In [1]:
```
# importing the pandas library
import pandas as pd
import numpy as np
```

In [2]:
```
# loading the csv data
lay_offs = pd.read_csv('layoffs.csv')
```

In [3]:
```
# summary of the dataset displaced from top to bottom
lay_offs
```

Out[3]:

| | company | location | industry | total_laid_off | percentage | date | funds_raised |
|---|---|---|---|---|---|---|---|
| **0** | Veev | SF Bay Area | Real Estate | 100.0 | 0.30 | 11/11/2022 | 597.0 |
| **1** | GoTo Group | Jakarta | Transportation | 1000.0 | 0.10 | 11/10/2022 | 1300.0 |
| **2** | Juul | SF Bay Area | Consumer | 400.0 | 0.30 | 11/10/2022 | 1500.0 |
| **3** | InfluxData | SF Bay Area | NaN | 65.0 | 0.27 | 11/10/2022 | 119.0 |
| **4** | Coinbase | SF Bay Area | Crypto | 60.0 | NaN | 11/10/2022 | 549.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **1569** | Service | Los Angeles | Travel | NaN | 1.00 | 3/16/2020 | 5.1 |
| **1570** | HopSkipDrive | Los Angeles | Transportation | 8.0 | 0.10 | 3/13/2020 | 45.0 |
| **1571** | Panda Squad | SF Bay Area | Consumer | 6.0 | 0.75 | 3/13/2020 | 1.0 |
| **1572** | Tamara Mellon | Los Angeles | Retail | 20.0 | 0.40 | 3/12/2020 | 90.0 |
| **1573** | EasyPost | Salt Lake City | Logistics | 75.0 | NaN | 3/11/2020 | 12.0 |

1574 rows × 9 columns

# EXPLORATORY ANALYSIS

In [4]:
```
# view the top five rows of the data
lay_offs.head()
```

Out[4]:

| | company | location | industry | total_laid_off | percentage | date | funds_raised | stag |
|---|---|---|---|---|---|---|---|---|
| **0** | Veev | SF Bay Area | Real Estate | 100.0 | 0.30 | 11/11/2022 | 597.0 | Series |
| **1** | GoTo Group | Jakarta | Transportation | 1000.0 | 0.10 | 11/10/2022 | 1300.0 | Unknow |
| **2** | Juul | SF Bay Area | Consumer | 400.0 | 0.30 | 11/10/2022 | 1500.0 | Unknow |
| **3** | InfluxData | SF Bay Area | NaN | 65.0 | 0.27 | 11/10/2022 | 119.0 | Series |
| **4** | Coinbase | SF Bay Area | Crypto | 60.0 | NaN | 11/10/2022 | 549.0 | IP |

In [5]:
```
# To get a statistical summary of our data
lay_offs.describe()
```

Out[5]:

| | total_laid_off | percentage | funds_raised |
|---|---|---|---|
| **count** | 1132.000000 | 1053.000000 | 1472.000000 |
| **mean** | 191.422261 | 0.277756 | 903.870445 |
| **std** | 511.777912 | 0.266320 | 6663.154374 |
| **min** | 3.000000 | 0.000000 | 0.000000 |
| **25%** | 30.000000 | 0.100000 | 42.000000 |
| **50%** | 70.000000 | 0.200000 | 123.000000 |
| **75%** | 150.000000 | 0.330000 | 359.250000 |
| **max** | 11000.000000 | 1.000000 | 121900.000000 |

In [6]:
```
# To check for missing values in our data
lay_offs .isnull().sum()
```

Out[6]:
```
company             0
location            0
industry            3
total_laid_off    442
percentage        521
date                0
funds_raised      102
stage               4
country             0
dtype: int64
```

In [7]:
```
# To check for rows that has duplicated data.
lay_offs.duplicated().sum()
```

Out[7]: 1

In [8]:
```
# To view the five last rows of our data
lay_offs.tail()
```

Out[8]:

| | company | location | industry | total_laid_off | percentage | date | funds_raised | |
|---|---|---|---|---|---|---|---|---|
| **1569** | Service | Los Angeles | Travel | NaN | 1.00 | 3/16/2020 | 5.1 | |
| **1570** | HopSkipDrive | Los Angeles | Transportation | 8.0 | 0.10 | 3/13/2020 | 45.0 | U |
| **1571** | Panda Squad | SF Bay Area | Consumer | 6.0 | 0.75 | 3/13/2020 | 1.0 | |
| **1572** | Tamara Mellon | Los Angeles | Retail | 20.0 | 0.40 | 3/12/2020 | 90.0 | S |
| **1573** | EasyPost | Salt Lake City | Logistics | 75.0 | NaN | 3/11/2020 | 12.0 | S |

In [9]:
```
# to check the rows and columns in our dataset
lay_offs.shape
```

Out[9]:
```
(1574, 9)
```

In [10]:
```
# To get a summary of our dataset
lay_offs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1574 entries, 0 to 1573
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   company         1574 non-null   object 
 1   location        1574 non-null   object 
 2   industry        1571 non-null   object 
 3   total_laid_off  1132 non-null   float64
 4   percentage      1053 non-null   float64
 5   date            1574 non-null   object 
 6   funds_raised    1472 non-null   float64
 7   stage           1570 non-null   object 
 8   country         1574 non-null   object 
dtypes: float64(3), object(6)
memory usage: 110.8+ KB
```

In [11]:
```
# To know the total columns in our dataset
len(lay_offs)
```

Out[11]:
```
1574
```

In [12]:
```
# To retrieve all column names in our data
lay_offs.columns
```

Out[12]:
```
Index(['company', 'location', 'industry', 'total_laid_off', 'percentage',
       'date', 'funds_raised', 'stage', 'country'],
      dtype='object')
```

In [13]:
```
# To check the number of unique vales in our data
lay_offs.nunique()
```

Out[13]:
```
company           1327
location           154
industry            27
total_laid_off     221
percentage          69
date               384
funds_raised       523
stage               15
country             54
dtype: int64
```

In [14]:
```
# To retrieve random 50 rows
lay_offs.sample(50)
```

Out[14]:

| | company | location | industry | total_laid_off | percentage | date | funds_raised |
|---|---|---|---|---|---|---|---|
| **768** | BookClub | Salt Lake City | Education | 12.0 | 0.25 | 5/31/2022 | 26.0 |
| **1037** | OYO | Dallas | Travel | NaN | NaN | 6/24/2020 | 3200.0 |
| **1365** | RainFocus | Salt Lake City | Marketing | NaN | NaN | 4/7/2020 | 41.0 |
| **381** | Nomad | Sao Paulo | Finance | NaN | 0.20 | 8/4/2022 | 290.0 |
| **935** | Instacart | SF Bay Area | Food | 1877.0 | NaN | 1/21/2021 | 2400.0 |
| **195** | Ola | Bengaluru | Transportation | 200.0 | NaN | 9/19/2022 | 5000.0 |
| **370** | Talkdesk | SF Bay Area | Support | NaN | NaN | 8/5/2022 | 497.0 |
| **32** | Code42 | Minneapolis | Security | NaN | 0.15 | 11/7/2022 | 137.0 |
| **508** | Wave | Dakar | Finance | 300.0 | 0.15 | 7/13/2022 | 292.0 |
| **127** | Qin1 | Noida | Education | NaN | 1.00 | 10/14/2022 | NaN |
| **641** | Balto | St. Louis | Sales | 30.0 | NaN | 6/22/2022 | 51.0 |
| **50** | LiveRamp | SF Bay Area | Marketing | NaN | 0.10 | 11/3/2022 | 16.0 |
| **310** | Tempo Automation | SF Bay Area | Other | 54.0 | NaN | 8/17/2022 | 74.0 |
| **1393** | Opencare | Toronto | Healthcare | 18.0 | 0.25 | 4/3/2020 | 24.0 |
| **646** | TaskUs | Los Angeles | Support | 52.0 | 0.00 | 6/21/2022 | 279.0 |
| **1217** | Jiobit | Chicago | Consumer | 6.0 | 0.21 | 4/24/2020 | 12.0 |
| **138** | Sketch | The Hague | Other | 80.0 | NaN | 10/11/2022 | 20.0 |
| **1147** | Uber | SF Bay Area | Transportation | 3700.0 | 0.14 | 5/6/2020 | 24700.0 |
| **565** | Zepto | Brisbane | Finance | NaN | 0.10 | 7/2/2022 | 25.0 |
| **1344** | Spyce | Boston | Food | 4.0 | 0.12 | 4/8/2020 | 26.0 |
| **1162** | League | Toronto | Healthcare | NaN | NaN | 5/5/2020 | 76.0 |
| **539** | Emotive | Los Angeles | Marketing | 30.0 | 0.18 | 7/7/2022 | 78.0 |
| **1521** | Lyric | SF Bay Area | Real Estate | 100.0 | NaN | 3/25/2020 | 179.1 |
| **1454** | AdRoll | Salt Lake City | Marketing | 210.0 | 0.30 | 3/31/2020 | 89.0 |
| **505** | Alto Pharmacy | SF Bay Area | Healthcare | NaN | NaN | 7/14/2022 | 560.0 |
| **364** | DataRobot | Boston | Data | NaN | NaN | 8/8/2022 | 1000.0 |

| | company | location | industry | total_laid_off | percentage | date | funds_raised |
|---|---|---|---|---|---|---|---|
| **523** | Airlift | Lahore | Logistics | NaN | 1.00 | 7/12/2022 | 109.0 |
| **665** | JOKR | New York City | Food | 50.0 | 0.05 | 6/16/2022 | 430.0 |
| **204** | DayTwo | SF Bay Area | Healthcare | NaN | NaN | 9/15/2022 | 90.0 |
| **1087** | Teamwork | Cork | Other | 21.0 | NaN | 5/22/2020 | NaN |
| **1466** | Kazoo | Austin | HR | NaN | 0.35 | 3/31/2020 | 8.0 |
| **1514** | Ecobee | Toronto | Energy | 47.0 | 0.10 | 3/26/2020 | 149.0 |
| **1550** | Cabin | SF Bay Area | Travel | NaN | 0.20 | 3/23/2020 | 3.0 |
| **871** | Thinkific | Vancouver | Education | 100.0 | 0.20 | 3/29/2022 | 22.0 |
| **406** | Hash | Sao Paulo | Finance | 58.0 | 0.50 | 8/1/2022 | 58.0 |
| **231** | GoStudent | Vienna | Education | 200.0 | NaN | 9/8/2022 | 686.0 |
| **1099** | Pollen | London | Travel | 69.0 | 0.31 | 5/19/2020 | 88.0 |
| **541** | Twitter | SF Bay Area | Consumer | NaN | NaN | 7/7/2022 | 5700.0 |
| **556** | Lightricks | Jerusalem | Consumer | 80.0 | 0.12 | 7/4/2022 | 335.0 |
| **438** | Skai | Tel Aviv | Marketing | 30.0 | 0.04 | 7/27/2022 | 60.0 |
| **975** | Akerna | Denver | Logistics | NaN | NaN | 9/2/2020 | NaN |
| **1314** | Geekwire | Seattle | Media | 5.0 | 0.31 | 4/10/2020 | NaN |
| **603** | Vezeeta | Dubai | Healthcare | 50.0 | 0.10 | 6/28/2022 | 71.0 |
| **860** | Meesho | Bengaluru | Retail | 150.0 | NaN | 4/11/2022 | 1100.0 |
| **589** | Oye Rickshaw | New Delhi | Transportation | 40.0 | 0.20 | 6/29/2022 | 13.0 |
| **439** | Shopify | Ottawa | Retail | 1000.0 | 0.10 | 7/26/2022 | 122.0 |
| **424** | Ribbon | New York City | Real Estate | 136.0 | NaN | 7/28/2022 | 405.0 |
| **21** | Faze Medicines | Boston | Healthcare | NaN | 1.00 | 11/9/2022 | 81.0 |
| **217** | Mode Analytics | SF Bay Area | Data | 25.0 | NaN | 9/12/2022 | 81.0 |
| **1421** | Sauce Labs | SF Bay Area | Infrastructure | 30.0 | 0.15 | 4/2/2020 | 151.0 |

In [15]:
```
# To know the proportion funds raised to non-funds raised activity
funds_prop = lay_offs.funds_raised.value_counts(normalize = True)
pd.DataFrame(funds_prop)
```

Out[15]:

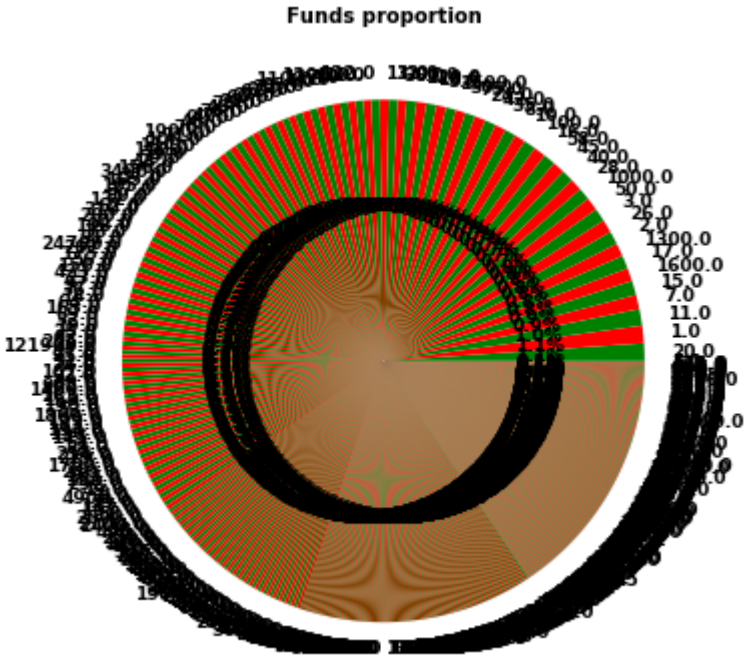
| | funds_raised |
|---|---|
| **20.0** | 0.010870 |
| **1.0** | 0.010870 |
| **11.0** | 0.010190 |
| **7.0** | 0.008832 |
| **15.0** | 0.008832 |
| **...** | ... |
| **500.0** | 0.000679 |
| **461.0** | 0.000679 |
| **171.0** | 0.000679 |
| **429.0** | 0.000679 |
| **5.1** | 0.000679 |

523 rows × 1 columns

## let's create a pie chart to better visualize the proportion of funds raised to non-funds raised activities

In [16]:
```
# To import Data Visualization libraries
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

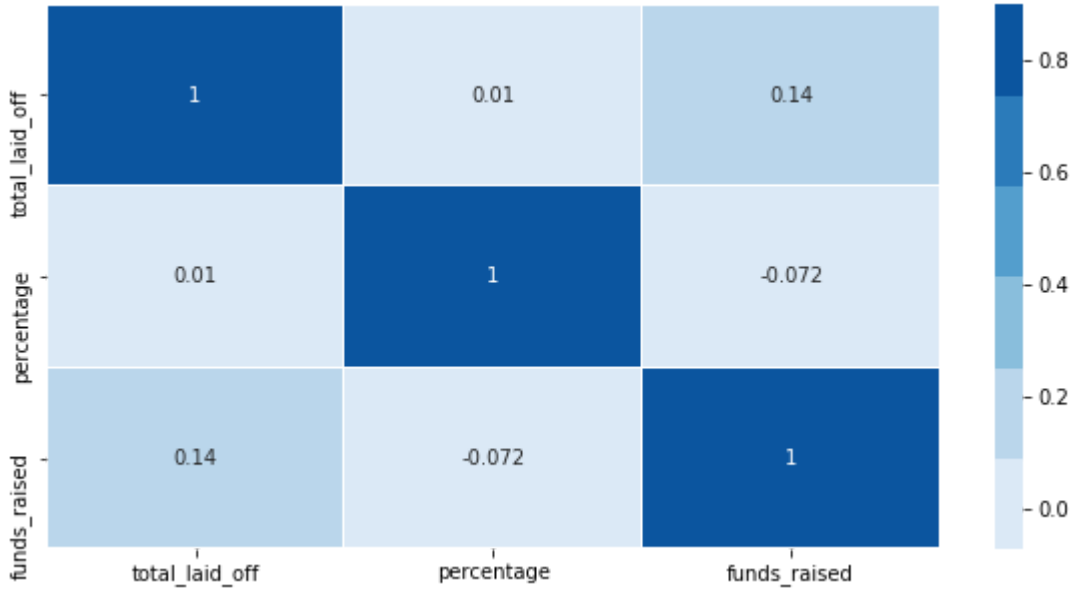
In [17]:
```
plt.figure(figsize=[10,6], facecolor='white')
plt.pie(funds_prop, labels = funds_prop.index, colors=['green', 'red'], textpr
ops={'fontsize':'9', 'color':'black', 'fontweight':'bold'},autopct='%1.1f%%')
plt.title("Funds proportion", fontsize=10, color="black", fontweight="bold" )
plt.show()
```




In [18]:
```
# Let's visualize the relationship between our independent and dependent varia
bles by plotting a heatmap
# importing the seaborn library for visaualization
import seaborn as sns
```

In [19]:
```
# To visualize the correlation matrix in a Heatmap
plt.figure(figsize=(10, 5))
colormap = sns.color_palette("Blues")
sns.heatmap(lay_offs.corr(), cmap=colormap, vmax=0.9, annot=True, linecolor="white", linewidths=0.02)
```

Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x26a5188c348>




In [20]:
```
#Creating a separate dataframe that excludes Company, location, industry,date, stage and country. These columns will not be used for the clustering but they are still neccessary
lay_off = lay_offs.drop(['company', 'location', 'date', 'stage', 'country'], axis = 1)
```

In [21]:
```
lay_off
```

Out[21]:

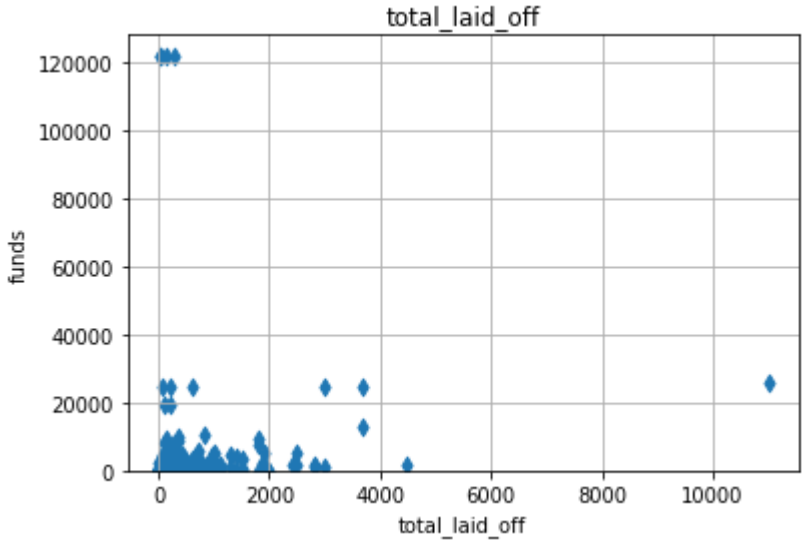
| | industry | total_laid_off | percentage | funds_raised |
|---|---|---|---|---|
| **0** | Real Estate | 100.0 | 0.30 | 597.0 |
| **1** | Transportation | 1000.0 | 0.10 | 1300.0 |
| **2** | Consumer | 400.0 | 0.30 | 1500.0 |
| **3** | NaN | 65.0 | 0.27 | 119.0 |
| **4** | Crypto | 60.0 | NaN | 549.0 |
| **...** | ... | ... | ... | ... |
| **1569** | Travel | NaN | 1.00 | 5.1 |
| **1570** | Transportation | 8.0 | 0.10 | 45.0 |
| **1571** | Consumer | 6.0 | 0.75 | 1.0 |
| **1572** | Retail | 20.0 | 0.40 | 90.0 |
| **1573** | Logistics | 75.0 | NaN | 12.0 |

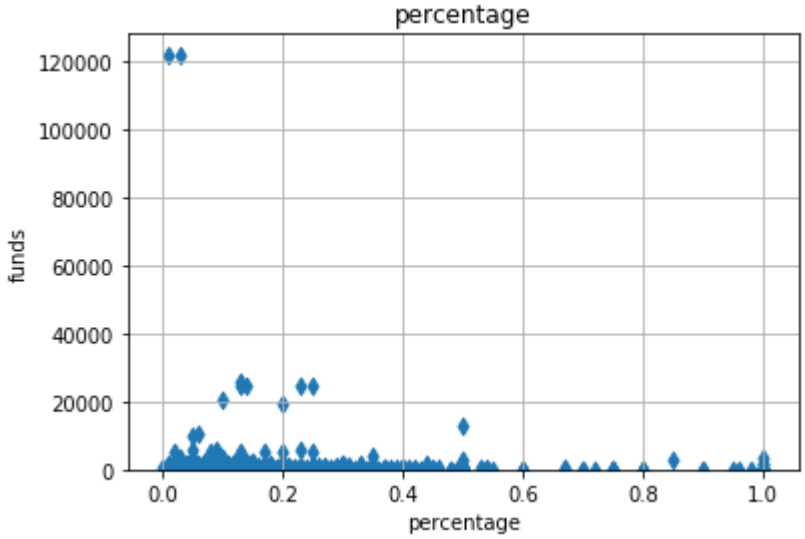
1574 rows × 4 columns

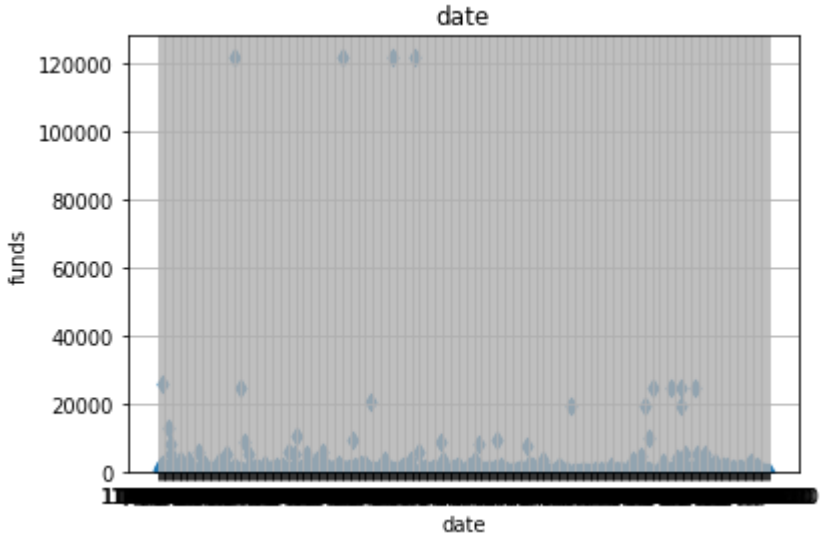
## As part of understanding your data you may choose to plot some charts

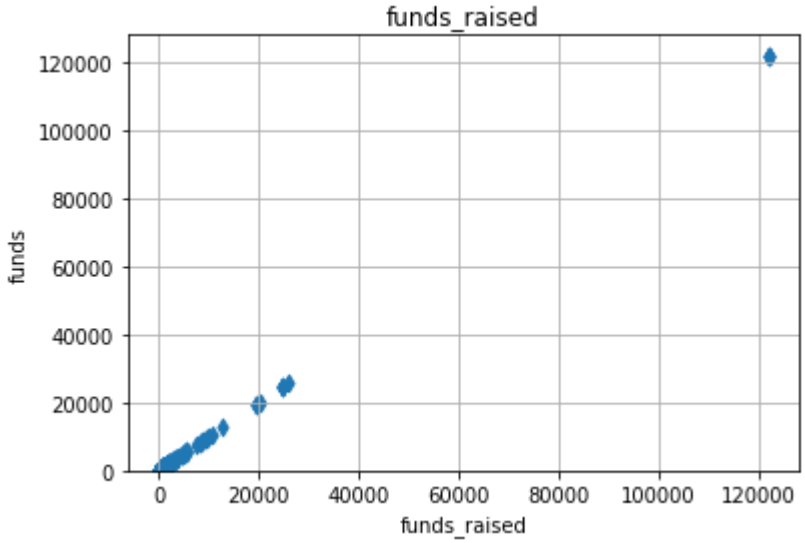
In [22]:
```
#importing Data Visualization modules
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

In [23]:
```
import warnings
warnings.filterwarnings('ignore')
for i in list(lay_offs.columns)[3:7]:
    x_axis = i
    x = lay_offs[x_axis]
    y = lay_offs['funds_raised']
    plt.figure(figsize=(6, 4))
    plt.scatter(x,y, marker = 'd')
    plt.grid()
    plt.ylim(ymin=0)
    plt.xlabel(x_axis)
    plt.ylabel('funds')
    plt.title(x_axis)
    plt.subplots_adjust(hspace = 0.5)
    plt.show();
```


total_laid_off
funds
total_laid_off
0
20000
40000
60000
80000
100000
120000
0
2000
4000
6000
8000
10000


percentage
funds
percentage
0
20000
40000
60000
80000
100000
120000
0.0
0.2
0.4
0.6
0.8
1.0


date
funds
date
0
20000
40000
60000
80000
100000
120000




## OUTLIER DETECTION AND REMOVAL

We are going to check for outliers using total laid off columns, we take min of the total laid off = 3 and the max is 11000.00

In [24]:
```
# Using the describe function to check for outliers
lay_off.total_laid_off.describe()
```

Out[24]:
```
count     1132.000000
mean       191.422261
std        511.777912
min          3.000000
25%         30.000000
50%         70.000000
75%        150.000000
max      11000.000000
Name: total_laid_off, dtype: float64
```

In [25]:
```
total_laid_off_3 = lay_off.loc[lay_off['total_laid_off']== 3]
total_laid_off_3
```

Out[25]:

| | industry | total_laid_off | percentage | funds_raised |
|---|---|---|---|---|
| **1054** | Retail | 3.0 | 0.27 | 2.0 |

In [26]:
```
total_laid_off_3.industry.unique()
```

Out[26]:
```
array(['Retail'], dtype=object)
```

In [27]:
```
total_laid_off_3 = lay_offs.loc[lay_off['total_laid_off']== 3]
total_laid_off_3
```

Out[27]:

| | company | location | industry | total_laid_off | percentage | date | funds_raised | stage | co |
|---|---|---|---|---|---|---|---|---|---|
| **1054** | Branch | New York City | Retail | 3.0 | 0.27 | 6/11/2020 | 2.0 | Seed | U S |

In [28]:
```
lay_off_no_outliers = lay_off[~((lay_off.total_laid_off == 3))]
lay_off_no_outliers
```

Out[28]:

| | industry | total_laid_off | percentage | funds_raised |
|---|---|---|---|---|
| **0** | Real Estate | 100.0 | 0.30 | 597.0 |
| **1** | Transportation | 1000.0 | 0.10 | 1300.0 |
| **2** | Consumer | 400.0 | 0.30 | 1500.0 |
| **3** | NaN | 65.0 | 0.27 | 119.0 |
| **4** | Crypto | 60.0 | NaN | 549.0 |
| **...** | ... | ... | ... | ... |
| **1569** | Travel | NaN | 1.00 | 5.1 |
| **1570** | Transportation | 8.0 | 0.10 | 45.0 |
| **1571** | Consumer | 6.0 | 0.75 | 1.0 |
| **1572** | Retail | 20.0 | 0.40 | 90.0 |
| **1573** | Logistics | 75.0 | NaN | 12.0 |

1573 rows × 4 columns

In [29]:
```
lay_off_no_outliers.nunique()
```

Out[29]:
```
industry           27
total_laid_off    220
percentage         69
funds_raised      523
dtype: int64
```

## Lets adopt the target encoding method for our model

In [30]:
```
!pip install category_encoders
```

```
Requirement already satisfied: category_encoders in c:\users\lekue\anaconda3
\lib\site-packages (2.5.1.post0)
Requirement already satisfied: statsmodels>=0.9.0 in c:\users\lekue\anaconda3
\lib\site-packages (from category_encoders) (0.11.0)
Requirement already satisfied: scikit-learn>=0.20.0 in c:\users\lekue\anacond
a3\lib\site-packages (from category_encoders) (1.0.2)
Requirement already satisfied: pandas>=1.0.5 in c:\users\lekue\anaconda3\lib
\site-packages (from category_encoders) (1.3.5)
Requirement already satisfied: scipy>=1.0.0 in c:\users\lekue\anaconda3\lib\s
ite-packages (from category_encoders) (1.4.1)
Requirement already satisfied: patsy>=0.5.1 in c:\users\lekue\anaconda3\lib\s
ite-packages (from category_encoders) (0.5.1)
Requirement already satisfied: numpy>=1.14.0 in c:\users\lekue\anaconda3\lib
\site-packages (from category_encoders) (1.18.1)
Requirement already satisfied: joblib>=0.11 in c:\users\lekue\anaconda3\lib\s
ite-packages (from scikit-learn>=0.20.0->category_encoders) (0.14.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\lekue\anacond
a3\lib\site-packages (from scikit-learn>=0.20.0->category_encoders) (3.1.0)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\lekue\anaco
nda3\lib\site-packages (from pandas>=1.0.5->category_encoders) (2.8.1)
Requirement already satisfied: pytz>=2017.3 in c:\users\lekue\anaconda3\lib\s
ite-packages (from pandas>=1.0.5->category_encoders) (2019.3)
Requirement already satisfied: six in c:\users\lekue\anaconda3\lib\site-packa
ges (from patsy>=0.5.1->category_encoders) (1.14.0)
```

In [31]:
```
from category_encoders import TargetEncoder
columns = ['industry', 'percentage']
target = 'funds_raised'
for column in columns:
    targetE = TargetEncoder()
    targetE.fit(X = lay_off_no_outliers[columns], y = lay_off_no_outliers[targ
et])
```

In [32]:
```
transformed_values = targetE.transform(X = lay_off_no_outliers[columns], y = l
ay_off_no_outliers[target])
transformed_values
```

Out[32]:

| | industry | percentage |
|---|---|---|
| **0** | 1067.392045 | 0.30 |
| **1** | 2488.690495 | 0.10 |
| **2** | 1375.463514 | 0.30 |
| **3** | 361.780239 | 0.27 |
| **4** | 255.279070 | NaN |
| **...** | ... | ... |
| **1569** | 481.794118 | 1.00 |
| **1570** | 2488.690495 | 0.10 |
| **1571** | 1375.463514 | 0.75 |
| **1572** | 350.065323 | 0.40 |
| **1573** | 248.437500 | NaN |

1573 rows × 2 columns

In [33]:
```
#Deleting the industry and the percentage columns
lay_off_no_outliers.drop(['industry', 'percentage'], axis = 1, inplace = True)
lay_off_no_outliers
```

Out[33]:

| | total_laid_off | funds_raised |
|---|---|---|
| **0** | 100.0 | 597.0 |
| **1** | 1000.0 | 1300.0 |
| **2** | 400.0 | 1500.0 |
| **3** | 65.0 | 119.0 |
| **4** | 60.0 | 549.0 |
| **...** | ... | ... |
| **1569** | NaN | 5.1 |
| **1570** | 8.0 | 45.0 |
| **1571** | 6.0 | 1.0 |
| **1572** | 20.0 | 90.0 |
| **1573** | 75.0 | 12.0 |

1573 rows × 2 columns

In [34]:
```
# Merging the two tables
transaction = pd.concat([transformed_values, lay_off_no_outliers], axis = 1)
transaction
```

Out[34]:

| | industry | percentage | total_laid_off | funds_raised |
|---|---|---|---|---|
| **0** | 1067.392045 | 0.30 | 100.0 | 597.0 |
| **1** | 2488.690495 | 0.10 | 1000.0 | 1300.0 |
| **2** | 1375.463514 | 0.30 | 400.0 | 1500.0 |
| **3** | 361.780239 | 0.27 | 65.0 | 119.0 |
| **4** | 255.279070 | NaN | 60.0 | 549.0 |
| **...** | ... | ... | ... | ... |
| **1569** | 481.794118 | 1.00 | NaN | 5.1 |
| **1570** | 2488.690495 | 0.10 | 8.0 | 45.0 |
| **1571** | 1375.463514 | 0.75 | 6.0 | 1.0 |
| **1572** | 350.065323 | 0.40 | 20.0 | 90.0 |
| **1573** | 248.437500 | NaN | 75.0 | 12.0 |

1573 rows × 4 columns

In [35]:
```
# separating the independent from the dependent variables
X = transaction.drop(['funds_raised'], axis = 1)
X
```

Out[35]:

| | industry | percentage | total_laid_off |
|---|---|---|---|
| **0** | 1067.392045 | 0.30 | 100.0 |
| **1** | 2488.690495 | 0.10 | 1000.0 |
| **2** | 1375.463514 | 0.30 | 400.0 |
| **3** | 361.780239 | 0.27 | 65.0 |
| **4** | 255.279070 | NaN | 60.0 |
| **...** | ... | ... | ... |
| **1569** | 481.794118 | 1.00 | NaN |
| **1570** | 2488.690495 | 0.10 | 8.0 |
| **1571** | 1375.463514 | 0.75 | 6.0 |
| **1572** | 350.065323 | 0.40 | 20.0 |
| **1573** | 248.437500 | NaN | 75.0 |

1573 rows × 3 columns

In [36]:
```
y = transaction['funds_raised']
y
```

Out[36]:
```
0        597.0
1       1300.0
2       1500.0
3        119.0
4        549.0
         ...
1569       5.1
1570      45.0
1571       1.0
1572      90.0
1573      12.0
Name: funds_raised, Length: 1573, dtype: float64
```

In [37]:
```
# Let's scale our data before fitting, This process ensures that we do not get
our scaled data in arrays but in a dataframe.
from sklearn.preprocessing import StandardScaler
# Create a scaler object
sc = StandardScaler()
X[['industry','percentage','total_laid_off']] = sc.fit_transform(X[['industry'
,'percentage','total_laid_off']])
X
```

Out[37]:

| | industry | percentage | total_laid_off |
|---|---|---|---|
| **0** | 0.070321 | 0.083495 | -0.178973 |
| **1** | 0.853393 | -0.667483 | 1.579708 |
| **2** | 0.240054 | 0.083495 | 0.407254 |
| **3** | -0.318440 | -0.029152 | -0.247366 |
| **4** | -0.377118 | NaN | -0.257136 |
| **...** | ... | ... | ... |
| **1569** | -0.252318 | 2.711917 | NaN |
| **1570** | 0.853393 | -0.667483 | -0.358749 |
| **1571** | 0.240054 | 1.773194 | -0.362657 |
| **1572** | -0.324894 | 0.458983 | -0.335300 |
| **1573** | -0.380887 | NaN | -0.227825 |

1573 rows × 3 columns

## DUE TO SOME INPUT CONTAINS, INFINITE OR TOO LARGE VALUES

WE INTRODUCE DATETIME INDEX

In [39]:
```
# CONVERTING THE DATE COLUMNS TO DATETIME DATA TYPE
from datetime import datetime
lay_offs.date = pd.to_datetime(lay_offs.date)
lay_offs.percentage = pd.to_datetime(lay_offs.percentage)
```

In [40]:
```
lay_offs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1574 entries, 0 to 1573
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype         
---  ------          --------------  -----         
 0   company         1574 non-null   object        
 1   location        1574 non-null   object        
 2   industry        1571 non-null   object        
 3   total_laid_off  1132 non-null   float64       
 4   percentage      1053 non-null   datetime64[ns]
 5   date            1574 non-null   datetime64[ns]
 6   funds_raised    1472 non-null   float64       
 7   stage           1570 non-null   object        
 8   country         1574 non-null   object        
dtypes: datetime64[ns](2), float64(2), object(5)
memory usage: 110.8+ KB
```

## The columns are now converted to datetime data format, Now we can perform some datetime functions

In [41]:
```
# let's extract the transaction hour, month and date it will be relevant to ou
r project
trans_hour = pd.DatetimeIndex(lay_offs.date).hour
trans_day = pd.DatetimeIndex(lay_offs.date).dayofweek + 1
trans_month = pd.DatetimeIndex(lay_offs.date).month
```

In [42]:
```
lay_offs.insert(1,'trans_hour',trans_hour)
lay_offs.insert(2,'trans_day',trans_day)
lay_offs.insert(3,'trans_month',trans_month)
```

In [43]:
```
lay_offs.head()
```

Out[43]:

| | company | trans_hour | trans_day | trans_month | location | industry | total_laid_off | percenta |
|---|---|---|---|---|---|---|---|---|
| **0** | Veev | 0 | 5 | 11 | SF Bay Area | Real Estate | 100.0 | 1970-01 |
| **1** | GoTo Group | 0 | 4 | 11 | Jakarta | Transportation | 1000.0 | 1970-01 |
| **2** | Juul | 0 | 4 | 11 | SF Bay Area | Consumer | 400.0 | 1970-01 |
| **3** | InfluxData | 0 | 4 | 11 | SF Bay Area | NaN | 65.0 | 1970-01 |
| **4** | Coinbase | 0 | 4 | 11 | SF Bay Area | Crypto | 60.0 | N |

In [44]:
```
### The three new independent variables have been added to our data, let's go ahead to create another predictor(total laid off) from the existing data of columns
```

In [45]:
```
# to find the funds raised using the column
amount_generated = 121900 - pd.DatetimeIndex(lay_offs.funds_raised).day
```

In [46]:
```
#inserting the percentage
lay_offs.insert(10,'amount_generated',amount_generated)
```

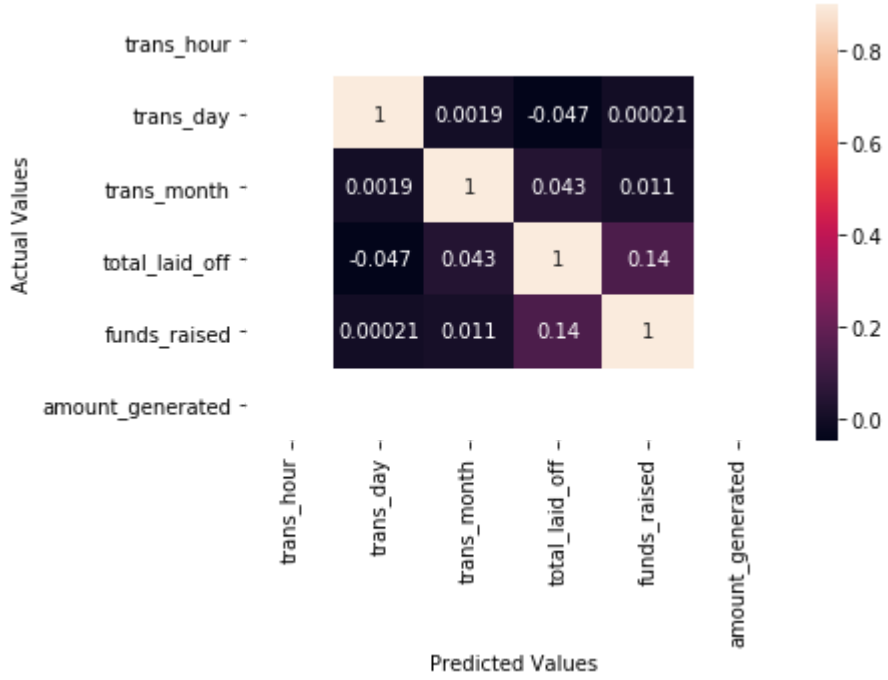
In [47]:
```
lay_offs.head()
```

Out[47]:

| | company | trans_hour | trans_day | trans_month | location | industry | total_laid_off | percenta |
|---|---|---|---|---|---|---|---|---|
| **0** | Veev | 0 | 5 | 11 | SF Bay Area | Real Estate | 100.0 | 1970-01 |
| **1** | GoTo Group | 0 | 4 | 11 | Jakarta | Transportation | 1000.0 | 1970-01 |
| **2** | Juul | 0 | 4 | 11 | SF Bay Area | Consumer | 400.0 | 1970-01 |
| **3** | InfluxData | 0 | 4 | 11 | SF Bay Area | NaN | 65.0 | 1970-01 |
| **4** | Coinbase | 0 | 4 | 11 | SF Bay Area | Crypto | 60.0 | N |

In [57]:
```
import seaborn as sns
#plotting the confusion metrics
plt.figure(figsize =(6, 4))
sns.heatmap(lay_offs.corr(), vmax=0.9, annot = True)
plt.xlabel('Predicted Values')
plt.ylabel('Actual Values')
```

Out[57]: Text(32.09374999999999, 0.5, 'Actual Values')




In [ ]:

In [ ]: