```python
import csv
import os
import pandas as pd
import matplotlib

matplotlib.use('Agg')
import matplotlib.pyplot as plt
import math

sourcedir = './ExtractedFiles/2019/'
tmpdir = './tmp/2019/'
CSV_IN = sourcedir + 'daVinci_MAUDE_Classified_2000_2019.xls'
img_dir = './curves/2019'
if not os.path.exists(img_dir):
    os.makedirs(img_dir)

colors = ['yellowgreen',
          'lightskyblue',
          'g',  # green
          'c',  # cyan
          'm',  # magenta
          'y',  # yellow
          'b',  # blue
          'r',  # red
          'k',  # black
          'w'   # white
          ]


def fun1():
    """
    Analysis each event type's percentage
    :return:
    """
    df = pd.read_excel(CSV_IN)
    df_new = df[['Patient Impact', 'Injuries', 'Surgery_Type',
      'Surgery_Class']]
    Event_type = df_new['Patient Impact'].groupby(by=df_new['Patient
      Impact'])
    event_dict = {}
    event_map = {'M': 'Malfunction', 'D': 'Death', 'IN': 'Injury', 'O':
      'Other'}
    labels = []
    nums = []
    colormaps = []
    explode = []
    for i, (name, group) in enumerate(Event_type):
        print(name)
        print(len(group.index))
        labels.append(event_map[name])
        nums.append(len(group.index))
        colormaps.append(colors[i])
        if i == 0:
            explode.append(0.05)
        else:
```

```python
            explode.append(0)
        # event_dict[name] = len(group.index)
    patches, l_text, p_text = plt.pie(nums, labels=labels,
     colors=colormaps, autopct='%3.1f%%', explode=explode)
    for t in l_text:
        t.set_size = (30)
    for t in p_text:
        t.set_size = (20)
    plt.axis('equal')
    plt.legend()
    plt.title('the distribution of event type')
    plt.savefig(os.path.join(img_dir, 'event_type_pie.jpg'))


def fun2():
    df = pd.read_excel(CSV_IN)
    Sugery_Class = df['Surgery_Class'].groupby(df['Surgery_Class'])
    labels = []
    nums = []
    colormaps = []
    explode = []
    for i, (name, group) in enumerate(Sugery_Class):
        print(name)
        print(len(group.index))
        labels.append(name)
        nums.append(len(group.index))
        colormaps.append(colors[i])
        if i == 0:
            explode.append(0.03)
        else:
            explode.append(0)
    patches, l_text, p_text = plt.pie(nums, labels=labels,
     colors=colormaps, autopct='%3.1f%%', explode=explode)
    for t in l_text:
        t.set_size = (30)
    for t in p_text:
        t.set_size = (20)
    plt.axis('equal')
    plt.legend()
    plt.title('the distribution of sugery class')
    plt.savefig(os.path.join(img_dir, 'sugery_class_pie.jpg'))


def fun3():
    df = pd.read_excel(CSV_IN)
    Sugery_Class = df['Surgery_Class'].groupby(df['Surgery_Class'])

    Event_Type = df['Patient Impact'].groupby(df['Surgery_Class'])
    n_class = len(Event_Type)
    fig, axes = plt.subplots(nrows=int(math.ceil(n_class / 3)), ncols=3,
     figsize=(12, 12))
    print(int(math.ceil(n_class / 3)))
    for i, (sugery_class, group) in enumerate(Event_Type):
        group = group.groupby(group)
        r = i // 3
```

```python
            c = i - r * 3
            event_map = {'M': 'Malfunction', 'D': 'Death', 'IN': 'Injury', 'O':
             'Other'}
            labels = []
            nums = []
            colormaps = []
            explode = []
            for j, (event_type, group2) in enumerate(group):
                print(event_type)
                print(len(group2.index))
                labels.append(event_map[event_type])
                nums.append(len(group2.index))
                colormaps.append(colors[j])
                if j == 0:
                    explode.append(0.05)
                else:
                    explode.append(0)
            patches, l_text, p_text = axes[r][c].pie(nums, labels=labels,
             colors=colormaps, autopct='%3.1f%%',
                                                     explode=explode)
            for t in l_text:
                t.set_size = (20)
            for t in p_text:
                t.set_size = (20)
            axes[r][c].axis('equal')
            axes[r][c].legend(loc='upper left')

            # axes[r][c].spines['top'].set_visible(False)
            # axes[r][c].spines['right'].set_visible(False)
            # axes[r][c].spines['bottom'].set_visible(False)
            # axes[r][c].spines['left'].set_visible(False)
            print(type(str(sugery_class)))
            axes[r][c].set_title(sugery_class)
        fig.savefig(os.path.join(img_dir, 'event_type_sugery_class_pie.jpg'))

    pass


if __name__ == '__main__':
    fun3()
    fun2()
    fun1()
```