Below are the various visualizations used to undestand the data given of the fictional company called Crisco

Import the necessary packages for the visualization of Crisco's visitor's volume at 40 venues

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import numpy as np
import holoviews as hv

!pip install hvplot
import hvplot.pandas
```

```
     Collecting hvplot
       Downloading hvplot-0.7.3-py2.py3-none-any.whl (3.1 MB)
          |████████████████████████████████| 3.1 MB 7.2 MB/s
     Requirement already satisfied: bokeh>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from hvplot) (2.3.3)
     Requirement already satisfied: holoviews>=1.11.0 in /usr/local/lib/python3.7/dist-packages (from hvplot) (1.14.
     Requirement already satisfied: colorcet>=2 in /usr/local/lib/python3.7/dist-packages (from hvplot) (3.0.0)
     Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from hvplot) (1.3.5)
     Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-packages (from hvplot) (1.21.5)
     Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.7/dist-packages (from bokeh>=1.0.0->hvpl
     Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.7/dist-packages (from bokeh>=1.0.0->hvplot
     Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from bokeh>=1.0.
     Requirement already satisfied: tornado>=5.1 in /usr/local/lib/python3.7/dist-packages (from bokeh>=1.0.0->hvplo
     Requirement already satisfied: typing-extensions>=3.7.4 in /usr/local/lib/python3.7/dist-packages (from bokeh>=
     Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.7/dist-packages (from bokeh>=1.0.0->hvplo
     Requirement already satisfied: packaging>=16.8 in /usr/local/lib/python3.7/dist-packages (from bokeh>=1.0.0->hv
     Requirement already satisfied: pyct>=0.4.4 in /usr/local/lib/python3.7/dist-packages (from colorcet>=2->hvplot)
     Requirement already satisfied: param>=1.7.0 in /usr/local/lib/python3.7/dist-packages (from colorcet>=2->hvplot
     Requirement already satisfied: pyviz-comms>=0.7.4 in /usr/local/lib/python3.7/dist-packages (from holoviews>=1.
     Requirement already satisfied: panel>=0.8.0 in /usr/local/lib/python3.7/dist-packages (from holoviews>=1.11.0->
     Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (from Jinja2>=2.9->bo
     Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packagi
     Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas->hvplot) (20
     Requirement already satisfied: tqdm>=4.48.0 in /usr/local/lib/python3.7/dist-packages (from panel>=0.8.0->holov
     Requirement already satisfied: bleach in /usr/local/lib/python3.7/dist-packages (from panel>=0.8.0->holoviews>=
     Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from panel>=0.8.0->holoviews
     Requirement already satisfied: markdown in /usr/local/lib/python3.7/dist-packages (from panel>=0.8.0->holoviews
     Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.1->b
     Requirement already satisfied: webencodings in /usr/local/lib/python3.7/dist-packages (from bleach->panel>=0.8.
     Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-packages (from markdown
     Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata>=4.
     Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-package
     Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->pane
     Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->panel>=0.
     Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests->pan
     Installing collected packages: hvplot
     Successfully installed hvplot-0.7.3
```

Data frame of Daily Visitors Data at the 40 venues of Crisco Company

```
daily_visitors_data = pd.read_csv('https://raw.githubusercontent.com/ChrisWalshaw/DataViz/master/Data/001203958/VenueDailyVisitors.csv',
         index_col=0)
daily_visitors_data.index = pd.to_datetime(daily_visitors_data.index)

daily_visitors_data
```

	CQC	ZJB	PDT	BQV	QRY	QJL	YRU	XXO	WXV	ZPL	...	WDZ	ZLH	RDA	SPF	XPE	UZO	GLQ	TLJ	AEQ	BKI
Date																					
2019-01-01	208	0	323	0	259	253	64	99	82	55	...	85	64	603	513	87	69	58	72	0	0

The daily number of visitors at each venue of the company is represented as one data frame as shown above

Data frame for the summary data of the given csv files

```
daily_visitors_data=pd.read_csv('https://raw.githubusercontent.com/ChrisWalshaw/DataViz/master/Data/001203958/VenueDailyVisitors.csv',
         index_col=0)
avg_age_of_visitors=pd.read_csv('https://raw.githubusercontent.com/ChrisWalshaw/DataViz/master/Data/001203958/VenueAge.csv',index_col=0)

max_travel_distance=pd.read_csv('https://raw.githubusercontent.com/ChrisWalshaw/DataViz/master/Data/001203958/VenueDistance.csv',index_cc

avg_visit_duration=pd.read_csv('https://raw.githubusercontent.com/ChrisWalshaw/DataViz/master/Data/001203958/VenueDuration.csv',index_col

female_proportion_data= pd.read_csv('https://raw.githubusercontent.com/ChrisWalshaw/DataViz/master/Data/001203958/VenueGender.csv', inde›

avg_spend_by_visitors_data = pd.read_csv('https://raw.githubusercontent.com/ChrisWalshaw/DataViz/master/Data/001203958/VenueSpend.csv', i

summary_data =pd.DataFrame(index=daily_visitors_data.columns)

summary_data['daily_visitors_data'] = daily_visitors_data.sum().values
summary_data['avg_age_of_visitors'] = avg_age_of_visitors.values
summary_data['max_travel_distance'] = max_travel_distance.values
summary_data['avg_visit_duration'] =  avg_visit_duration.values
summary_data['female_proportion_data'] = female_proportion_data.values
summary_data['avg_spend_by_visitors_data'] = avg_spend_by_visitors_data.values

print(summary_data.head(10))
```

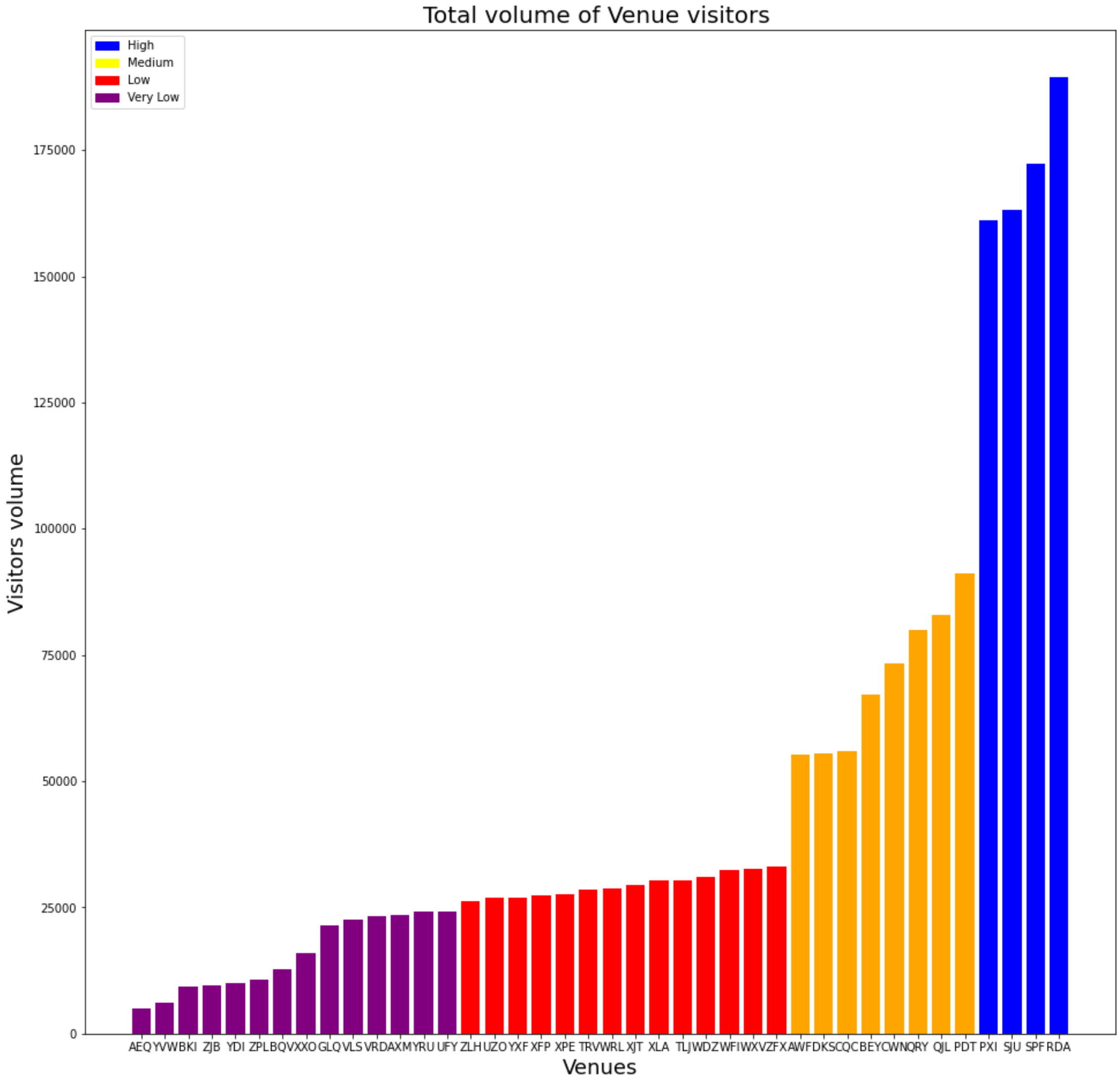
```
     daily_visitors_data  avg_age_of_visitors  max_travel_distance  \
CQC                55899                   46                   24
ZJB                 9655                   29                    4
PDT                91256                   50                   24
BQV                12793                   32                    5
QRY                79865                   31                   16
QJL                83005                   44                   26
YRU                24098                   39                    8
XXO                16056                   27                    7
WXV                32668                   26                    5
ZPL                10787                   33                    5

     avg_visit_duration  female_proportion_data  avg_spend_by_visitors_data
CQC                  72                      48                          31
ZJB                 108                      50                          20
PDT                 136                      52                          20
BQV                 110                      48                          20
QRY                 103                      47                          17
QJL                 103                      49                          23
YRU                 106                      44                          20
XXO                 120                      48                          19
WXV                 109                      41                          18
ZPL                  94                      54                          19
```

All the variables associated with the daily visitor's data such as average visit duration, female proportion data, maximum distance travelled, average amount spent and average age of the visitors in csv file is represented as a single sumary data as above.

1. Bar Chart to represent the total volume of visitors at the 40 venues sorted in ascending order

```
daily_visitors_data = daily_visitors_data.reindex(daily_visitors_data.sum().sort_values(ascending=True).index, axis=1)

colours = []
for name in daily_visitors_data.columns:
    total_volume = daily_visitors_data[name].sum()
    if total_volume > 100000:
        colour = 'blue'
    elif total_volume > 50000:
        colour = 'orange'
    elif total_volume > 25000:
        colour = 'red'
    else:
        colour = 'purple'
    colours.append(colour)
```

```
plt.figure(figsize=(16,16))
x_pos = np.arange(len(daily_visitors_data.columns))
plt.bar(x_pos, daily_visitors_data.sum(), align='center', color=colours)
plt.xticks(x_pos, daily_visitors_data.columns)
plt.xlabel('Venues', fontsize=18)
plt.ylabel('Visitors volume', fontsize=18)
plt.title('Total volume of Venue visitors ', fontsize=20)

colors = {'High':'blue', 'Medium':'yellow', 'Low':'red','Very Low':'purple'}
labels = list(colors.keys())
handles = [plt.Rectangle((0,0),1,1, color=colors[label]) for label in labels]
plt.legend(handles, labels)

plt.show()
```



Bar chart is used here to show the complete data of the visitor’s volume at all its venues, sorted in ascending order according to the maximum number of visitors in each venue.

2. To categorize the dataset to High, Medium, Low and Very low volume of visitors

```
daily_visitors_data = daily_visitors_data.reindex(daily_visitors_data.sum().sort_values(ascending=True).index, axis=1)

categories = ['High', 'Medium', 'Low', 'Very low']
categories_selected = [[] for i in range(len(categories))]
for name in daily_visitors_data.columns:
    total_volume = daily_visitors_data[name].sum()
    if total_volume > 100000:
        category = 0
    elif total_volume > 50000:

        category = 1
    elif total_volume > 25000:

        category = 2
    else:
        category = 3
    categories_selected[category].append(name)
    print('Venue ' + name + ' has ' + categories[category] + ' volume of visitors')
counter = 1
fig = plt.figure(figsize=(15, 15))
fig.suptitle('Visitors by category', fontsize=16, position=(0.5, 1.02))
for i in range(len(categories)):
    print(categories[i] + ': ' + str(categories_selected[i]))


for i, selected in enumerate(categories_selected):
    sub = fig.add_subplot(4, 4, counter)
    sub.set_title(categories[i], fontsize=10)
    x_pos = np.arange(len(daily_visitors_data[selected].columns))
    plt.bar(x_pos, daily_visitors_data[selected].sum(), align='center')
    plt.xticks(x_pos, daily_visitors_data[selected].columns)
    plt.xlabel('Venue', fontsize=10)
    plt.ylabel('Visitors at each venue', fontsize=10)
    counter += 1
plt.subplots_adjust(wspace=1.0, hspace=0.7)
plt.tight_layout()
plt.show()
```

```
Venue AEQ has Very low volume of visitors
Venue YVW has Very low volume of visitors
Venue BKI has Very low volume of visitors
Venue ZJB has Very low volume of visitors
Venue YDI has Very low volume of visitors
Venue ZPL has Very low volume of visitors
Venue BQV has Very low volume of visitors
Venue XXO has Very low volume of visitors
Venue GLQ has Very low volume of visitors
Venue VLS has Very low volume of visitors
Venue VRD has Very low volume of visitors
```

The data is further categorized as High, Medium, Low and very low volume of visitors at each of the company venues as shown above.

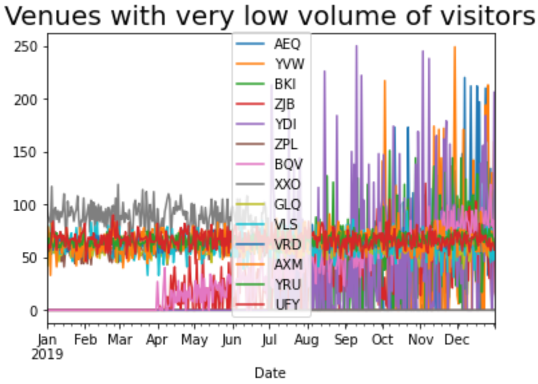
```
Venue UFY has Very low volume of visitors
```

To find all the venues that were opened newly and those that were closed during the year

```
Venue XFP has Low volume of visitors
```

```
daily_visitors__data = pd.read_csv('https://raw.githubusercontent.com/ChrisWalshaw/DataViz/master/Data/001203958/VenueDailyVisitors.csv',
daily_visitors__data.index = pd.to_datetime(daily_visitors__data.index)

selected  =  ['AEQ', 'YVW', 'BKI', 'ZJB', 'YDI', 'ZPL', 'BQV', 'XXO', 'GLQ', 'VLS', 'VRD', 'AXM', 'YRU', 'UFY']
daily_visitors__data = daily_visitors__data[selected]

daily_visitors__data = daily_visitors__data.loc[pd.to_datetime('2019-01-01'): pd.to_datetime('2019-12-31')]
print(daily_visitors__data.head())

daily_visitors__data.plot.line()
plt.title('Venues with very low volume of visitors', fontsize=20)
plt.show()
```

```
            AEQ  YVW  BKI  ZJB  YDI  ZPL  BQV  XXO  GLQ  VLS  VRD  AXM  YRU  \
Date
2019-01-01    0    0    0    0    0   55    0   99   58   55   50   66   64
2019-01-02    0    0    0    0    0   62    0   89   42   63   65   79   68
2019-01-03    0    0    0    0    0   54    0   86   67   64   61   68   56
2019-01-04    0    0    0    0    0   55    0   82   65   81   59   33   73
2019-01-05    0    0    0    0    0   59    0  117   69   53   57   82   64

            UFY
Date
2019-01-01   61
2019-01-02   56
2019-01-03   62
2019-01-04   59
2019-01-05   73
```

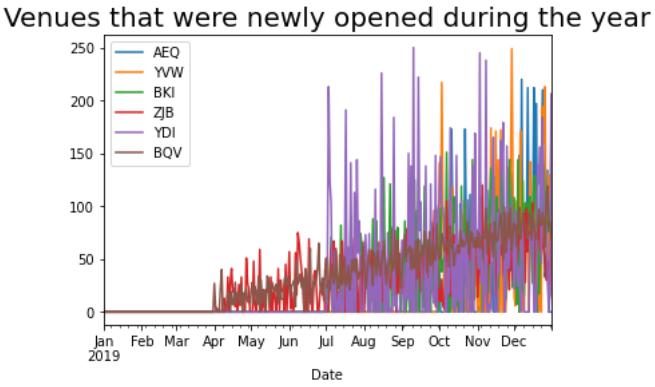


From the above plot , only those venues that were newly opened got closed during the year are separately visualized below

Visualizing only those venues that were newly opened during the year

```
daily_visitors__data = pd.read_csv('https://raw.githubusercontent.com/ChrisWalshaw/DataViz/master/Data/001203958/VenueDailyVisitors.csv',
daily_visitors__data.index = pd.to_datetime(daily_visitors__data.index)

selected =  ['AEQ', 'YVW', 'BKI', 'ZJB', 'YDI', 'BQV']
daily_visitors__data = daily_visitors__data[selected]

daily_visitors__data = daily_visitors__data.loc[pd.to_datetime('2019-01-01'): pd.to_datetime('2019-12-31')]
print(daily_visitors__data.head())
print(daily_visitors__data.tail())

daily_visitors__data.plot.line()
plt.title('Venues that were newly opened during the year', fontsize=20)
plt.show()
```

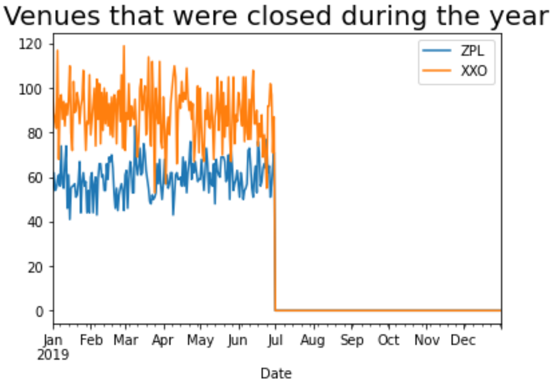
```
            AEQ  YVW  BKI  ZJB  YDI  BQV
Date
2019-01-01    0    0    0    0    0    0
2019-01-02    0    0    0    0    0    0
2019-01-03    0    0    0    0    0    0
2019-01-04    0    0    0    0    0    0
2019-01-05    0    0    0    0    0    0
            AEQ  YVW  BKI  ZJB  YDI  BQV
Date
2019-12-27   42   58   73   30   82  104
2019-12-28  134   49   85   48   43  119
2019-12-29    0  130   76    9    0   80
2019-12-30    0   41   92   17   17   85
2019-12-31    0  159  113   71  206   69
```



Visualizing the venues that got closed during the year

```
daily_visitors__data = pd.read_csv('https://raw.githubusercontent.com/ChrisWalshaw/DataViz/master/Data/001203958/VenueDailyVisitors.csv',
daily_visitors__data.index = pd.to_datetime(daily_visitors__data.index)

selected =  ['ZPL','XXO']
daily_visitors__data = daily_visitors__data[selected]

daily_visitors__data = daily_visitors__data.loc[pd.to_datetime('2019-01-01'): pd.to_datetime('2019-12-31')]
print(daily_visitors__data.head())
print(daily_visitors__data.tail())
daily_visitors__data.plot.line()
plt.title('Venues that were closed during the year', fontsize=20)
plt.show()
```

```
            ZPL  XXO
Date
2019-01-01   55   99
2019-01-02   62   89
2019-01-03   54   86
2019-01-04   55   82
2019-01-05   59  117
            ZPL  XXO
Date
2019-12-27    0    0
2019-12-28    0    0
2019-12-29    0    0
2019-12-30    0    0
2019-12-31    0    0
```



3. Correlation between the venues with high and medium volume of visitors using heatmap

```
selected = daily_visitors_data.columns[daily_visitors_data.sum() > 50000]
print(daily_visitors_data[selected].head())

plt.figure(figsize=(10, 10))
corr = daily_visitors_data[selected].corr()
ax = sns.heatmap(corr, vmin=-1, vmax=1, center=0, cmap=sns.diverging_palette(220, 20, n=200), square=True, annot=True,
                 annot_kws={"size": 8})
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, horizontalalignment='right')
plt.title('Heatmap showing correlation between high and medium volume venue visitors ', fontsize=12)
plt.show()
```

Correlation is used to establish the relationship between the two venues and heat map is applied to visualize the same

4. Visualisation showing histogram subplots of the venue with high volume of visitors with x_min and x_max determined

```
selected = ['RDA','SPF','SJU','PXI']

x_min = 150
x_max = 750
bin_width = 40

n_bins = int((bin_width + x_max - x_min) / bin_width)
print(str(n_bins) + ' bins')
bins = [(x_min + x * (bin_width + x_max - x_min) / n_bins) for x in range(int(n_bins))]

fig = plt.figure(figsize=(8, 8))
fig.suptitle('Venue with High Volume of Visitors', fontsize=20, position=(0.5, 1.0))
counter = 1
for name in selected:
    sub = fig.add_subplot(2, 2, counter)
    sub.hist(daily_visitors_data[name], bins, edgecolor='w')
    sub.set_title('Venue ' + name, fontsize=10)
    sub.set_xlim(xmin=x_min, xmax=x_max)
    sub.set_ylim(ymin=0, ymax=100)
    counter += 1
plt.subplots_adjust(wspace=0.4, hspace=0.4)
plt.show()
```

x min and x max are determined from the give data and venue with high volume of visitors is analysed using histogramm subplots

5.Visualizaion of radar sub plot for four selected venues with low volume of visitors

```
normalised_data = summary_data / summary_data.max()
print(normalised_data.head())

selected = ['ZLH', 'UZO', 'YXF', 'XFP']

n_attributes = len(normalised_data.columns)
angles = [n / float(n_attributes) * 2 * np.pi for n in range(n_attributes + 1)]
fig=plt.figure(figsize=(10,10))
fig.suptitle(' Radar subplots of four venues with low volume visitors', fontsize=15, position=(0.5, 1.02))
counter = 1
for name in selected:
    values = normalised_data.loc[[name]].values.flatten().tolist()
    values += values[:1]
    sub = plt.subplot(2, 2, counter, polar=True)
    sub.plot(angles, values)
    sub.fill(angles, values, alpha=0.1)
    sub.set_ylim(ymax=1.05)
    sub.set_yticks([0.2, 0.4, 0.6, 0.8, 1.0])
    sub.set_xticks(angles[0:-1])
    sub.set_xticklabels(normalised_data.columns, fontsize=8)
    sub.set_title('venue ' + name, fontsize=12, loc='center')
    counter += 1
plt.tight_layout()
plt.show()
```

The above plot shows the comparisons of the variables associated with the venues having low volume of visitors The same is plotted using radar plots.

Scatter plot visualization of venue with High volume of visitors to detect correlation

```
selected = ['RDA','SPF','SJU','PXI']

counter = 1
fig = plt.figure(figsize=(8, 8))
fig.suptitle('correlation of venues with high volume of visitors', fontsize=16, position=(0.5, 1.0))
for i, name_i in enumerate(selected):
    for j in range(i + 1, len(selected)):
        name_j = selected[j]
        sub = fig.add_subplot(4, 4, counter)
        sub.set_title(name_i + ' vs ' + name_j, fontsize=10)
        sub.scatter(daily_visitors_data[name_i], daily_visitors_data[name_j], s=0.5)
        counter += 1
plt.subplots_adjust(wspace=0.5, hspace=0.5)
plt.tight_layout()
plt.show()
```

6.Line-subplot visualization showing all the venues with high and medium volume of visitors to observe any quarterly or large scale seasonality

```
selected = ['RDA','SPF','SJU','PXI'] + ['AWF', 'DKS', 'CQC', 'BEY', 'CWN', 'QRY', 'QJL', 'PDT']
counter = 1
fig = plt.figure(figsize=(8, 8))
fig.suptitle('High and Medium volume Venue visitors distribution', fontsize=20, position=(0.5, 1.0))
for name in selected:
    sub = fig.add_subplot(4,4, counter)
    sub.set_title('Venue ' + name, fontsize=10)
    sub.plot(daily_visitors_data.index, daily_visitors_data[name], linewidth=0.5)
    sub.axes.get_xaxis().set_ticks([])  # remove the x ticks
    counter += 1
plt.subplots_adjust(wspace=0.4, hspace=0.4)
plt.show()
```

To understand large scale or quarterly seasonality in the venues with high and medium volume of visitors line plot is applied.Volume of visitors does not show much variations and do not exhibit much seasonality.

7. Interactive visualization with buble plot by making use of the summary data and comparison between the Female proportion to the average visit duration along with the daily visitorsa data

```
summary_data['BubbleSize'] = summary_data['daily_visitors_data'] * 0.01

plot = summary_data.hvplot.scatter(
    frame_height=500, frame_width=500,
    title='Female Proportion  vs Average visit duration (vs Daily visitors data)',
    xlabel='Avg visit duration (hr/unit)', ylabel='Female proportion (%/unit)',
    alpha=0.5, padding=0.1, hover_cols='all',
    tools=['pan', 'box_zoom', 'wheel_zoom', 'undo', 'redo', 'hover', 'save', 'reset'],
    x='avg_visit_duration', y='female_proportion_data', size='BubbleSize'
)
```

```
hv.extension('bokeh')
plot
```

Above is the comparison of the female proportion data to the average visit duration and the daily visitors data .Interaction is used here to show the access enabled for the user to understand the graph better.

Tools available include the following: PAN to adjust the plot to get the focused part of the graph, BOX ZOOM creates a box over the area of plot reqired and gives a zoom version, WHEEL ZOOM:zooms in and out of the graph with the control of the mouse wheel, HOVER helps to hover on a particular point and retrive the data at that point, SAVE enables the user to download the plot, RESET to reset the graph, UNDO to undo the previously performed action, REDO to redo the latter operation

8. Interactive visualisation for superimposed histograms of the distribution of two venues with high volume of visitors with customised bin sizes

```
selected = ['RDA','SPF']

x_min = 250
x_max = 550
bin_width = 10
n_bins = int((bin_width + x_max - x_min) / bin_width)
print(str(n_bins) + ' bins')
bins = [(x_min + x * (bin_width + x_max - x_min) / n_bins) for x in range(int(n_bins))]

plot = daily_visitors_data[selected].hvplot.hist(
    frame_height=500, frame_width=500,
    xlabel='Visitors volume per day', ylabel='freuency',
    title='Venue with High Volume of visitors ',
    alpha=0.5, muted_alpha=0, muted_fill_alpha=0, muted_line_alpha=0,
    tools=['pan', 'box_zoom', 'wheel_zoom', 'undo', 'redo', 'hover', 'save', 'reset'],
    bins=bins
)
hv.extension('bokeh')
plot
```

Two venues from the list of hih volume venues is chosen and superimposed histogram is applied above.

Tools available include the following: PAN to adjust the plot to get the focused part of the graph, BOX ZOOM creates a box over the area of plot reqired and gives a zoom version, WHEEL ZOOM:zooms in and out of the graph with the control of the mouse wheel, HOVER helps to hover on a particular point and retrive the data at that point, SAVE enables the user to download the plot, RESET to reset the graph, UNDO to undo the previously performed action, REDO to redo the latter operation