_xticklabels(df_clean_percentage['stage'].unique(), rotation=90)

sns.scatterplot(x='stage',
y='percentage_laid_off', data=df_clean_percentage, ax=ax1, alpha=0.5)

sns.lineplot(x='stage',
y='percentage_laid_off', data=df_clean_percentage, ax=ax1, color='red', errorbar=None)

sns.heatmap(table, annot=True, fmt="d", linewidths=.5,
ax=ax2, cmap="crest", cbar_kws={'label': 'Number of companies'})
ax2.set_title('Stage, quy mô layoff và tổng số công ty diễn ra layoff với quy mô đó', wrap=True)
ax2.set_xlabel('Percentage laid off (%)')
plt.show()
```

Một vài điểm đáng chú ý:

- Nhìn vào biểu đồ scatter và heatmap ta có thể thấy dữ liệu phân bố chủ yếu ở mức sa thải dưới 20% nhân sự mỗi đợt.
- Nhìn và heat map ta thấy tương đồng nhiều nhất là 150 công ty ở stage Post - IPO có xu hướng sa thải >10% nhân sự.
- Nhìn vào mean line, các công ty có stage là Seed sa thải % nhân sự rất cao và không bao giờ sa thải dưới 10% nhân sự.
- Dữ liệu chủ yếu tập trung ở nhóm các công ty có stage từ Series A đến Series E. Nhóm này có xu hướng sa thải trung bình 10-20% mỗi đợt.
- Các công ty stage từ Series F đến Series J và Subsidiary có rất ít dữ liệu sa thải và không bao giờ sa thải hơn 50% nhân sự trong một đợt. Có thể lý giải đây là các giai đoạn gọi vốn sau khi đã khá ổn định của các công ty nên ít diễn ra sa thải. Các công ty thuộc Subsidiary thì có công ty mẹ chống lưng nên tình trạng cũng tương đối ổn định.

Stage nào có tổng số laid off cao nhất:

```
stage_df = df.groupby('stage').sum().reset_index()
```

```
px.pie(stage_df, names='stage', values='total_laid_off',
        color_discrete_sequence=px.colors.sequential.thermal,
        title='Tổng số laid off của từng stage (theo tỷ lệ %)')
```

```
{
  "config": {
    "plotlyServerURL": "https://plot.ly",
    "data": [
      {
        "domain": [
          {
            "x": [0, 1],
            "y": [0, 1],
            "hovertemplate": "stage=%{label}<br>total_laid_off=%{value}<extra></extra>",
            "labels": [
              "ACQUIRED",
              "POST-IPO",
              "PRIVATE EQUITY",
              "SEED",
              "SERIES A",
              "SERIES B",
              "SERIES C",
              "SERIES D",
              "SERIES E",
              "SERIES F",
              "SERIES G",
              "SERIES H",
              "SERIES I",
              "SERIES J",
              "SUBSIDIARY",
              "UNKNOWN"
            ],
            "legendgroup": "",
            "name": "",
            "showlegend": true,
            "type": "pie",
            "values": [
              27576, 203632, 7957, 1636, 5678, 15311, 20017, 19225, 12697, 9932, 3697, 7244, 2855, 3570, 1094, 40716
            ]
          }
        ],
        "layout": {
          "legend": {
            "tracegroupgap": 0,
            "piecolorway": [
              "rgb(3, 35, 51)",
              "rgb(13, 48, 100)",
              "rgb(53, 50, 155)",
              "rgb(93, 62, 153)",
              "rgb(126, 77, 143)",
              "rgb(158, 89, 135)",
              "rgb(193, 100, 121)",
              "rgb(225, 113, 97)",
              "rgb(246, 139, 69)",
              "rgb(251, 173, 60)",
              "rgb(246, 211, 135)",
              "rgb(251, 246, 135)",
              "rgb(251, 246, 135)",
              "rgb(251, 246, 135)",
              "rgb(251, 246, 135)"
            ]
          }
        }
      }
    ]
  }
}
```

```

70)","rgb(231, 250, 90)"],["template":{"data":{"bar":[{"error_x":
{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},{"type":"bar"}]},"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},{"type":"barpolar"}]},"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}]},"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":"","type":"choropleth"}]},"contour":
[{"colorbar":{"outlinewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]},"type":"contour"}]},"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":"","type":"contourcarpet"}]},"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]},"type":"heatmap"}]},"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":"","colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]},"type":"heatmapgl"}]},"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},{"type":"histogram"}]},"
histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]},"type":"histogram2d"}]},"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]},"type":"histogram2dcontour"}]},"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":"","type":"mesh3d"}]},"parcoords":[{"line":

```

```

{"colorbar":{"outlinewidth":0,"ticks":""},"type":"parcoords"}], "pie":
[{"automargin":true, "type":"pie"}], "scatter":[{"fillpattern":
{"fillmode":"overlay", "size":10, "solidity":0.2}, "type":"scatter"}], "scatter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"scatter3d"}], "scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"scattercarpet"}], "scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"scattergeo"}], "scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"scattergl"}], "scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"scattermapbox"}], "scatterpolar":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"scatterpolar"}], "scatterpolargl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"scatterpolargl"}], "scatterternary":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"scatterternary"}], "surface":
[{"colorbar":{"outlinewidth":0,"ticks":""}, "colorscale":
[[0, "#0d0887"], [0.11111111111111111, "#46039f"],
[0.22222222222222222, "#7201a8"], [0.33333333333333333, "#9c179e"],
[0.44444444444444444, "#bd3786"], [0.55555555555555556, "#d8576b"],
[0.66666666666666666, "#ed7953"], [0.77777777777777778, "#fb9f3a"],
[0.88888888888888888, "#fdca26"],
[1, "#f0f921"]], "type":"surface"}], "table":[{"cells":{"fill":
{"color":"#EBF0F8"}, "line":{"color":"white"}}, "header":{"fill":
{"color":"#C8D4E3"}, "line":
{"color":"white"}}, "type":"table"}], "layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f", "arrowhead":0, "arrowwidth":1}, "autotypenumbers":
"strict", "coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}, "colorscale":{"diverging":
[[0, "#8e0152"], [0.1, "#c51b7d"], [0.2, "#de77ae"], [0.3, "#f1b6da"],
[0.4, "#fde0ef"], [0.5, "#f7f7f7"], [0.6, "#e6f5d0"], [0.7, "#b8e186"],
[0.8, "#7fb341"], [0.9, "#4d9221"], [1, "#276419"]], "sequential":
[[0, "#0d0887"], [0.11111111111111111, "#46039f"],
[0.22222222222222222, "#7201a8"], [0.33333333333333333, "#9c179e"],
[0.44444444444444444, "#bd3786"], [0.55555555555555556, "#d8576b"],
[0.66666666666666666, "#ed7953"], [0.77777777777777778, "#fb9f3a"],
[0.88888888888888888, "#fdca26"], [1, "#f0f921"]], "sequentialminus":
[[0, "#0d0887"], [0.11111111111111111, "#46039f"],
[0.22222222222222222, "#7201a8"], [0.33333333333333333, "#9c179e"],
[0.44444444444444444, "#bd3786"], [0.55555555555555556, "#d8576b"],
[0.66666666666666666, "#ed7953"], [0.77777777777777778, "#fb9f3a"],
[0.88888888888888888, "#fdca26"], [1, "#f0f921"]], "colorway":
["#636efa", "#EF553B", "#00cc96", "#ab63fa", "#FFA15A", "#19d3f3", "#FF6692",
"#B6E880", "#FF97FF", "#FECB52"], "font":{"color":"#2a3f5f"}, "geo":
{"bgcolor":"white", "lakecolor":"white", "landcolor":"#E5ECF6", "showlakes":
true, "showland":true, "subunitcolor":"white"}, "hoverlabel":

```

```
{
  "align": "left",
  "hovermode": "closest",
  "mapbox": {
    "style": "light",
    "paper_bgcolor": "white",
    "plot_bgcolor": "#E5ECF6",
    "polar": {
      "angularaxis": {
        "gridcolor": "white",
        "linecolor": "white",
        "ticks": ""
      },
      "bgcolor": "#E5ECF6",
      "radialaxis": {
        "gridcolor": "white",
        "linecolor": "white",
        "ticks": ""
      }
    },
    "scene": {
      "xaxis": {
        "backgroundcolor": "#E5ECF6",
        "gridcolor": "white",
        "gridwidth": 2,
        "linecolor": "white",
        "showbackground": true,
        "ticks": "",
        "zerolinecolor": "white"
      },
      "yaxis": {
        "backgroundcolor": "#E5ECF6",
        "gridcolor": "white",
        "gridwidth": 2,
        "linecolor": "white",
        "showbackground": true,
        "ticks": "",
        "zerolinecolor": "white"
      },
      "zaxis": {
        "backgroundcolor": "#E5ECF6",
        "gridcolor": "white",
        "gridwidth": 2,
        "linecolor": "white",
        "showbackground": true,
        "ticks": "",
        "zerolinecolor": "white"
      }
    },
    "shapedefaults": {
      "line": {
        "color": "#2a3f5f"
      }
    },
    "ternary": {
      "aaxis": {
        "gridcolor": "white",
        "linecolor": "white",
        "ticks": ""
      },
      "baxis": {
        "gridcolor": "white",
        "linecolor": "white",
        "ticks": ""
      },
      "bgcolor": "#E5ECF6",
      "caxis": {
        "gridcolor": "white",
        "linecolor": "white",
        "ticks": ""
      }
    },
    "title": {
      "x": 5.0e-2,
      "xaxis": {
        "automargin": true,
        "gridcolor": "white",
        "linecolor": "white",
        "ticks": ""
      },
      "title": {
        "standoff": 15,
        "zerolinecolor": "white",
        "zerolinewidth": 2
      },
      "yaxis": {
        "automargin": true,
        "gridcolor": "white",
        "linecolor": "white",
        "ticks": ""
      },
      "title": {
        "standoff": 15,
        "zerolinecolor": "white",
        "zerolinewidth": 2
      }
    },
    "text": "Tổng số laid off của từng stage (theo tỷ lệ %)"
  }
}
```

Từ biểu đồ tổng số laid off của từng stage ta có thể thấy:

- Stage POST-IPO chiếm 53.2% tổng số lượng laid off, chiếm hơn 1/2 tổng số laid off của tất cả các công ty.
- POST-IPO cũng chính là Stage có tổng số laid off cao nhất.

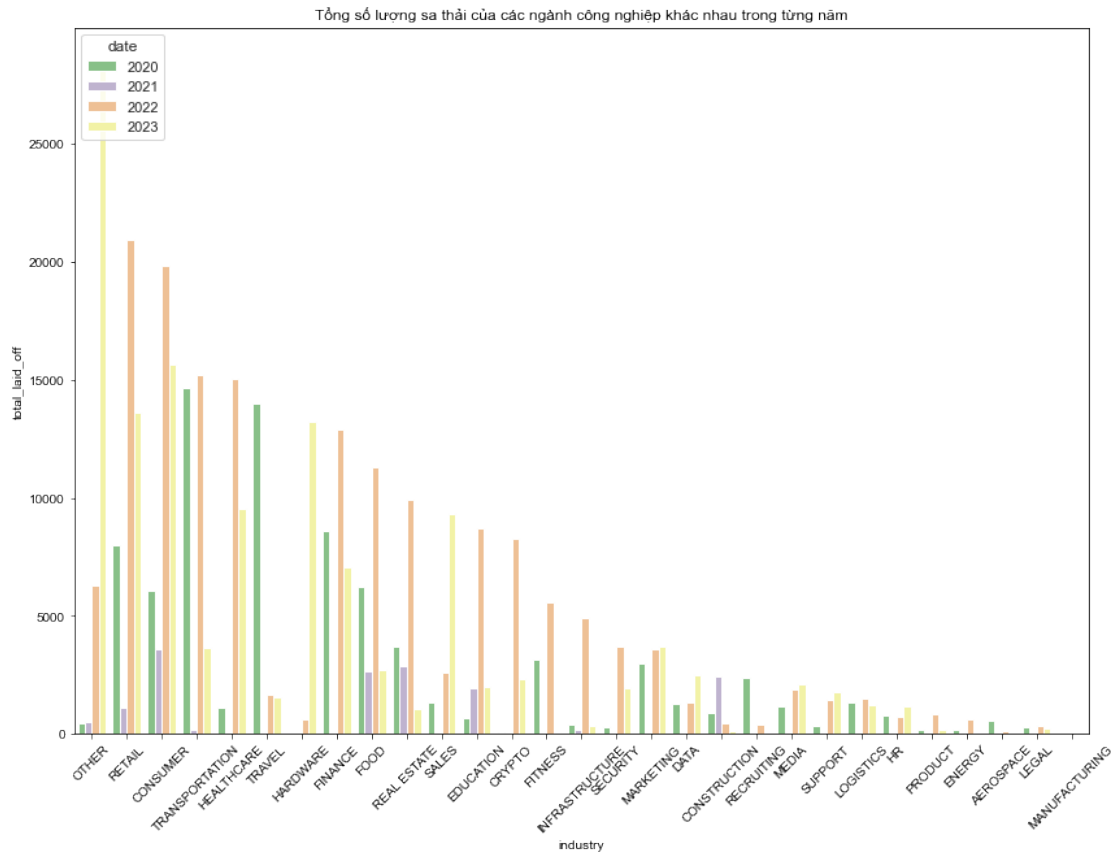
Tổng số lượng nhân viên bị sa thải của các ngành công nghiệp khác nhau theo từng năm:

```
df_tmp = df.reset_index()
df_tmp['date'] = pd.to_datetime(df_tmp['date'])
df_industry = df_tmp.groupby([df_industry, df.date.dt.year]).sum()
df_industry = df_industry.reset_index()

plt.figure(figsize=(14, 10))
plt.xticks(rotation=45)
plt.title("Tổng số lượng sa thải của các ngành công nghiệp khác nhau trong từng năm")
sns.set(style="white", palette="Accent", color_codes=True)

sns.barplot(data=df_industry.sort_values(by=['total_laid_off', 'date'], ascending=False), x="industry", y="total_laid_off", hue="date")
```

```
<AxesSubplot: title={'center': 'Tổng số lượng sa thải của các ngành công nghiệp khác nhau trong từng năm'}, xlabel='industry', ylabel='total_laid_off'>
```



Xu hướng sa thải hằng năm của thế giới cho thấy rằng:

- Việc sa thải cao hơn vào năm 2020, sau đó giảm và năm 2021 và tăng vọt vào năm 2021.
- Vào năm 2020, các ngành công nghiệp Transportation (vận tải) và Travel (du lịch) đã bị ảnh hưởng nhiều nhất bởi dịch COVID-19. Điều đó được thể hiện thông qua số lượng sa thải vào năm 2020. Nhưng 2 ngành này đã phục hồi vào những năm sau đó.
- Các ngành công nghiệp Retail (bán lẻ), Consumer (tiêu dùng) và Food (thực phẩm) đã phải đối mặt với việc sa thải gia tăng mạnh mẽ qua các năm.

Theo từng năm, quốc gia có tổng số lượng sa thải lớn nhất:

```
df_tmp['year'] = df_tmp['date'].dt.year
layoff_year_country =
df_tmp.groupby(['year', 'country']).sum().reset_index()
[['year', 'country', 'total_laid_off']]

layoff_year_country =
layoff_year_country.sort_values(["year", "total_laid_off"], ascending=False)
```

```
px.bar(layoff_year_country,x='year',y='total_laid_off',color='country',
      text='country',
      title='Tổng lượng sa thải của từng năm theo quốc gia'
      )
```

```
{
  "config": {
    "plotlyServerURL": "https://plot.ly",
    "data": [
      {
        "alignmentgroup": "True",
        "hovertemplate": "country=%{text}<br>year=%{x}<br>total_laid_off=%{y}<extra></extra>",
        "legendgroup": "UNITED STATES",
        "marker": {
          "color": "#636efa",
          "pattern": {
            "shape": ""
          },
          "name": "UNITED STATES",
          "offsetgroup": "UNITED STATES",
          "orientation": "v",
          "showlegend": true,
          "text": [
            "UNITED STATES",
            "UNITED STATES",
            "UNITED STATES",
            "UNITED STATES"
          ],
          "textposition": "auto",
          "type": "bar",
          "x": [
            2023, 2022, 2021, 2020
          ],
          "xaxis": "x",
          "y": [
            89434, 106488, 9470, 50385
          ],
          "yaxis": "y"
        },
        "alignmentgroup": "True",
        "hovertemplate": "country=%{text}<br>year=%{x}<br>total_laid_off=%{y}<extra></extra>",
        "legendgroup": "SWEDEN",
        "marker": {
          "color": "#EF553B",
          "pattern": {
            "shape": ""
          },
          "name": "SWEDEN",
          "offsetgroup": "SWEDEN",
          "orientation": "v",
          "showlegend": true,
          "text": [
            "SWEDEN", "SWEDEN", "SWEDEN"
          ],
          "textposition": "auto",
          "type": "bar",
          "x": [
            2023, 2022, 2020
          ],
          "xaxis": "x",
          "y": [
            9100, 2064, 100
          ],
          "yaxis": "y"
        },
        "alignmentgroup": "True",
        "hovertemplate": "country=%{text}<br>year=%{x}<br>total_laid_off=%{y}<extra></extra>",
        "legendgroup": "NETHERLANDS",
        "marker": {
          "color": "#00cc96",
          "pattern": {
            "shape": ""
          },
          "name": "NETHERLANDS",
          "offsetgroup": "NETHERLANDS",
          "orientation": "v",
          "showlegend": true,
          "text": [
            "NETHERLANDS", "NETHERLANDS", "NETHERLANDS"
          ],
          "textposition": "auto",
          "type": "bar",
          "x": [
            2023, 2022, 2020
          ],
          "xaxis": "x",
          "y": [
            7500, 5120, 4600
          ],
          "yaxis": "y"
        },
        "alignmentgroup": "True",
        "hovertemplate": "country=%{text}<br>year=%{x}<br>total_laid_off=%{y}<extra></extra>",
        "legendgroup": "INDIA",
        "marker": {
          "color": "#ab63fa",
          "pattern": {
            "shape": ""
          },
          "name": "INDIA",
          "offsetgroup": "INDIA",
          "orientation": "v",
          "showlegend": true,
          "text": [
            "INDIA", "INDIA", "INDIA", "INDIA"
          ],
          "textposition": "auto",
          "type": "bar",
          "x": [
            2023, 2022, 2021, 2020
          ],
          "xaxis": "x",
          "y": [
            4717, 14224, 4080, 12932
          ],
          "yaxis": "y"
        },
        "alignmentgroup": "True",
        "hovertemplate": "country=%{text}<br>year=%{x}<br>total_laid_off=%{y}<extra></extra>",
        "legendgroup": "GERMANY",
        "marker": {
          "color": "#FFA15A",
          "pattern": {
            "shape": ""
          },
          "name": "GERMANY",
          "offsetgroup": "GERMANY",
          "orientation": "v",
          "showlegend": true,
          "text": [
            "GERMANY", "GERMANY", "GERMANY", "GERMANY"
          ],
          "textposition": "auto",
          "type": "bar",
          "x": [
            2023, 2022, 2021, 2020
          ],
          "xaxis": "x",
          "y": [
            4176, 3607, 387, 531
          ],
          "yaxis": "y"
        },
        "alignmentgroup": "True",
        "hovertemplate": "country=%{text}<br>year=%{x}<br>total_laid_off=%{y}<extra></extra>",
        "legendgroup": "UNITED KINGDOM",
        "marker": {
          "color": "#19d3f3",
          "pattern": {
            "shape": ""
          },
          "name": "UNITED KINGDOM",
          "offsetgroup": "UNITED KINGDOM",
          "orientation": "v",
          "showlegend": true,
          "text": [
            "UNITED KINGDOM", "UNITED KINGDOM", "UNITED KINGDOM", "UNITED KINGDOM"
          ],
          "textposition": "auto",
          "type": "bar",
          "x": [
            2023, 2022, 2021, 2020
          ],
          "xaxis": "x",
          "y": [
            1234, 1123, 1012, 901
          ],
          "yaxis": "y"
        }
      ]
    }
  }
}
```



```

{"shape":"","},"name":"UNITED KINGDOM","offsetgroup":"UNITED
KINGDOM","orientation":"v","showlegend":true,"text":["UNITED
KINGDOM","UNITED KINGDOM","UNITED KINGDOM","UNITED
KINGDOM"],"textposition":"auto","type":"bar","x":
[2023,2022,2021,2020],"xaxis":"x","y":
[2171,2349,20,1858],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"BRAZIL","marker":{"color":"#FF6692","pattern":
{"shape":"","},"name":"BRAZIL","offsetgroup":"BRAZIL","orientation":"v"
,"showlegend":true,"text":
["BRAZIL","BRAZIL","BRAZIL"],"textposition":"auto","type":"bar","x":
[2023,2022,2020],"xaxis":"x","y":[2161,4889,3341],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"CHINA","marker":{"color":"#B6E880","pattern":
{"shape":"","},"name":"CHINA","offsetgroup":"CHINA","orientation":"v","
showlegend":true,"text":
["CHINA","CHINA","CHINA"],"textposition":"auto","type":"bar","x":
[2023,2022,2021],"xaxis":"x","y":[1475,2630,1800],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"CANADA","marker":{"color":"#FF97FF","pattern":
{"shape":"","},"name":"CANADA","offsetgroup":"CANADA","orientation":"v"
,"showlegend":true,"text":
["CANADA","CANADA","CANADA","CANADA"],"textposition":"auto","type":"ba
r","x":[2023,2022,2021,2020],"xaxis":"x","y":
[1127,3936,45,1211],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"AUSTRALIA","marker":
{"color":"#FECB52","pattern":
{"shape":"","},"name":"AUSTRALIA","offsetgroup":"AUSTRALIA","orientatio
n":"v","showlegend":true,"text":
["AUSTRALIA","AUSTRALIA","AUSTRALIA"],"textposition":"auto","type":"ba
r","x":[2023,2022,2020],"xaxis":"x","y":[1010,1188,126],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"ISRAEL","marker":{"color":"#636efa","pattern":
{"shape":"","},"name":"ISRAEL","offsetgroup":"ISRAEL","orientation":"v"
,"showlegend":true,"text":
["ISRAEL","ISRAEL","ISRAEL"],"textposition":"auto","type":"bar","x":
[2023,2022,2020],"xaxis":"x","y":[680,2734,224],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"NIGERIA","marker":{"color":"#EF553B","pattern":
{"shape":"","},"name":"NIGERIA","offsetgroup":"NIGERIA","orientation":"
v","showlegend":true,"text":
["NIGERIA","NIGERIA","NIGERIA"],"textposition":"auto","type":"bar","x"
:[2023,2022,2020],"xaxis":"x","y":[471,1020,391],"yaxis":"y"},

```

```

{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"FRANCE","marker":{"color":"#00cc96","pattern":
{"shape":""}},{"name":"FRANCE","offsetgroup":"FRANCE","orientation":"v"
,"showlegend":true,"text":
["FRANCE","FRANCE"],"textposition":"auto","type":"bar","x":
[2023,2022],"xaxis":"x","y":[408,507],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"INDONESIA","marker":
{"color":"#ab63fa","pattern":
{"shape":""}},{"name":"INDONESIA","offsetgroup":"INDONESIA","orientatio
n":"v","showlegend":true,"text":
["INDONESIA","INDONESIA","INDONESIA"],"textposition":"auto","type":"ba
r","x":[2023,2022,2020],"xaxis":"x","y":[360,2281,880],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"SPAIN","marker":{"color":"#FFA15A","pattern":
{"shape":""}},{"name":"SPAIN","offsetgroup":"SPAIN","orientation":"v","
showlegend":true,"text":
["SPAIN"],"textposition":"auto","type":"bar","x":
[2023],"xaxis":"x","y":[250],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"COLOMBIA","marker":
{"color":"#19d3f3","pattern":
{"shape":""}},{"name":"COLOMBIA","offsetgroup":"COLOMBIA","orientation"
:"v","showlegend":true,"text":
["COLOMBIA","COLOMBIA"],"textposition":"auto","type":"bar","x":
[2023,2022],"xaxis":"x","y":[100,30],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"JAPAN","marker":{"color":"#FF6692","pattern":
{"shape":""}},{"name":"JAPAN","offsetgroup":"JAPAN","orientation":"v","
showlegend":true,"text":
["JAPAN"],"textposition":"auto","type":"bar","x":
[2023],"xaxis":"x","y":[85],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"SINGAPORE","marker":
{"color":"#B6E880","pattern":
{"shape":""}},{"name":"SINGAPORE","offsetgroup":"SINGAPORE","orientatio
n":"v","showlegend":true,"text":
["SINGAPORE","SINGAPORE","SINGAPORE","SINGAPORE"],"textposition":"auto
","type":"bar","x":[2023,2022,2021,2020],"xaxis":"x","y":
[69,3544,21,2361],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"ESTONIA","marker":{"color":"#FF97FF","pattern":
{"shape":""}},{"name":"ESTONIA","offsetgroup":"ESTONIA","orientation":"

```



```

v", "showlegend": true, "text":
["ESTONIA", "ESTONIA", "ESTONIA"], "textposition": "auto", "type": "bar", "x":
[2023, 2022, 2020], "xaxis": "x", "y": [66, 143, 124], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "IRELAND", "marker": {"color": "#FECB52", "pattern":
{"shape": ""}}, "name": "IRELAND", "offsetgroup": "IRELAND", "orientation": "
v", "showlegend": true, "text":
["IRELAND", "IRELAND", "IRELAND"], "textposition": "auto", "type": "bar", "x":
[2023, 2022, 2020], "xaxis": "x", "y": [27, 209, 21], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "CHILE", "marker": {"color": "#636efa", "pattern":
{"shape": ""}}, "name": "CHILE", "offsetgroup": "CHILE", "orientation": "v",
showlegend": true, "text":
["CHILE", "CHILE"], "textposition": "auto", "type": "bar", "x":
[2023, 2022], "xaxis": "x", "y": [0, 30], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "FINLAND", "marker": {"color": "#EF553B", "pattern":
{"shape": ""}}, "name": "FINLAND", "offsetgroup": "FINLAND", "orientation": "
v", "showlegend": true, "text":
["FINLAND", "FINLAND"], "textposition": "auto", "type": "bar", "x":
[2023, 2022], "xaxis": "x", "y": [0, 250], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "ITALY", "marker": {"color": "#00cc96", "pattern":
{"shape": ""}}, "name": "ITALY", "offsetgroup": "ITALY", "orientation": "v",
showlegend": true, "text":
["ITALY"], "textposition": "auto", "type": "bar", "x":
[2023], "xaxis": "x", "y": [0], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></extra>", "legendgroup": "SOUTH
KOREA", "marker": {"color": "#ab63fa", "pattern":
{"shape": ""}}, "name": "SOUTH KOREA", "offsetgroup": "SOUTH
KOREA", "orientation": "v", "showlegend": true, "text": ["SOUTH
KOREA"], "textposition": "auto", "type": "bar", "x": [2023], "xaxis": "x", "y":
[0], "yaxis": "y"}, {"alignmentgroup": "True", "hovertemplate": "country=%
{text}<br>year=%{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "AUSTRIA", "marker": {"color": "#FFA15A", "pattern":
{"shape": ""}}, "name": "AUSTRIA", "offsetgroup": "AUSTRIA", "orientation": "
v", "showlegend": true, "text":
["AUSTRIA"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [570], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></extra>", "legendgroup": "UNITED ARAB
EMIRATES", "marker": {"color": "#19d3f3", "pattern":
{"shape": ""}}, "name": "UNITED ARAB EMIRATES", "offsetgroup": "UNITED ARAB
EMIRATES", "orientation": "v", "showlegend": true, "text": ["UNITED ARAB
EMIRATES", "UNITED ARAB

```

```

EMIRATES"], "textposition": "auto", "type": "bar", "x":
[2022, 2020], "xaxis": "x", "y": [459, 536], "yaxis": "y"},
{"alignmentgroup": "True", "hovern": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "RUSSIA", "marker": {"color": "#FF6692", "pattern":
{"shape": ""}}, "name": "RUSSIA", "offsetgroup": "RUSSIA", "orientation": "v",
"showlegend": true, "text":
["RUSSIA"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [400], "yaxis": "y"},
{"alignmentgroup": "True", "hovern": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "KENYA", "marker": {"color": "#B6E880", "pattern":
{"shape": ""}}, "name": "KENYA", "offsetgroup": "KENYA", "orientation": "v",
"showlegend": true, "text":
["KENYA"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [349], "yaxis": "y"},
{"alignmentgroup": "True", "hovern": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "ARGENTINA", "marker":
{"color": "#FF97FF", "pattern":
{"shape": ""}}, "name": "ARGENTINA", "offsetgroup": "ARGENTINA", "orientation": "v",
"showlegend": true, "text":
["ARGENTINA"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [323], "yaxis": "y"},
{"alignmentgroup": "True", "hovern": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "SENEGAL", "marker": {"color": "#FECB52", "pattern":
{"shape": ""}}, "name": "SENEGAL", "offsetgroup": "SENEGAL", "orientation": "v",
"showlegend": true, "text":
["SENEGAL"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [300], "yaxis": "y"},
{"alignmentgroup": "True", "hovern": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "DENMARK", "marker": {"color": "#636efa", "pattern":
{"shape": ""}}, "name": "DENMARK", "offsetgroup": "DENMARK", "orientation": "v",
"showlegend": true, "text":
["DENMARK", "DENMARK"], "textposition": "auto", "type": "bar", "x":
[2022, 2020], "xaxis": "x", "y": [200, 40], "yaxis": "y"},
{"alignmentgroup": "True", "hovern": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "MEXICO", "marker": {"color": "#EF553B", "pattern":
{"shape": ""}}, "name": "MEXICO", "offsetgroup": "MEXICO", "orientation": "v",
"showlegend": true, "text":
["MEXICO", "MEXICO"], "textposition": "auto", "type": "bar", "x":
[2022, 2020], "xaxis": "x", "y": [180, 90], "yaxis": "y"},
{"alignmentgroup": "True", "hovern": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "NORWAY", "marker": {"color": "#00cc96", "pattern":
{"shape": ""}}, "name": "NORWAY", "offsetgroup": "NORWAY", "orientation": "v",
"showlegend": true, "text":

```

```

["NORWAY"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [140], "yaxis": "y"},
{"alignmentgroup": "True", "hovernment": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "ROMANIA", "marker": {"color": "#ab63fa", "pattern":
{"shape": ""}}, "name": "ROMANIA", "offsetgroup": "ROMANIA", "orientation":
"v", "showlegend": true, "text":
["ROMANIA"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [80], "yaxis": "y"},
{"alignmentgroup": "True", "hovernment": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "SEYCHELLES", "marker":
{"color": "#FFA15A", "pattern":
{"shape": ""}}, "name": "SEYCHELLES", "offsetgroup": "SEYCHELLES", "orientat
ion": "v", "showlegend": true, "text":
["SEYCHELLES"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [75], "yaxis": "y"},
{"alignmentgroup": "True", "hovernment": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "LITHUANIA", "marker":
{"color": "#19d3f3", "pattern":
{"shape": ""}}, "name": "LITHUANIA", "offsetgroup": "LITHUANIA", "orientatio
n": "v", "showlegend": true, "text":
["LITHUANIA"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [60], "yaxis": "y"},
{"alignmentgroup": "True", "hovernment": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "THAILAND", "marker":
{"color": "#FF6692", "pattern":
{"shape": ""}}, "name": "THAILAND", "offsetgroup": "THAILAND", "orientation"
: "v", "showlegend": true, "text":
["THAILAND"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [55], "yaxis": "y"},
{"alignmentgroup": "True", "hovernment": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "MALAYSIA", "marker":
{"color": "#B6E880", "pattern":
{"shape": ""}}, "name": "MALAYSIA", "offsetgroup": "MALAYSIA", "orientation"
: "v", "showlegend": true, "text":
["MALAYSIA", "MALAYSIA"], "textposition": "auto", "type": "bar", "x":
[2022, 2020], "xaxis": "x", "y": [50, 50], "yaxis": "y"},
{"alignmentgroup": "True", "hovernment": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "LUXEMBOURG", "marker":
{"color": "#FF97FF", "pattern":
{"shape": ""}}, "name": "LUXEMBOURG", "offsetgroup": "LUXEMBOURG", "orientat
ion": "v", "showlegend": true, "text":
["LUXEMBOURG"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [45], "yaxis": "y"},
{"alignmentgroup": "True", "hovernment": "country=%{text}<br>year=%

```

```

{x}<br>total_laid_off=%{y}<extra></extra>", "legendgroup": "NEW
ZEALAND", "marker": {"color": "#FECB52", "pattern":
{"shape": ""}}, "name": "NEW ZEALAND", "offsetgroup": "NEW
ZEALAND", "orientation": "v", "showlegend": true, "text": ["NEW
ZEALAND"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [45], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></extra>", "legendgroup": "HONG
KONG", "marker": {"color": "#636efa", "pattern": {"shape": ""}}, "name": "HONG
KONG", "offsetgroup": "HONG
KONG", "orientation": "v", "showlegend": true, "text": ["HONG KONG", "HONG
KONG"], "textposition": "auto", "type": "bar", "x":
[2022, 2020], "xaxis": "x", "y": [30, 700], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "POLAND", "marker": {"color": "#EF553B", "pattern":
{"shape": ""}}, "name": "POLAND", "offsetgroup": "POLAND", "orientation": "v",
"showlegend": true, "text":
["POLAND"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [25], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "EGYPT", "marker": {"color": "#00cc96", "pattern":
{"shape": ""}}, "name": "EGYPT", "offsetgroup": "EGYPT", "orientation": "v", "
showlegend": true, "text":
["EGYPT"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [0], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "HUNGARY", "marker": {"color": "#ab63fa", "pattern":
{"shape": ""}}, "name": "HUNGARY", "offsetgroup": "HUNGARY", "orientation": "
v", "showlegend": true, "text":
["HUNGARY"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [0], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "PAKISTAN", "marker":
{"color": "#FFA15A", "pattern":
{"shape": ""}}, "name": "PAKISTAN", "offsetgroup": "PAKISTAN", "orientation"
: "v", "showlegend": true, "text":
["PAKISTAN"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [0], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>", "legendgroup": "TURKEY", "marker": {"color": "#19d3f3", "pattern":
{"shape": ""}}, "name": "TURKEY", "offsetgroup": "TURKEY", "orientation": "v",
"showlegend": true, "text":
["TURKEY"], "textposition": "auto", "type": "bar", "x":
[2022], "xaxis": "x", "y": [0], "yaxis": "y"},
{"alignmentgroup": "True", "hovertemplate": "country=%{text}<br>year=%

```

```

{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"VIETNAM","marker":{"color":"#FF6692","pattern":
{"shape":""}},"name":"VIETNAM","offsetgroup":"VIETNAM","orientation":
"v","showlegend":true,"text":
["VIETNAM","VIETNAM"],"textposition":"auto","type":"bar","x":
[2022,2020],"xaxis":"x","y":[0,0],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"MYANMAR","marker":{"color":"#B6E880","pattern":
{"shape":""}},"name":"MYANMAR","offsetgroup":"MYANMAR","orientation":
"v","showlegend":true,"text":
["MYANMAR"],"textposition":"auto","type":"bar","x":
[2020],"xaxis":"x","y":[200],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"BULGARIA","marker":
{"color":"#FF97FF","pattern":
{"shape":""}},"name":"BULGARIA","offsetgroup":"BULGARIA","orientation":
"v","showlegend":true,"text":
["BULGARIA"],"textposition":"auto","type":"bar","x":
[2020],"xaxis":"x","y":[120],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"PORTUGAL","marker":
{"color":"#FECB52","pattern":
{"shape":""}},"name":"PORTUGAL","offsetgroup":"PORTUGAL","orientation":
"v","showlegend":true,"text":
["PORTUGAL"],"textposition":"auto","type":"bar","x":
[2020],"xaxis":"x","y":[115],"yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"country=%{text}<br>year=%
{x}<br>total_laid_off=%{y}<extra></
extra>","legendgroup":"SWITZERLAND","marker":
{"color":"#636efa","pattern":
{"shape":""}},"name":"SWITZERLAND","offsetgroup":"SWITZERLAND","orient
ation":"v","showlegend":true,"text":
["SWITZERLAND"],"textposition":"auto","type":"bar","x":
[2020],"xaxis":"x","y":[62],"yaxis":"y"},"layout":
{"barmode":"relative","legend":{"title":
{"text":"country"},"tracegroupgap":0},"template":{"data":{"bar":
[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch

```



```

oropleth":[{"colorbar":
{"linewidth":0,"ticks":"","type":"choropleth"}],"contour":
[{"colorbar":{"linewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"linewidth":0,"ticks":"","type":"contourcarpet"}],"heatmap":
[{"colorbar":{"linewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"linewidth":0,"ticks":"","colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"histogram"}],
"histogram2d":[{"colorbar":{"linewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"linewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"linewidth":0,"ticks":"","type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"linewidth":0,"ticks":"","type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"scatter3d":
[{"line":{"colorbar":{"linewidth":0,"ticks":"","type":"scatter3d"}},
{"marker":
{"colorbar":
{"linewidth":0,"ticks":"","type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":

```

```

{"linewidth":0,"ticks":"","type":"scattergl"},"scattermapbox":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scattermapbox"},"scatterpolar":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scatterpolar"},"scatterpolargl":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scatterpolargl"},"scatterternary":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scatterternary"},"surface":
[{"colorbar":{"linewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"},"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers":
"strict","coloraxis":{"colorbar":
{"linewidth":0,"ticks":"","colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fb341"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692",
"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlakes":
true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","polar":
{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":"","bgcolor":"#E5ECF6",
"radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":"","scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":
"white","showbackground":true,"ticks":"","zerolinecolor":"white"},
"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":

```

```

lor:"white", "showbackground":true, "ticks":"","zerolinecolor":"white"}
, "zaxis":
{"backgroundcolor":"#E5ECF6", "gridcolor":"white", "gridwidth":2, "linecolor":
"white", "showbackground":true, "ticks":"","zerolinecolor":"white"}
}, "shapedefaults":{"line":{"color":"#2a3f5f"}}, "ternary":{"aaxis":
{"gridcolor":"white", "linecolor":"white", "ticks":""}, "baxis":
{"gridcolor":"white", "linecolor":"white", "ticks":""}, "bgcolor":"#E5ECF
6", "caxis":
{"gridcolor":"white", "linecolor":"white", "ticks":""}}, "title":
{"x":5.0e-2}, "xaxis":
{"automargin":true, "gridcolor":"white", "linecolor":"white", "ticks":"","
"title":
{"standoff":15}, "zerolinecolor":"white", "zerolinewidth":2}, "yaxis":
{"automargin":true, "gridcolor":"white", "linecolor":"white", "ticks":"","
"title":
{"standoff":15}, "zerolinecolor":"white", "zerolinewidth":2}}}, "title":
{"text":"Tổng lượng sa thải của từng năm theo quốc gia"}, "xaxis":
{"anchor":"y", "domain":[0,1], "title":{"text":"year"}}, "yaxis":
{"anchor":"x", "domain":[0,1], "title":{"text":"total_laid_off"}}}

```

Từ biểu đồ trên ta thấy:

- Tổng số lượng sa thải giảm trong giai đoạn từ 2020 sang 2021. Nhưng sau đó lại tăng mạnh vào giai đoạn 2022.
- Với tất cả các năm 2020, 2021, 2022, 2023 thì United States luôn là quốc gia có số lượng sa thải lớn nhất. Thường chiếm gần 1/2 tổng số lượng sa thải của tất cả các quốc gia.

Mối quan hệ giữa total layoffs và funds raised của các công ty:

```

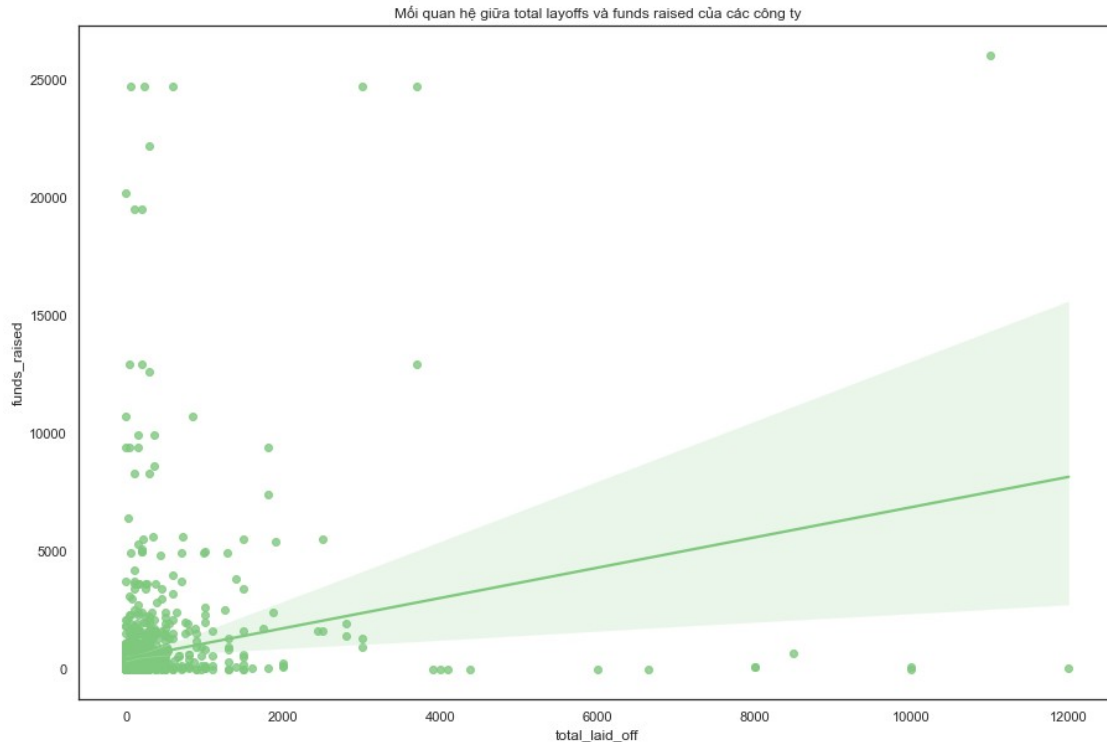
funds_raised = df[(df.funds_raised<40000)][['total_laid_off',
'funds_raised']]

```

```

fig = plt.figure(figsize=(15,10))
sns.regplot(data=df, x=funds_raised.total_laid_off,
y=funds_raised.funds_raised)
plt.title("Mối quan hệ giữa total layoffs và funds raised của các công
ty")
plt.show()

```

Dựa vào biểu đồ Mối quan hệ của total layoffs và funds raised:

- Đa số các công ty có số funds raised vừa và ít thì cũng có total layoffs vừa và thấp.
- Những công ty có total laidoff cao, thậm chí là rất cao có funds raised rất thấp (gần bằng 0).
- Ngược lại, những công ty có funds raised cao thì thường sẽ có total laidoff thấp.

Số lượng sa thải trên thế giới giai đoạn 2020 - 2023

group the data by country and industry and sum the total_laid_off column

```
grouped = df.groupby(['country']).agg({'total_laid_off':
'sum'}).reset_index()
```

create a choropleth map using Plotly

```
fig = px.choropleth(grouped, locations='country',
locationmode='country names', color='total_laid_off',
# color_continuous_scale='OrRd',
color_continuous_scale='YlOrRd',
hover_name='country', projection='natural earth',
title='Number of People Laid Off by Country')
```

show the map

```
fig.show()
```

```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"coloraxis":"coloraxis","geo":"geo","hovertemplate":"<b>%
{hovertext}</b><br><br>country=%{location}<br>total_laid_off=%
{z}<extra></extra>","hovertext":
```

```
[ "ARGENTINA", "AUSTRALIA", "AUSTRIA", "BRAZIL", "BULGARIA", "CANADA", "CHILE",
  ", "CHINA", "COLOMBIA", "DENMARK", "EGYPT", "ESTONIA", "FINLAND", "FRANCE", "GERMANY", "HONG KONG",
  "HUNGARY", "INDIA", "INDONESIA", "IRELAND", "ISRAEL", "ITALY", "JAPAN", "KENYA", "LITHUANIA",
  "LUXEMBOURG", "MALAYSIA", "MEXICO", "MYANMAR", "NETHERLANDS", "NEW ZEALAND", "NIGERIA",
  "NORWAY", "PAKISTAN", "POLAND", "PORTUGAL", "ROMANIA", "RUSSIA", "SENEGAL", "SEYCHELLES",
  "SINGAPORE", "SOUTH KOREA", "SPAIN", "SWEDEN", "SWITZERLAND", "THAILAND", "TURKEY", "UNITED ARAB EMIRATES",
  "UNITED KINGDOM", "UNITED STATES", "VIETNAM"], "locationmode": "country names", "locations":
[ "ARGENTINA", "AUSTRALIA", "AUSTRIA", "BRAZIL", "BULGARIA", "CANADA", "CHILE",
  ", "CHINA", "COLOMBIA", "DENMARK", "EGYPT", "ESTONIA", "FINLAND", "FRANCE", "GERMANY", "HONG KONG",
  "HUNGARY", "INDIA", "INDONESIA", "IRELAND", "ISRAEL", "ITALY", "JAPAN", "KENYA", "LITHUANIA",
  "LUXEMBOURG", "MALAYSIA", "MEXICO", "MYANMAR", "NETHERLANDS", "NEW ZEALAND", "NIGERIA",
  "NORWAY", "PAKISTAN", "POLAND", "PORTUGAL", "ROMANIA", "RUSSIA", "SENEGAL", "SEYCHELLES",
  "SINGAPORE", "SOUTH KOREA", "SPAIN", "SWEDEN", "SWITZERLAND", "THAILAND", "TURKEY", "UNITED ARAB EMIRATES",
  "UNITED KINGDOM", "UNITED STATES", "VIETNAM"], "name": "", "type": "choropleth", "z":
[323,2324,570,10391,120,6319,30,5905,130,240,0,333,250,915,8701,730,0,35953,3521,257,3638,0,85,349,60,45,100,270,200,17220,45,1882,140,0,25,115,80,400,300,75,5995,0,250,11264,62,55,0,995,6398,255777,0]], "layout":
{"coloraxis": {"colorbar": {"title": {"text": "total_laid_off"}, "colorscale": [[0, "rgb(255,255,204)"], [0.125, "rgb(255,237,160)"], [0.25, "rgb(254,217,118)"], [0.375, "rgb(254,178,76)"], [0.5, "rgb(253,141,60)"], [0.625, "rgb(252,78,42)"], [0.75, "rgb(227,26,28)"], [0.875, "rgb(189,0,38)"], [1, "rgb(128,0,38)"]], "geo": {"center": {}, "domain": {"x": [0,1], "y": [0,1]}, "projection": {"type": "natural earth"}}, "legend": {"tracegroupgap": 0}, "template": {"data": {"bar": [{"error_x": {"color": "#2a3f5f"}, "error_y": {"color": "#2a3f5f"}, "marker": {"line": {"color": "#E5ECF6", "width": 0.5}, "pattern": {"fillmode": "overlay", "size": 10, "solidity": 0.2}}, "type": "bar"}], "barpolar": [{"marker": {"line": {"color": "#E5ECF6", "width": 0.5}, "pattern": {"fillmode": "overlay", "size": 10, "solidity": 0.2}}, "type": "barpolar"}], "carpet": [{"aaxis": {"endlinecolor": "#2a3f5f", "gridcolor": "white", "linecolor": "white", "minorgridcolor": "white", "startlinecolor": "#2a3f5f"}, "baxis": {"endlinecolor": "#2a3f5f", "gridcolor": "white", "linecolor": "white", "minorgridcolor": "white", "startlinecolor": "#2a3f5f"}, "type": "carpet"}], "choropleth": [{"colorbar": {"outlinewidth": 0, "ticks": ""}, "type": "choropleth"}], "contour": [{"colorbar": {"outlinewidth": 0, "ticks": ""}, "colorscale": [[0, "#0d0887"], [0.1111111111111111, "#46039f"], [0.2222222222222222, "#7201a8"], [0.3333333333333333, "#9c179e"], [0.4444444444444444, "#bd3786"], [0.5555555555555556, "#d8576b"]],
```

```

[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]], "type": "contour"}], "contourcarpet": [{"colorbar":
{"linewidth":0, "ticks":""}, "type": "contourcarpet"}], "heatmap":
[{"colorbar": {"linewidth":0, "ticks":""}, "colorscale":
[[0, "#0d0887"], [0.1111111111111111, "#46039f"],
[0.2222222222222222, "#7201a8"], [0.3333333333333333, "#9c179e"],
[0.4444444444444444, "#bd3786"], [0.5555555555555556, "#d8576b"],
[0.6666666666666666, "#ed7953"], [0.7777777777777778, "#fb9f3a"],
[0.8888888888888888, "#fdca26"],
[1, "#f0f921"]], "type": "heatmap"}], "heatmapgl": [{"colorbar":
{"linewidth":0, "ticks":""}, "colorscale": [[0, "#0d0887"],
[0.1111111111111111, "#46039f"], [0.2222222222222222, "#7201a8"],
[0.3333333333333333, "#9c179e"], [0.4444444444444444, "#bd3786"],
[0.5555555555555556, "#d8576b"], [0.6666666666666666, "#ed7953"],
[0.7777777777777778, "#fb9f3a"], [0.8888888888888888, "#fdca26"],
[1, "#f0f921"]], "type": "heatmapgl"}], "histogram": [{"marker": {"pattern":
{"fillmode": "overlay", "size": 10, "solidity": 0.2}}, "type": "histogram"}],
"histogram2d": [{"colorbar": {"linewidth":0, "ticks":""}, "colorscale":
[[0, "#0d0887"], [0.1111111111111111, "#46039f"],
[0.2222222222222222, "#7201a8"], [0.3333333333333333, "#9c179e"],
[0.4444444444444444, "#bd3786"], [0.5555555555555556, "#d8576b"],
[0.6666666666666666, "#ed7953"], [0.7777777777777778, "#fb9f3a"],
[0.8888888888888888, "#fdca26"],
[1, "#f0f921"]], "type": "histogram2d"}], "histogram2dcontour":
[{"colorbar": {"linewidth":0, "ticks":""}, "colorscale":
[[0, "#0d0887"], [0.1111111111111111, "#46039f"],
[0.2222222222222222, "#7201a8"], [0.3333333333333333, "#9c179e"],
[0.4444444444444444, "#bd3786"], [0.5555555555555556, "#d8576b"],
[0.6666666666666666, "#ed7953"], [0.7777777777777778, "#fb9f3a"],
[0.8888888888888888, "#fdca26"],
[1, "#f0f921"]], "type": "histogram2dcontour"}], "mesh3d": [{"colorbar":
{"linewidth":0, "ticks":""}, "type": "mesh3d"}], "parcoords": [{"line":
{"colorbar": {"linewidth":0, "ticks":""}, "type": "parcoords"}], "pie":
[{"automargin": true, "type": "pie"}], "scatter": [{"fillpattern":
{"fillmode": "overlay", "size": 10, "solidity": 0.2}, "type": "scatter"}], "sc
atter3d": [{"line": {"colorbar": {"linewidth":0, "ticks":""}, "marker":
{"colorbar":
{"linewidth":0, "ticks":""}, "type": "scatter3d"}], "scattercarpet":
[{"marker": {"colorbar":
{"linewidth":0, "ticks":""}, "type": "scattercarpet"}], "scattergeo":
[{"marker": {"colorbar":
{"linewidth":0, "ticks":""}, "type": "scattergeo"}], "scattergl":
[{"marker": {"colorbar":
{"linewidth":0, "ticks":""}, "type": "scattergl"}], "scattermapbox":
[{"marker": {"colorbar":
{"linewidth":0, "ticks":""}, "type": "scattermapbox"}], "scatterpolar":
[{"marker": {"colorbar":
{"linewidth":0, "ticks":""}, "type": "scatterpolar"}], "scatterpolargl":
[{"marker": {"colorbar":

```

```

{"outlinewidth":0,"ticks":"","type":"scatterpolargl"}], "scatterternary": [{"marker": {"colorbar": {"outlinewidth":0,"ticks":"","type":"scatterternary"}}, "surface": [{"colorbar": {"outlinewidth":0,"ticks":"","colorscale": [[0, "#0d0887"], [0.1111111111111111, "#46039f"], [0.2222222222222222, "#7201a8"], [0.3333333333333333, "#9c179e"], [0.4444444444444444, "#bd3786"], [0.5555555555555556, "#d8576b"], [0.6666666666666666, "#ed7953"], [0.7777777777777778, "#fb9f3a"], [0.8888888888888888, "#fdca26"], [1, "#f0f921"]]}, "type": "surface"}], "table": [{"cells": {"fill": {"color": "#EBF0F8"}, "line": {"color": "white"}}, "header": {"fill": {"color": "#C8D4E3"}, "line": {"color": "white"}}, "type": "table"}]}, "layout": {"annotationdefaults": {"arrowcolor": "#2a3f5f", "arrowhead": 0, "arrowwidth": 1}, "autotypenumbers": "strict", "coloraxis": {"colorbar": {"outlinewidth":0,"ticks":"","colorscale": {"diverging": [[0, "#8e0152"], [0.1, "#c51b7d"], [0.2, "#de77ae"], [0.3, "#f1b6da"], [0.4, "#fde0ef"], [0.5, "#f7f7f7"], [0.6, "#e6f5d0"], [0.7, "#b8e186"], [0.8, "#7fbcb4"], [0.9, "#4d9221"], [1, "#276419"]]}, "sequential": [[0, "#0d0887"], [0.1111111111111111, "#46039f"], [0.2222222222222222, "#7201a8"], [0.3333333333333333, "#9c179e"], [0.4444444444444444, "#bd3786"], [0.5555555555555556, "#d8576b"], [0.6666666666666666, "#ed7953"], [0.7777777777777778, "#fb9f3a"], [0.8888888888888888, "#fdca26"], [1, "#f0f921"]]}, "sequentialminus": [[0, "#0d0887"], [0.1111111111111111, "#46039f"], [0.2222222222222222, "#7201a8"], [0.3333333333333333, "#9c179e"], [0.4444444444444444, "#bd3786"], [0.5555555555555556, "#d8576b"], [0.6666666666666666, "#ed7953"], [0.7777777777777778, "#fb9f3a"], [0.8888888888888888, "#fdca26"], [1, "#f0f921"]]}, "colorway": ["#636efa", "#EF553B", "#00cc96", "#ab63fa", "#FFA15A", "#19d3f3", "#FF6692", "#B6E880", "#FF97FF", "#FECB52"]}, "font": {"color": "#2a3f5f"}, "geo": {"bgcolor": "white", "lakecolor": "white", "landcolor": "#E5ECF6", "showlakes": true, "showland": true, "subunitcolor": "white"}, "hoverlabel": {"align": "left", "hovermode": "closest", "mapbox": {"style": "light"}, "paper_bgcolor": "white", "plot_bgcolor": "#E5ECF6", "polar": {"angularaxis": {"gridcolor": "white", "linecolor": "white", "ticks": ""}, "bgcolor": "#E5ECF6", "radialaxis": {"gridcolor": "white", "linecolor": "white", "ticks": ""}}, "scene": {"xaxis": {"backgroundcolor": "#E5ECF6", "gridcolor": "white", "gridwidth": 2, "linecolor": "white", "showbackground": true, "ticks": "", "zerolinecolor": "white"}, "yaxis": {"backgroundcolor": "#E5ECF6", "gridcolor": "white", "gridwidth": 2, "linecolor": "white", "showbackground": true, "ticks": "", "zerolinecolor": "white"}, "zaxis": {"backgroundcolor": "#E5ECF6", "gridcolor": "white", "gridwidth": 2, "linecolor": "white", "showbackground": true, "ticks": "", "zerolinecolor": "white"}}, "shapedefaults": {"line": {"color": "#2a3f5f"}}, "ternary": {"aaxis": {"gridcolor": "white", "linecolor": "white", "ticks": ""}, "baxis":

```

```
{
  "gridcolor": "white", "linecolor": "white", "ticks": "", "bgcolor": "#E5ECF6", "caxis": {
    "gridcolor": "white", "linecolor": "white", "ticks": ""
  }, "title": {
    "x": 5.0e-2, "xaxis": {
      "automargin": true, "gridcolor": "white", "linecolor": "white", "ticks": "", "title": {
        "standoff": 15, "zerolinecolor": "white", "zerolinewidth": 2, "yaxis": {
          "automargin": true, "gridcolor": "white", "linecolor": "white", "ticks": "", "title": {
            "standoff": 15, "zerolinecolor": "white", "zerolinewidth": 2
          }
        }
      }
    }
  }, "text": "Number of People Laid Off by Country"
}
```

Biểu đồ trên thể hiện tình hình người bị sa thải cụ thể ở các đất nước trên thế giới. Cụ thể:

- Đất nước đang đứng dưới nguy cơ lớn về việc chuyển đổi nhân sự là Mỹ với thống kê khoảng 250.000 người mất việc.
- Các đất nước khác thì có tỉ lệ ít đáng kể so với Mỹ, như với Ấn Độ với khoảng 35.953 người mất việc, Brazil 10.391 người, Canada 6.319 người, các đất nước còn lại với con số không đáng kể.

Vì số lượng ngành nghề khá nhiều để visualize, nên ta sẽ nhóm các ngành nghề thành các nhóm ngành

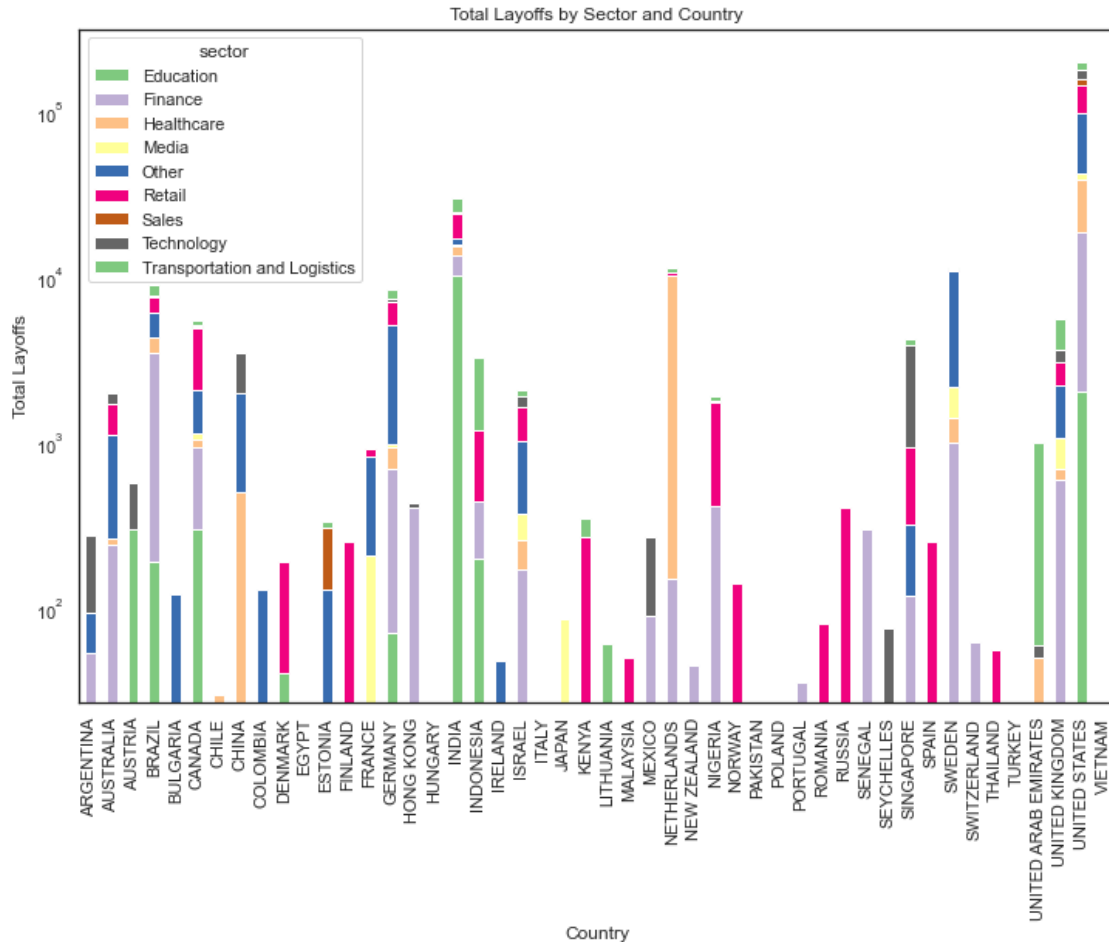
Mapping industries into groups

```
industry_to_sector = {
  'CRYPTO': 'Technology',
  'DATA': 'Technology',
  'HARDWARE': 'Technology',
  'SOFTWARE': 'Technology',
  'HEALTHCARE': 'Healthcare',
  'FITNESS': 'Healthcare',
  'RETAIL': 'Retail',
  'FOOD': 'Retail',
  'FINANCE': 'Finance',
  'EDUCATION': 'Education',
  'MEDIA': 'Media',
  'TRANSPORTATION': 'Transportation and Logistics',
  'LOGISTICS': 'Transportation and Logistics',
  'INFRASTRUCTURE': 'Other',
  'PRODUCT': 'Other',
  'CONSTRUCTION': 'Other',
  'OTHER': 'Other',
  'LEGAL': 'Other',
  'REAL ESTATE': 'Other',
  'HR': 'Other',
  'RECRUITING': 'Other',
  'SALES': 'Sales',
  'AEROSPACE': 'Other',
}
```

```
    'SECURITY': 'Other'  
}
```

Tình hình sa thải trong các ngành ở các quốc gia

```
df_sub=df.copy()  
df_sub['sector'] = df_sub['industry'].map(industry_to_sector)  
  
# Group the data by country and industry and calculate the total  
# number of layoffs  
grouped = df_sub.groupby(['country', 'sector'])  
['total_laid_off'].sum()  
  
# Reshape the data to create a pivot table with countries as rows,  
# industries as columns, and total layoffs as values  
pivoted = grouped.reset_index().pivot(index='country',  
columns='sector', values='total_laid_off')  
  
# Create the stacked bar chart  
fig, ax = plt.subplots(figsize=(12, 8))  
pivoted.plot(kind='bar', stacked=True, ax=ax)  
  
# Set the y-axis scale to a log scale  
ax.set_yscale('log')  
  
# Set the title and axis labels  
plt.title('Total Layoffs by Sector and Country')  
plt.xlabel('Country')  
plt.ylabel('Total Layoffs')  
  
# Show the plot  
plt.show()
```



Nhóm các ngành nghề có liên quan thành các lĩnh vực: **Technology, Healthcare, Retail, Finance, Education, Media, Transportation and Logistics, Other, Sales**. Biểu diễn bằng biểu đồ cột chồng cho thấy tỉ lệ các lĩnh vực xuất hiện sa thải ở từng quốc gia, *biểu thị đầy đủ về các lĩnh vực* đang được tác động trong một đất nước như thế nào. Các yếu tố đáng chú ý về tỷ lệ sa thải có thể bao gồm:

- Ấn Độ là quốc gia gặp phải vấn đề sa thải nghiêm trọng nhất liên quan đến giáo dục.
- Netherlands chỉ gặp vấn đề ở 2 lĩnh vực chính là Healthcare và Finance.
- Lĩnh vực Finance có nguy cơ bị sa thải cao nhất và bị ảnh hưởng tại hầu hết các quốc gia.

Các quốc gia đang dẫn đầu trong làn sóng sa thải ở từng lĩnh vực

```
# Map industries to sectors in the DataFrame
df['sector'] = df['industry'].map(industry_to_sector)
```

```
# Group the data by sector and country
grouped = df.groupby(['sector', 'country'])
['total_laid_off'].sum().reset_index()
```



```

fig= plt.figure(figsize=(12,13))
# Loop through the sectors and create a bar chart for each one
for index, sector in enumerate(grouped['sector'].unique()):

    # Filter the data for the current sector
    sector_data = grouped[grouped['sector'] == sector]

    # Sort the data by total layoffs for each country in the sector
    sorted_data = sector_data.sort_values('total_laid_off',
ascending=False)

    # Filter to include only the top 5 countries
    top_5 = sorted_data.head(5)

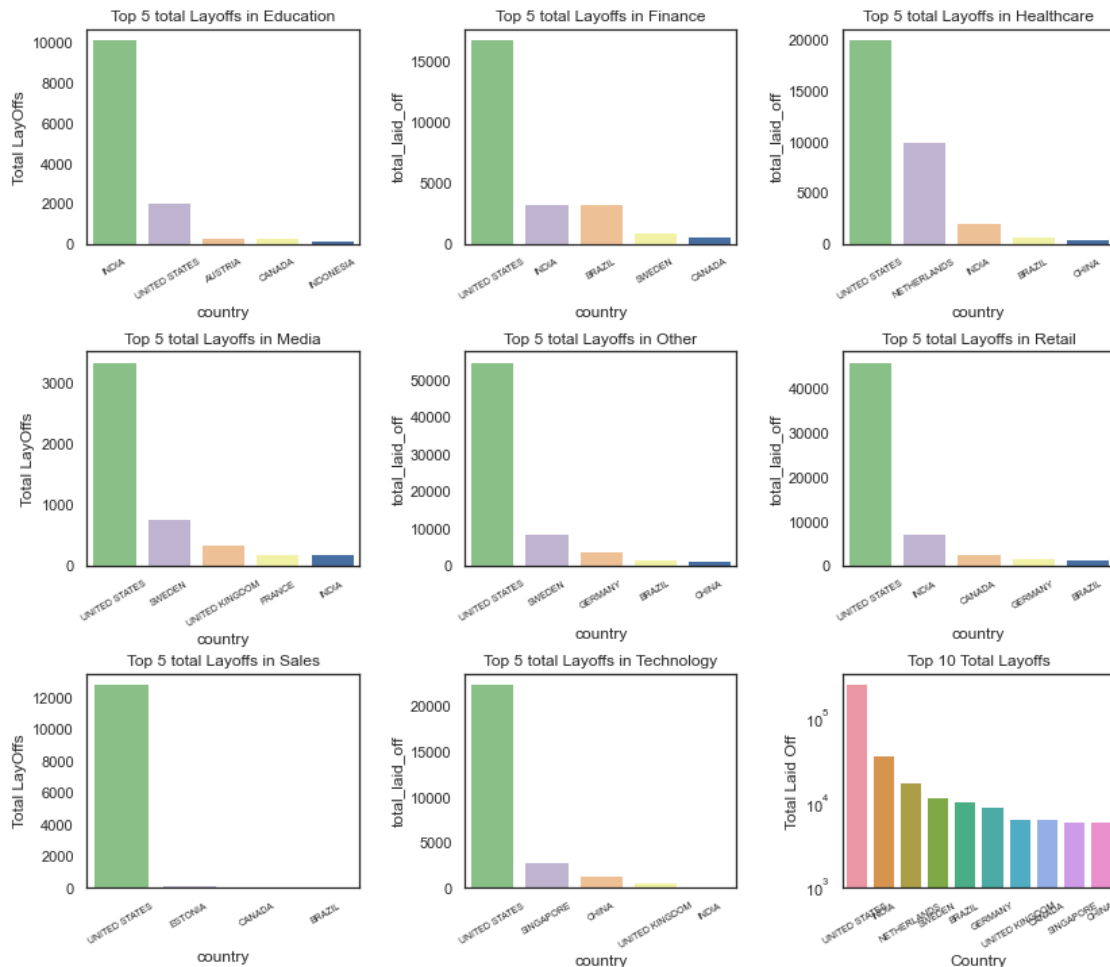
    # Create the bar chart
    ax = fig.add_subplot(4,3,index+1)
    ax = sns.barplot(data=top_5,
                    x= "country",
                    y='total_laid_off')
    # ax.bar(top_5['country'], top_5['total_laid_off'])
    ax.set_title(f'Top 5 total Layoffs in {sector}')
    if index % 3 ==0 : ax.set_ylabel('Total LayOffs')
    ax.tick_params(axis='x')
    plt.xticks(fontsize=7,rotation=30)


top_n_countries = 10
# create a chart that ranks countries based on their total layoffs in
all industries
all_data = df.groupby('country')
['total_laid_off'].sum().nlargest(top_n_countries)


# fig.subplots_adjust(left=0.1, bottom=0.2)
sns.barplot(data=all_data.to_frame().reset_index(),x="country", y
="total_laid_off")
# ax.bar(all_data.index, all_data.values)
ax.set_title(f'Top {top_n_countries} Total Layoffs')
ax.set_xlabel('Country')
ax.set_ylabel('Total Laid Off')
ax.set_yscale('log')


# add padding between subplots and adjust layout
plt.tight_layout(pad=2)
# add margin between rows
fig.subplots_adjust(hspace=0.5)
plt.show()

```

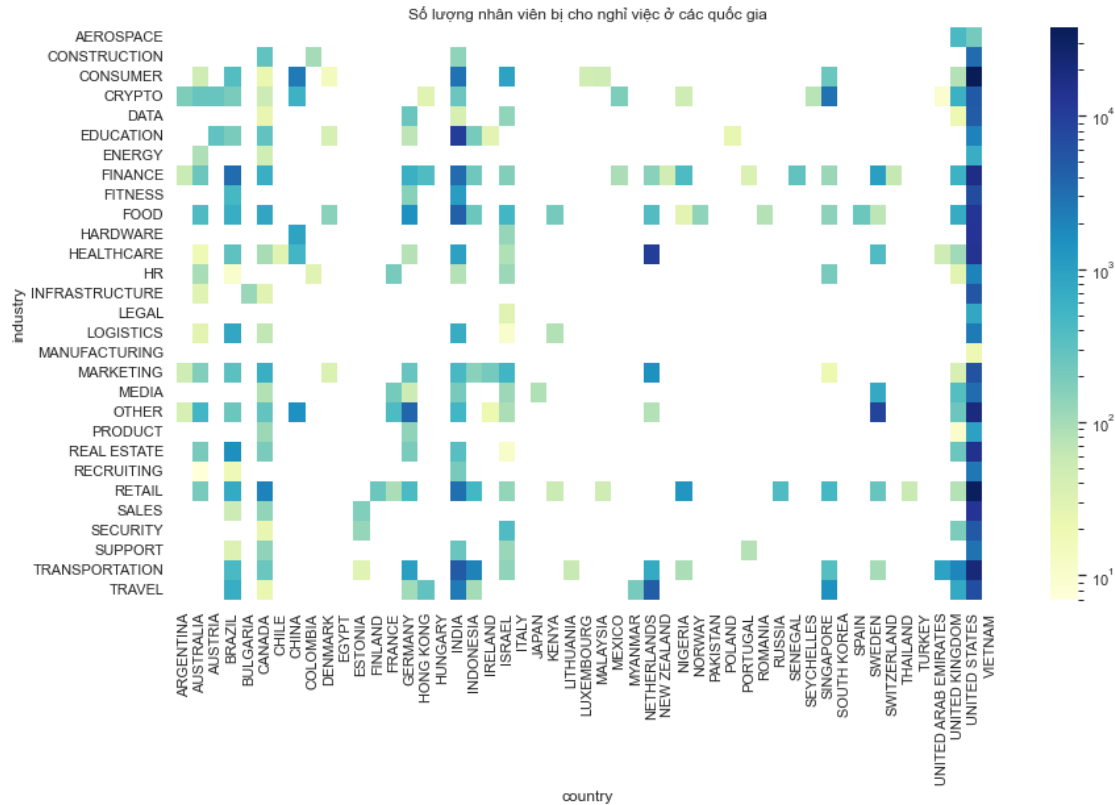



Từ những nhóm lĩnh vực mà ta đã chia, ta có các biểu đồ cột thể hiện những quốc gia với số lượng sa thải nhân viên **đạt top đầu**. Dễ thấy Mỹ là quốc gia đứng đầu trong tất cả các bảng cho thấy tỉ lệ sa thải ở đất nước này là **đáng báo động**. Đặc biệt đáng quan tâm về tỉ lệ bị sa thải của trong các ngành công nghệ liên quan tới Sales, Retail, Technology.

```
plt.figure(figsize=(14, 8))
pivot_df = pd.pivot_table(df, values='total_laid_off',
columns=['country'], index=['industry'], aggfunc='sum')
# sns.heatmap(pivot_df, cmap='YlGnBu')
```

```
sns.heatmap(pivot_df, cmap='YlGnBu', norm=colors.LogNorm())
plt.title('Số lượng nhân viên bị cho nghỉ việc ở các quốc gia')
```

```
Text(0.5, 1.0, 'Số lượng nhân viên bị cho nghỉ việc ở các quốc gia')
```



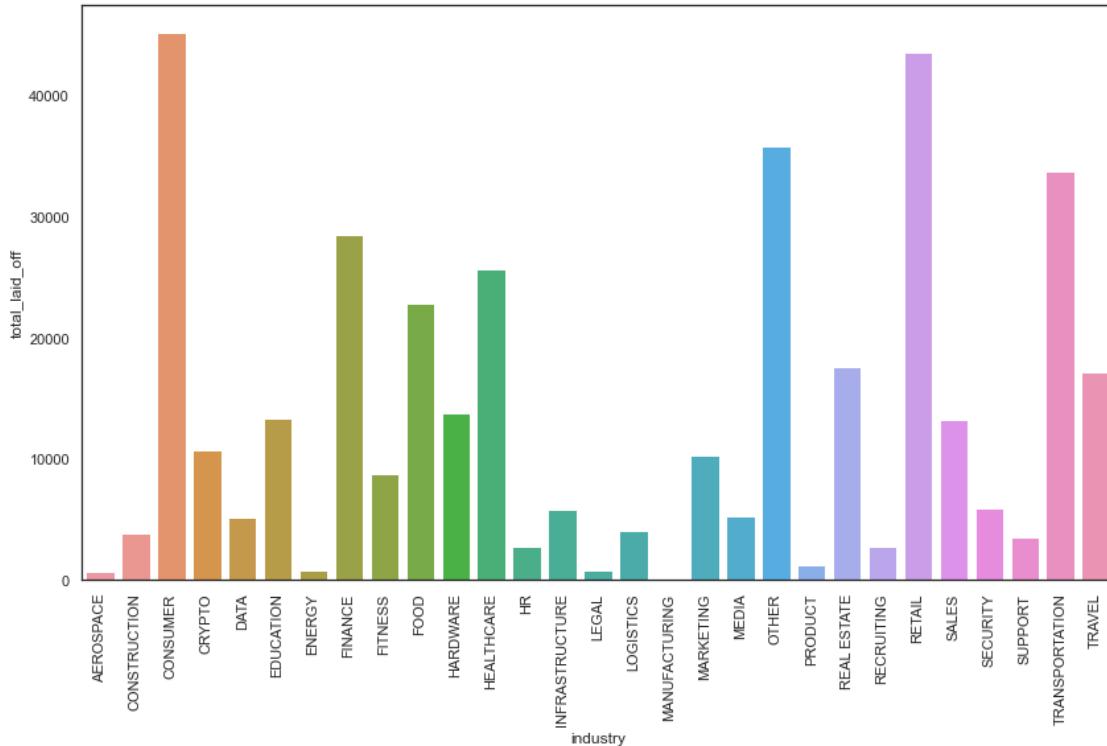
Heat map trực quan số lượng nhân viên bị cho nghỉ việc ở các quốc gia. Mỹ đứng đầu ở hầu hết mọi lĩnh vực. Tiếp đến là Ấn Độ với các ngành như: Consumer, Education, Food, Finance, Transportation.

Ngành nào đang có số lượng layoff cao nhất trên thế giới

```
industry_grouped = df.groupby(["industry"])
["total_laid_off"].sum().reset_index()
```

```
plt.figure(figsize=(14, 8))
plt.xticks(rotation=90)
sns.barplot(industry_grouped, x="industry", y="total_laid_off",
errorbar=None)
```

```
<AxesSubplot: xlabel='industry', ylabel='total_laid_off'>
```



Ở thời điểm hiện tại xét trên những mảng của các công ty công nghệ, nhận thấy lần lượt là các công ty chuyên về Customer và Rental đang có lượng người bị sa thải nhiều nhất. Cụ thể:

- Customer company: chỉ một công ty tập trung chủ yếu vào khách hàng của mình và đặt khách hàng là trung tâm của mọi hoạt động kinh doanh của mình.
- Rental company: là một công ty hoạt động trong lĩnh vực cho thuê các sản phẩm hoặc dịch vụ cho khách hàng.

Một số nguyên nhân chính của nó là do: sự cạnh tranh, nhu cầu thị trường thay đổi, tác động của đại dịch COVID-19. Những điều này làm cho kinh tế bị suy thoái dẫn đến cắt giảm nguồn nhân sự công ty.

Xét tất cả các ngành, dùng biểu đồ cột thể hiện số lượng quốc gia có ngành nghề đó layoff top đầu / cuối.

Các bước thực hiện:

- Ta gom nhóm các Industry, sau đó đếm số lượng country laid off (có unique để đảm bảo 1 country không được đếm quá 1 lần)
- dữ liệu trong df đã thông qua xử lý, chỉ có các trường hợp laid off

Nhận xét:

- Industry có số lượng country laid off cao nhất là finance (số lượng: 20) và food (số lượng: 20)

- Industry có số lượng country laid off thấp nhất là manufacturing (số lượng: 1)

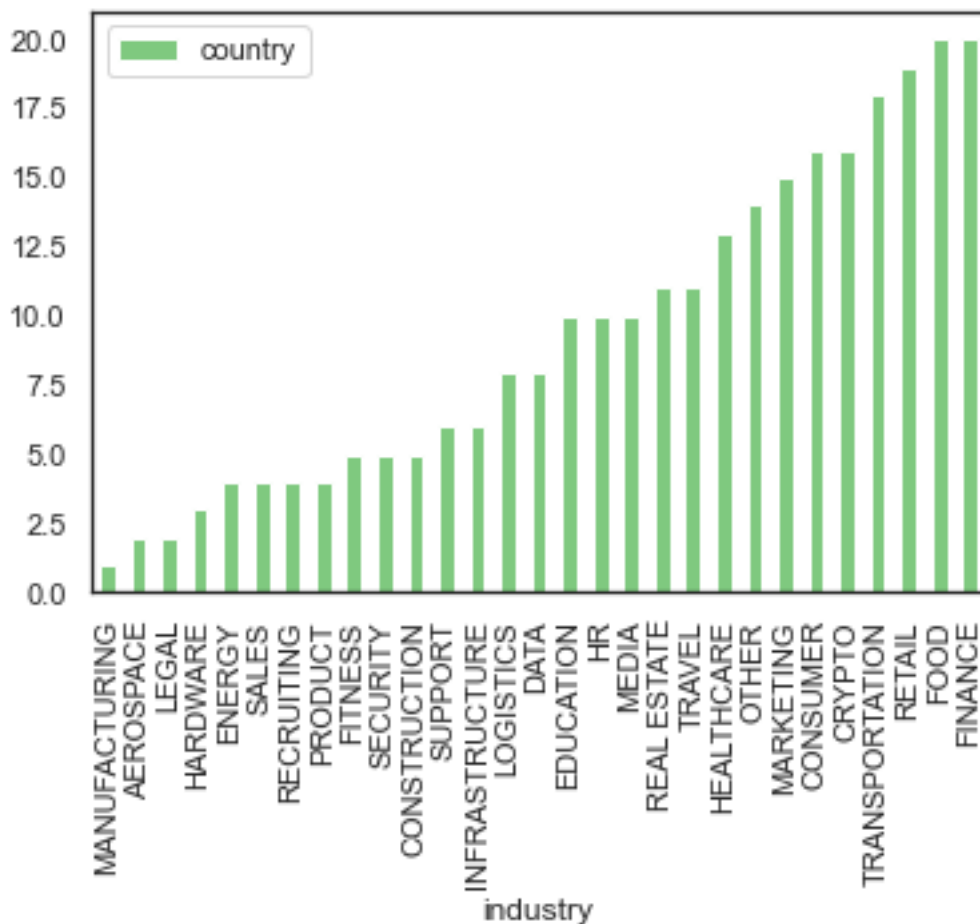
```
df_industry = df.groupby("industry").agg({"country":
pd.Series.nunique}).sort_values("country")

print("industry max:
",df_industry[df_industry["country"]==df_industry["country"].max()],
"\n")
print("industry min:
",df_industry[df_industry["country"]==df_industry["country"].min()],
"\n")

ax = df_industry.plot.bar()
ax.set_xlabel="industry"
ax.set_ylabel="total countries"
ax.set_title="Total laid off group by industry"
```

industry max:	country
industry	
FOOD	20
FINANCE	20

industry min:	country
industry	
MANUFACTURING	1



Thêm `total_employee` để chọn ra top company theo quy mô nhân sự. Từ đây phát triển các phân tích trên nhóm top company.

`n=10`

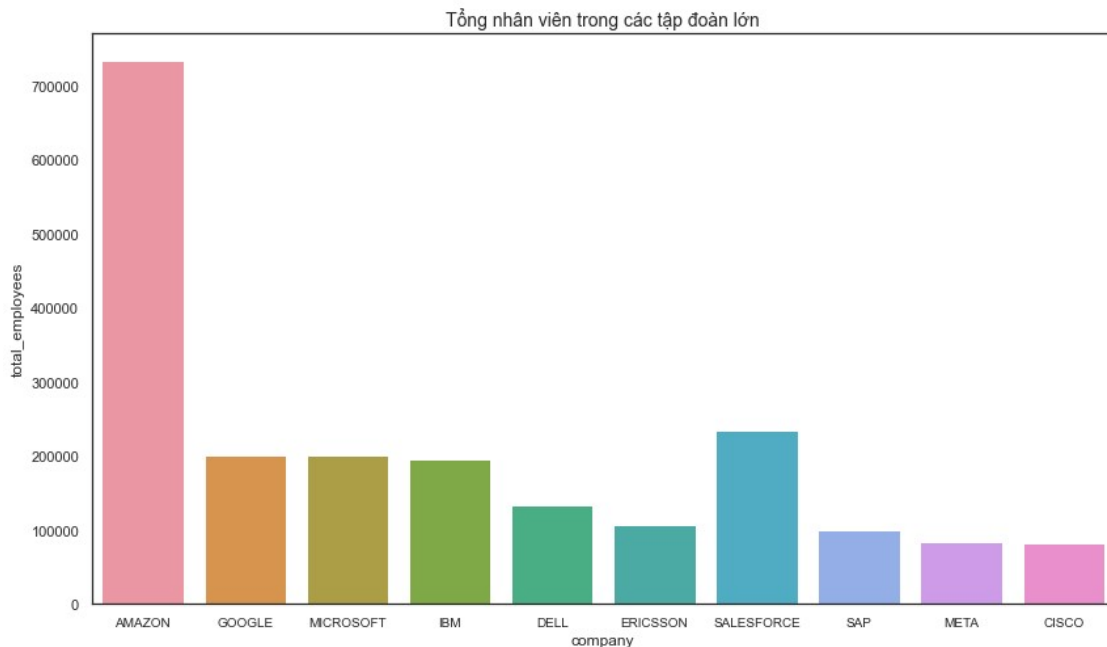
```
top_n_biggest_companies
=list(df_with_total_employees.groupby("company")\
      .agg({"total_employees": "max"})\
      .sort_values("total_employees",ascending=False)[:n]\
      .index)
```

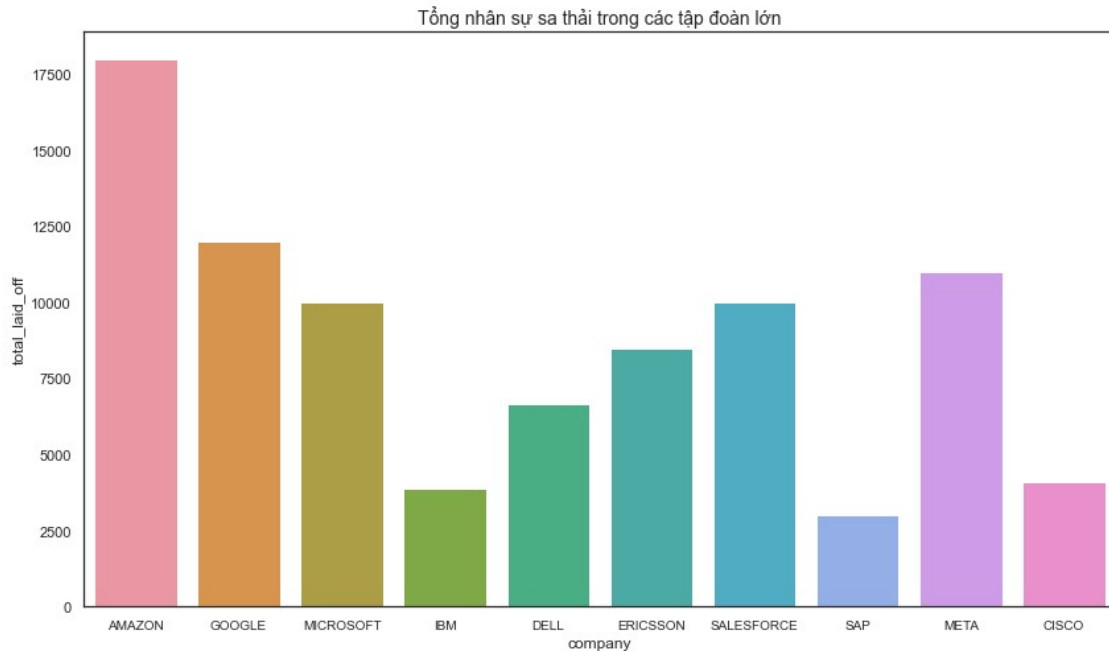
```
plt.figure(figsize=(14, 8))
plt.xticks(size=10)
plt.title("Tổng nhân viên trong các tập đoàn lớn",size=14)
sns.barplot(df_with_total_employees[df_with_total_employees['company']
.isin(top_n_biggest_companies)],
            x="company",
            y="total_employees",
            errorbar=None,
```

```
estimator="sum")
```

```
plt.figure(figsize=(14, 8))
plt.xticks(size=10)
plt.title("Tổng nhân sự sa thải trong các tập đoàn lớn",size=14)
sns.barplot(df_with_total_employees[df_with_total_employees['company']
.isin(top_n_biggest_companies)],
            x="company",
            y="total_laid_off",
            errorbar=None,
            estimator="sum")
```

<AxesSubplot: title={'center': 'Tổng nhân sự sa thải trong các tập đoàn lớn'}, xlabel='company', ylabel='total_laid_off'>





```
df_with_total_employees.describe()['total_laid_off']
```

```
count      1189.000000
mean        272.983179
std         883.553635
min           3.000000
25%         38.000000
50%         80.000000
75%        200.000000
max       12000.000000
Name: total_laid_off, dtype: float64
```

Trong top 10 công ty lớn, số lượng sa thải đạt thấp nhất là khoảng 2200 người. Nhận thấy được là các tập đoàn lớn có xu hướng sa thải nhân viên cao hơn mức bình quân của toàn bộ các công ty trong tập dữ liệu (272 người).

Khám phá dữ liệu thực tế

Địa điểm tiêu biểu nơi xảy ra layoff

```
plt.figure(facecolor='k')
wordcloud = WordCloud(width=400,
height=200,max_font_size=100).generate(' '.join(df['location']))

plt.imshow(wordcloud,interpolation="bilinear")
plt.axis("off")
plt.tight_layout(pad=0)
plt.show()
```




San Francisco area bay với Silicon Valley nổi tiếng lại cũng là nơi diễn ra layoff nhiều nhất :(

Số lượng layoff theo thời gian

```
df_clean_total_lo=df[df['total_laid_off']>0]
df_clean_total_lo['date'].min()
df_clean_total_lo['date'].max()
df_clean_total_lo['month_year'] =
pd.to_datetime(df_clean_total_lo['date']).dt.strftime('%Y-%m')
total_by_month_year = df_clean_total_lo.groupby('month_year').sum()
['total_laid_off']
total_by_month_year.reindex(pd.period_range('2020-
01',total_by_month_year.index[-1],freq='M'))
total_by_month_year.fillna(0.0)
total_by_month_year.max()
```

84424.0

```
df_total_employees_time=df_with_total_employees[df_with_total_employee
s.total_laid_off>0]
df_total_employees_time['year']=df_total_employees_time['date'].apply(
lambda x :x.year)
df_total_employees_time['month']=df_total_employees_time['date'].apply
(lambda x :x.month)
df_total_employees_time['month_year'] =
pd.to_datetime(df_total_employees_time['date']).dt.strftime('%Y-%m')
```

```
total_by_time=df_total_employees_time.groupby('month_year').sum()
['total_laid_off']
```

```
fig, ax = plt.subplots( figsize=(12, 7), tight_layout=True)
fig.tight_layout(pad=5.0)
```

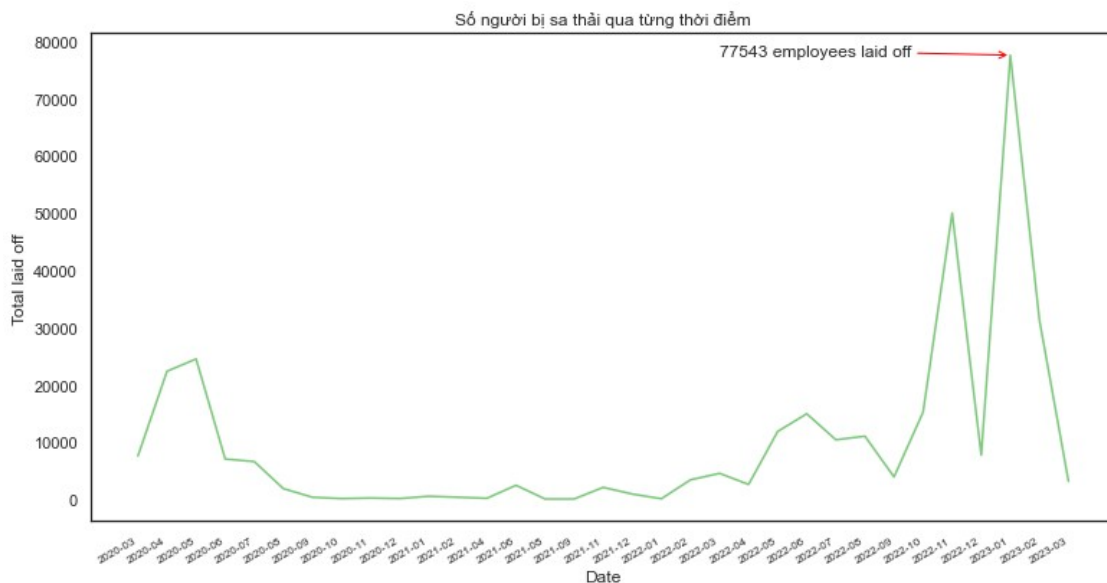
```
plt.plot(total_by_time.index,total_by_time.values)
plt.xlabel('Date')
plt.ylabel('Total laid off')
plt.title('Số người bị sa thải qua từng thời điểm')

x_max=np.where(total_by_time == total_by_time.max())[0]
y_max=total_by_time.max()
plt.annotate(f'{int(y_max)} employees laid off', xy=(x_max, y_max),
xytext=(x_max-10, y_max),arrowprops=dict(arrowstyle='->',color='red'))

# beautify the x-labels
plt.gcf().autofmt_xdate()
plt.xticks(fontsize=8)
```

```
print(f"Số lượng lay off nhiều nhất trong 1 tháng: {y_max} người")
```

Số lượng lay off nhiều nhất trong 1 tháng: 77543.0 người



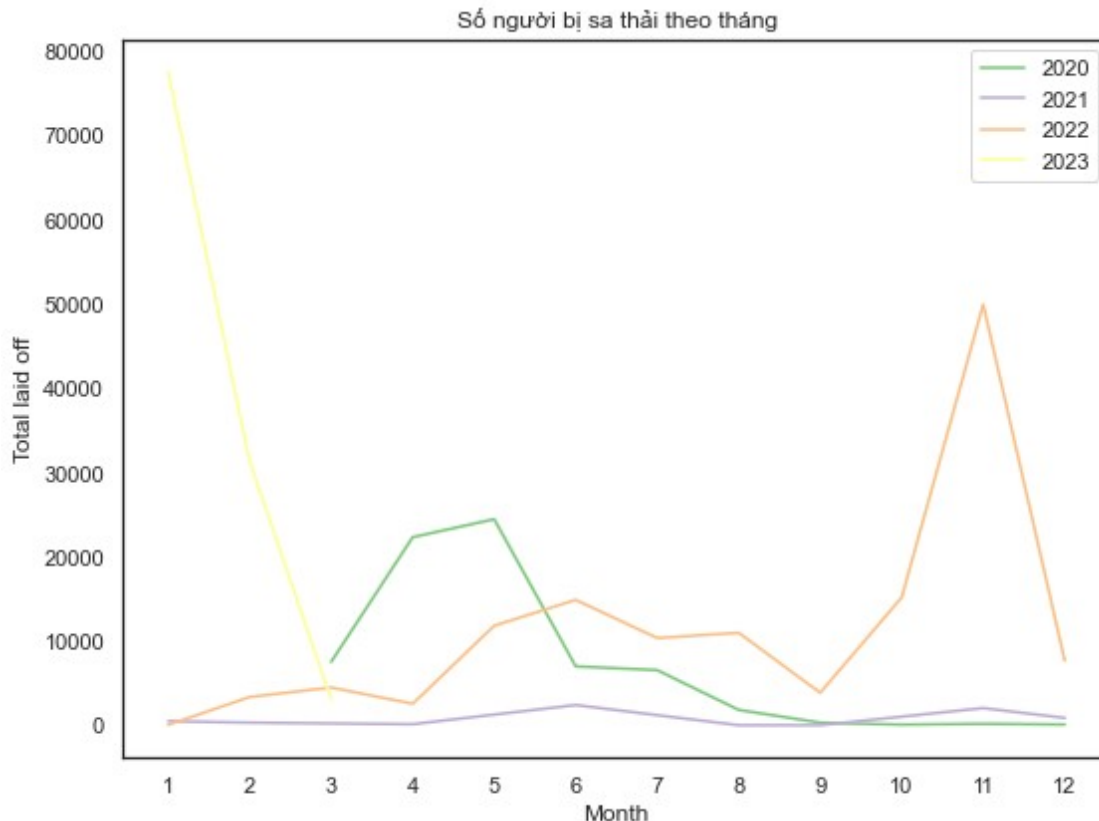
Làn sóng layoff diễn ra mạnh mẽ nhất từ tháng 9 2022 và đạt đỉnh vào tháng 1 2023 với tổng cộng gần 80 ngàn người bị sa thải trong 1 tháng.

Số lượng layoff theo tháng

```
analysis_year = df_total_employees_time.groupby(["year"],sort=True)
fig, ax = plt.subplots( figsize=(9, 7), tight_layout=True)
fig.tight_layout(pad=5.0)
for year, dfx in analysis_year:
    df_m=dfx.groupby('month').sum()['total_laid_off']
    plt.plot(df_m.index,df_m.values,label=f'{year}')
```

```
plt.xticks(np.arange(1, 13, 1))
plt.xlabel('Month')
plt.ylabel('Total laid off')
plt.title('Số người bị sa thải theo tháng')
plt.legend()
```

<matplotlib.legend.Legend at 0x223eb885e20>



Tháng 11 và tháng 1 là hai thời điểm có nhiều người bị sa thải nhất. Nhìn rộng hơn thì quý 1 và quý 4 là những thời điểm có lượng sa thải lớn nhất.

Liệu có đợt layoff này diễn ra vào holiday week?

holiday week: Tuần lễ cuối cùng của năm gồm cả Christmas (25/12) và ngày đầu năm mới (01/01). Đây là dịp lễ lớn ở các nước phương tây.

```
df_time=df.set_index('date',drop=True)
holiday_laid_off_2021=df_time['2020-12-24':'2021-01-01']
holiday_laid_off_2022=df_time['2021-12-24':'2022-01-01']
holiday_laid_off_2023=df_time['2022-12-24':'2023-01-01']

print(holiday_laid_off_2021)
```

Empty DataFrame

Columns: [company, location, industry, total_laid_off,

```
percentage_laid_off, stage, country, funds_raised, sector]
Index: []
```

```
print(holiday_laid_off_2022)
```

Empty DataFrame

```
Columns: [company, location, industry, total_laid_off,
percentage_laid_off, stage, country, funds_raised, sector]
Index: []
```

```
print(holiday_laid_off_2023)
```

	company	location	industry	total_laid_off	\
date					
2022-12-27	BILIBILI	SHANGHAI	MEDIA	0.0	
2022-12-25	ELEMENT	LONDON	OTHER	0.0	
2022-12-27	OCTOPUS NETWORK	BEAU VALLON	CRYPTO	0.0	
2022-12-26	PAYU	AMSTERDAM	FINANCE	150.0	

	percentage_laid_off	stage	country
funds_raised \			
date			
2022-12-27	0.30	POST-IP0	CHINA
3700.0			
2022-12-25	0.15	SERIES B	UNITED KINGDOM
96.0			
2022-12-27	0.40	SERIES A	SEYCHELLES
8.0			
2022-12-26	0.06	ACQUIRED	NETHERLANDS
0.0			

	sector
date	
2022-12-27	Media
2022-12-25	Other
2022-12-27	Technology
2022-12-26	Finance

Kết quả ta thấy cuối năm 2022 có 4 công ty cho một số lượng khá lớn nhân viên nghỉ khỏe ngay trong thời gian diễn ra trong holiday week 🤔.

Dữ liệu layoff ở Việt Nam

```
print(df[df.country=='VIETNAM'])
```

	company	location	industry	total_laid_off	\
1340	PROPZY	HO CHI MINH CITY	REAL ESTATE	0.0	
1341	PROPZY	HO CHI MINH CITY	REAL ESTATE	0.0	
1891	WEFIT	HANOI	FITNESS	0.0	

	percentage_laid_off	date	stage	country	funds_raised
--	---------------------	------	-------	---------	--------------

1340	0.5	2022-06-07	SERIES A	VIETNAM	33.0
1341	1.0	2022-09-13	SERIES A	VIETNAM	33.0
1891	1.0	2020-05-13	SEED	VIETNAM	1.0

	sector
1340	Other
1341	Other
1891	Healthcare

Ta thấy có 3 đợt sa thải được ghi nhận tại Việt Nam đến từ 2 công ty.

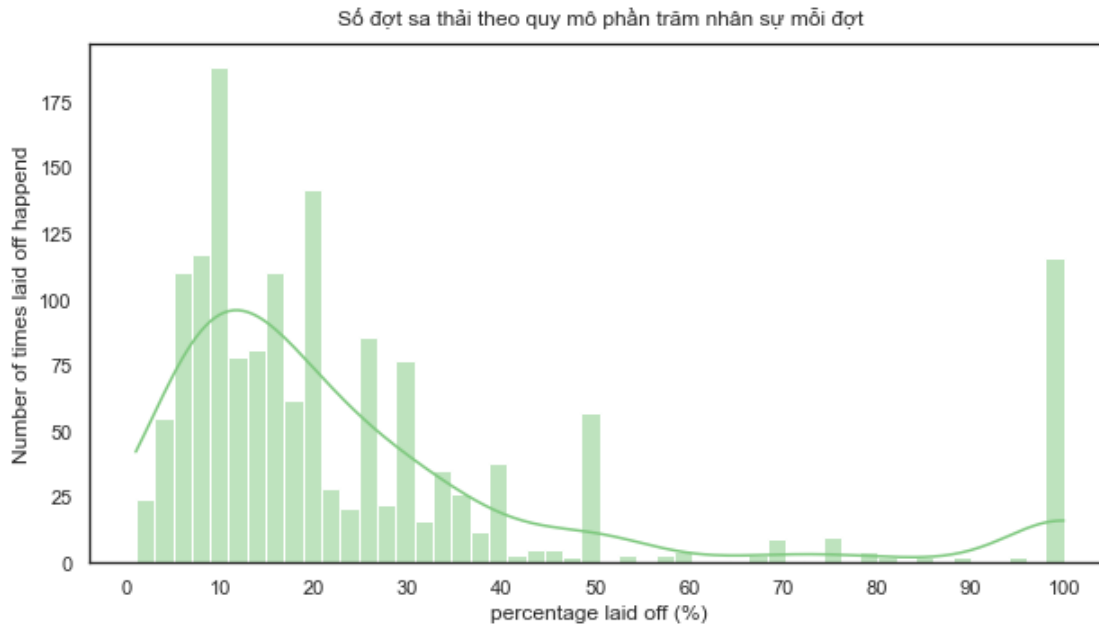
Với mỗi đợt sa thải, các công ty thường cho nghỉ bao nhiêu phần trăm nhân sự?

```
df_clean_percentage=df[df['percentage_laid_off']>0]
df_clean_percentage['percentage_laid_off']=df_clean_percentage['percentage_laid_off']*100
df_clean_percentage['percentage_lo_range']=pd.cut(x=df_clean_percentage['percentage_laid_off'], bins=[0, 10, 20, 30,
```

```
40,50,60,70,80,90,100],labels=['0-10', '10-20', '20-30',
                                '30-40','40-50','50-60','60-70',
                                '70-80','80-90','90-100'],include_lowest=True)
```

```
fig, ax = plt.subplots(figsize=(10, 6), tight_layout=True)
fig.tight_layout(pad=5.0)
ax.set_xticks(np.arange(0, 101, 10))
ax.set_title( 'Số đợt sa thải theo quy mô phân trăm nhân sự mỗi đợt',wrap=True,pad=10)
ax.set_xlabel('percentage laid off (%)')
ax.set_ylabel('Number of times laid off happend')
sns.histplot(df_clean_percentage['percentage_laid_off'], kde=True, bins=50)
```

```
<AxesSubplot: title={'center': 'Số đợt sa thải theo quy mô phân trăm nhân sự mỗi đợt'}, xlabel='percentage laid off (%)', ylabel='Number of times laid off happend'>
```

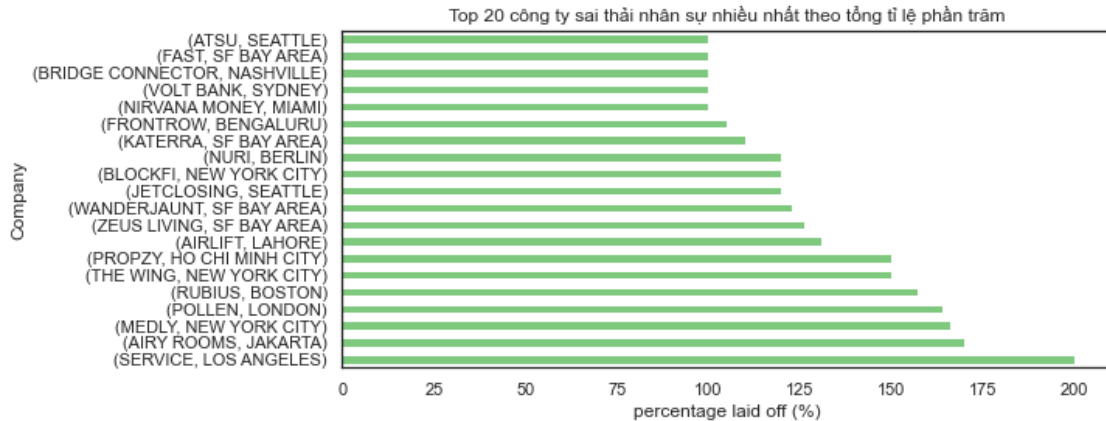


Ta thấy biểu đồ có 2 đỉnh. Đỉnh lớn nằm quanh 10% cho ta nhận định với mỗi đợt sa thải thì khoảng 10% nhân sự là con số ưu thích của các công ty. Biểu đồ lệch trái cho thấy các công ty có xu hướng luôn có gắng để sa thải ít hơn 50% nhân sự mỗi đợt. Điều thú vị khác là đỉnh thứ 2 nằm ở mức 100%. Điều này có thể liên quan đến các công ty quy nhỏ, toàn bộ các thành viên gắn kết với nhau đến những giây phút cuối cùng ??

Top 20 công ty có tổng % số lượng nhân sự bị sa thải nhiều nhất

```
fig, ax = plt.subplots( figsize=(9, 5), tight_layout=True)
fig.tight_layout(pad=5.0)
top_20 =
df.groupby(['company', 'location']).sum().sort_values(by='percentage_la
id_off', ascending=False).head(20)['percentage_laid_off'] * 100.0
top_20.plot(kind='barh', ax=ax)
ax.set_title('Top 20 công ty sai thải nhân sự nhiều nhất theo tổng tỉ
lệ phần trăm')
ax.set_xlabel('percentage laid off (%)')
ax.set_ylabel('Company')

Text(52.0, 0.5, 'Company')
```



Ta có thể thấy là có nhiều công ty sa thải với tổng hơn cả 100% nhân sự. Nguyên do là vì họ đã sa thải- tuyển lại và tiếp tục sa thải tiếp trong nhiều đợt. Một ví dụ là công ty Air rooms, ngày 2020-04-22 họ sa thải 70% nhân sự và sau đó vào ngày 2020-05-07 họ tiếp tục sa thải toàn bộ 100% nhân sự. Cá biệt có công ty Service trụ sở tại Los Angeles đã sa thải 2 lần với quy mô mỗi lần là 100% nhân sự chỉ cách nhau vài ngày 🤖.

```
print(df[df.company=='AIRY ROOMS'])
print(df[df.company=='SERVICE'])
```

	company	location	industry	total_laid_off	percentage_laid_off
44	AIRY ROOMS	JAKARTA	TRAVEL	0.0	1.0
45	AIRY ROOMS	JAKARTA	TRAVEL	0.0	0.7

	date	stage	country	funds_raised	sector
44	2020-05-07	UNKNOWN	INDONESIA	0.0	NaN
45	2020-04-22	UNKNOWN	INDONESIA	0.0	NaN

	company	location	industry	total_laid_off	percentage_laid_off
1487	SERVICE	LOS ANGELES	TRAVEL	0.0	1.0
1488	SERVICE	LOS ANGELES	TRAVEL	0.0	1.0

	date	stage	country	funds_raised	sector
1487	2020-03-16	SEED	UNITED STATES	5.1	NaN
1488	2020-03-20	SEED	UNITED STATES	5.0	NaN

Tìm hiểu về các công ty sa thải hơn 50% nhân sự trong các đợt

```
more_than_50percent_laid_off=df_clean_percentage[df_clean_percentage.p
ercentage_laid_off>=50]
# số lần sa thải có quy mô > 50% nhân viên
n_more_than_50percent_laid_off=more_than_50percent_laid_off.shape[0]
```



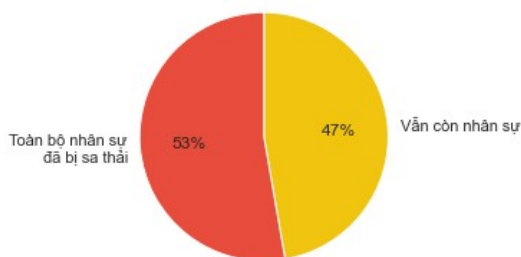
```
# số lần sa thải có quy mô 100% nhân viên
laid_off_all_employees
=df_clean_percentage[df_clean_percentage.percentage_laid_off==100]
n_laid_off_all_employees = laid_off_all_employees.shape[0]
print("Tổng số đợt sa thải >50% nhân sự:
"+str(n_more_than_50percent_laid_off))
print("Tổng số đợt sa thải 100% nhân sự:
"+str(n_laid_off_all_employees))

Tổng số đợt sa thải >50% nhân sự: 220
Tổng số đợt sa thải 100% nhân sự: 116

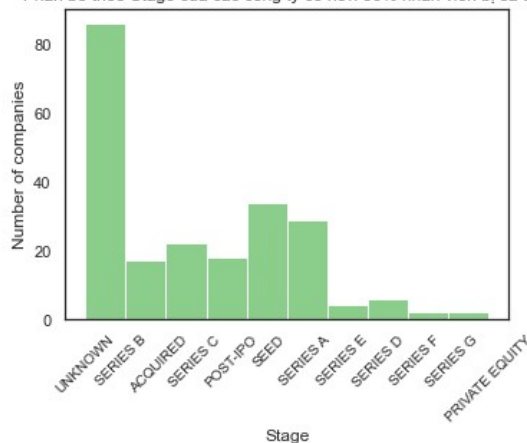
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5),
tight_layout=True)
fig.tight_layout(pad=5.0)
#pie chart
pie_lb=['\n'.join(wrap('Vẫn còn nhân sự', 15)),'\n'.join(wrap('Toàn bộ
nhân sự đã bị sa thải', 15))]
ax1.pie([n_more_than_50percent_laid_off-
n_laid_off_all_employees,n_laid_off_all_employees],labels=pie_lb,autop
ct='%.0f%',counterclock=False, startangle=-
270,colors=['#F1C40F','#E74C3C'])
ax1.set_title('Tình trạng nhân sự của các công ty có hơn 50% nhân viên
bị sa thải',wrap=True)

#histogram
ax2.hist(more_than_50percent_laid_off['stage'],alpha=0.9)
plt.xticks(rotation=45, size=10)
ax2.set_xlabel('Stage')
ax2.set_ylabel('Number of companies')
ax2.set_title('Phân bố theo Stage của các công ty có hơn 50% nhân viên
bị sa thải',wrap=True)
plt.show()
```

Tình trạng nhân sự của các công ty có hơn 50% nhân viên bị sa thải



Phân bố theo Stage của các công ty có hơn 50% nhân viên bị sa thải



Dựa vào biểu đồ tròn ta thấy rằng, một khi đã sa thải hơn 50% nhân sự thì có tới hơn 53% đó sẽ là những lần sa thải toàn bộ 100% nhân sự. Nhìn vào biểu đồ histogram ta thấy rằng

đa số những công ty thuộc nhóm ra thải nhiều này chủ yếu không có thông tin về stage hoặc thuộc những giai đoạn khởi đầu (Seed, SeriesA...). Những công ty này thiếu sự Ổn định.

Tìm hiểu quan hệ giữa total_laid_off,percentage_laid_off,funds_raised,total_employees

xử lý các dòng có giá trị 0

```
df_cor=df_with_total_employees[(df_with_total_employees != 0).all(1)]
df_cor.reset_index(drop=True)
```

lấy 4 cột

'total_laid_off','percentage_laid_off','funds_raised','total_employees

' để' so sánh correlation

```
df_cor=df_cor.loc[:,
```

```
['total_laid_off','percentage_laid_off','funds_raised','total_employees']]
```

```
df_cor.shape[0]
```

1088

vẽ correlation map

```
corr = df_cor.corr()
```

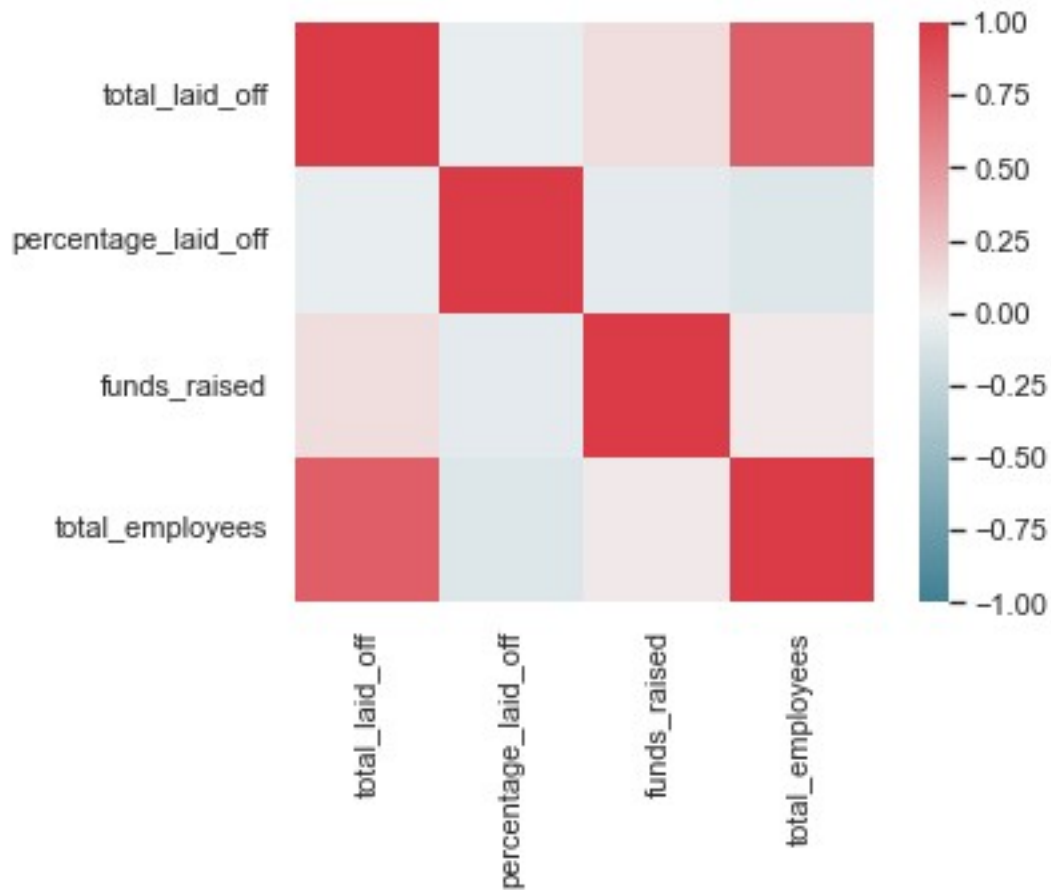
```
sns.heatmap(corr,
```

```
    cmap=sns.diverging_palette(220, 10, as_cmap=True),
```

```
    vmin=-1.0, vmax=1.0,
```

```
    square=True)
```

<AxesSubplot: >



Ta thấy rằng total_employees và total_laid_off có quan hệ phụ thuộc mạnh còn các cặp feature còn lại gần như không có quan hệ.