```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
pd.set_option('display.max_rows', 100)
pd.set_option('display.max_columns', 100)
import matplotlib.pyplot as plt
import seaborn as sns
import re
```

Data Sample

2018年1-2月保险业经营情况表

时间：2018年04月03日

	单位:万元
原保险保费收入	97034372.62
1、财产险	17721912.27
2、人身险	79312460.35
（1）寿险	69017036.90
（2）健康险	8749972.64
（3）人身意外伤害险	1545450.80
人身保险公司保户投资款新增交费	25134457.09
人身保险公司投连险独立账户新增交费	1509510.17
原保险赔付支出	22556676.55
1、财产险	8241803.44
2、人身险	14314873.10
（1）寿险	11268009.75
（2）健康险	2637951.58
（3）人身意外伤害险	408911.78
业务及管理费	6151768.99
银行存款	190830329.04
投资	1309099035.19
资产总额	1688522210.14

数据预处理

#读取数据

```
import os,sys
os.listdir()
```

```
['visualization.ipynb', 'month_data.xlsx', '.ipynb_checkpoints']
```

```
data = pd.read_csv('../data/industry_data.csv')
#data.info()
```

替换中文符号

```
columns = data.columns.to_list()
columns = [re.sub('_\d、','',x) for x in columns]
columns = [re.sub('_(\d)','',x) for x in columns]
columns
data.columns = columns
```

```
def create_nan_2015(x):
    tmp = {}
    for i in data.columns:
        if i == 'Year':
            tmp[i] = 2015
        elif i == 'Month':
            tmp[i] = f'1-{x}'
        elif i == 'date_time':
            tmp[i] = f'2011年1-{x}月'
        else:
            tmp[i] = np.nan
    return tmp
```

```
def create_nan_2011(x):
    tmp = {}
    for i in data.columns:
        if i == 'Year':
            tmp[i] = 2011
        elif i == 'Month':
            tmp[i] = f'1-{x}'
        elif i == 'date_time':
            tmp[i] = f'2011年1-{x}月'
        else:
            tmp[i] = np.nan
    return tmp
```

正则读取月份年份

```
print(data.shape)
data.drop('from_url',axis = 1,inplace=True)
data = data.drop_duplicates()
print(data.shape)
data['date_time'] = data['info'].apply(lambda x: re.findall(r'.*(?=保险.*)',x)[0])
data['date_time'] = data['date_time'].replace({'2011年': '2011年1-12月',
                            '2012年': '2012年1-12月',
                            '2013年': '2013年1-12月',
                            '2014年': '2014年1-12月',
                            '2015年': '2015年1-12月',
                            '2016年': '2016年1-12月',
                            '2017年': '2017年1-12月',
                            '2011年2月':'2011年1-2月',
                            '2011年3月':'2011年1-3月',
                            '2011年6月':'2011年1-6月',
                            '2011年7月':'2011年1-7月'})

data['Year'] = data.date_time.apply(lambda x: int(re.findall(r'\d{4}(?=年)',x)[0]))
data['Month'] = data.date_time.apply(lambda x: re.findall(r'(?<=年).*(?=月)',x)[0])
data.drop('info',inplace= True,axis = 1)
month_cat = ['1'] + [f'1-{i}' for i in range(2,13)]
#data = data.append(create_nan_2011(4),ignore_index=True)
#data = data.append(create_nan_2011(5),ignore_index=True)
#data = data.append(create_nan_2015(2),ignore_index=True)
#data = data.append(create_nan_2015(3),ignore_index=True)
data['Month'] = pd.Categorical(data['Month'],ordered = True,categories=month_cat)
data = data.sort_values(by=['Year','Month'])
```

```
(107, 23)
(106, 22)
```

读取所有的浮点型数据并interpolate缺失值

```
numerical_col = data.select_dtypes(float).columns.tolist()
data[numerical_col] = data[numerical_col].interpolate(method='linear')
```

```
def cal_each_month(group):
    #year = re.findall(r'\d{4}(?=年)',group.loc[0,'date_time'])[0]
    #print(year)
    vals = group.loc[:,numerical_col]
    vals_lag = vals.shift(1).fillna(0)
    rV = vals - vals_lag

    #index = range(1,13)
    #index = [f'{year}/{i}' for i in range(1,13)]
    #rV['dd'] = index
    #columns = numerical_col
    return vals - vals_lag
dd = data.groupby('Year').apply(cal_each_month)
```

```
al = []
for x in data.Year.unique():

    if x == 2018:
        ls = ['2018-1','2018-2']
    else:
        ls = [f'{x}-{i}' for i in range(1,13)]
    al.extend(ls)
len(al)
al = pd.Series(al)
```

```
dd.insert(0,'date',al)
dd['date'] = pd.to_datetime(dd['date'])
```

```
dd = dd.sort_values('date',ascending=True)
```

```
len(data)
```

```
106
```

寻找缺失的月份

```
data.groupby('Year')['资产总额'].count()
```

```
Year
2009    12
2010    12
2011    10
2012    12
2013    12
2014    12
2015    10
2016    12
2017    12
2018     2
Name: 资产总额, dtype: int64
```

找到年份数据

```
Date_Year = (data['date_time'].str.contains('12月'))
data[Date_Year]
```

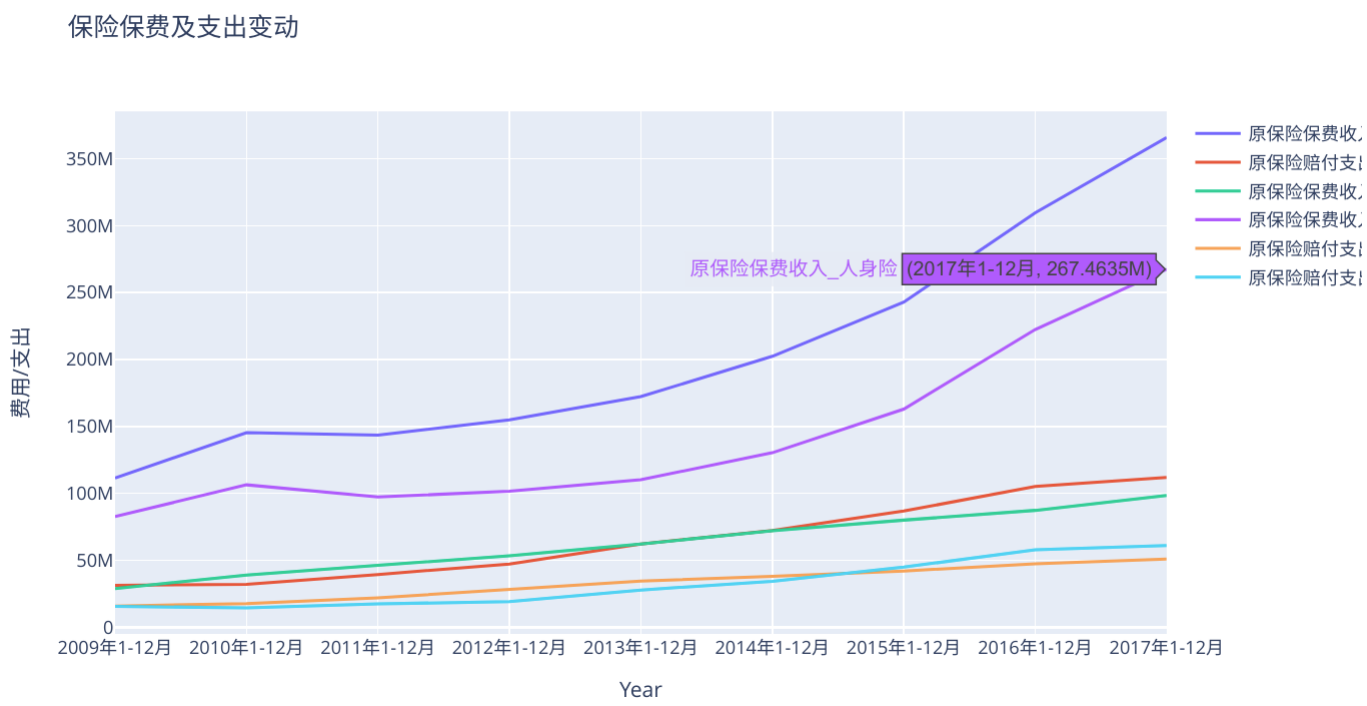
	原保险保费收入	原保险保费收入_财产险	原保险保费收入_人身险	原保险保费收入_人身险_寿险	原保险保费收入_人身险_健康险	原保险保费收入_人身险_人身意外伤害险	人身保险公司保户投资款新增交费	人身保险公司投连险独立账户新增交费	原保险赔付支出	原保险赔付支出_财产险	原保险赔付支出_人身险	原保险赔付支出_人身险_寿险	原保险赔付支出_人身险_健康险	原保险赔付支出_人身险_外
95	1.113730e+08	28758335.00	8.261465e+07	7.457438e+07	5739775.40	2300494.00	NaN	NaN	3.125483e+07	15757845.00	15496982.00	12687425.00	2170336.60	6392
83	1.452797e+08	38956424.71	1.063233e+08	9.679509e+07	6774658.47	2753537.94	NaN	NaN	3.200429e+07	17560273.50	14444018.14	11089892.04	2640219.87	7139
73	1.433925e+08	46178231.58	9.721428e+07	8.695591e+07	6917212.77	3341154.20	NaN	NaN	3.929373e+07	21869338.06	17424394.32	13009348.85	3596650.21	8183
61	1.548793e+08	53309273.47	1.015700e+08	8.908057e+07	8627607.13	3861847.73	NaN	NaN	4.716318e+07	28163316.38	18999868.23	15050143.88	2981707.45	9680
49	1.722224e+08	62122577.83	1.100998e+08	9.425141e+07	11234960.47	4613422.77	3.212318e+07	831751.31	6.212903e+07	34391379.13	27737651.16	22531329.39	4111271.11	1095
37	2.023481e+08	72033761.12	1.303143e+08	1.090169e+08	15871785.71	5425659.20	3.916754e+07	2894978.55	7.216212e+07	37882124.70	34279999.89	27284267.23	5711551.87	1284
26	2.428252e+08	79949687.36	1.628755e+08	1.324152e+08	24104715.15	6355584.95	7.646564e+07	6778832.75	8.674140e+07	41941665.57	44799731.92	35651680.05	7629656.85	1518
14	3.095910e+08	87244981.36	2.223460e+08	1.744222e+08	40424967.91	7498892.86	1.186016e+08	9389744.45	1.051289e+08	47261838.89	57867060.95	46029461.77	10007522.22	1830
2	3.658101e+08	98346579.05	2.674635e+08	2.145557e+08	43894603.83	9013240.68	5.892364e+07	4704175.97	1.118079e+08	50874495.97	60933436.60	45748906.94	12947670.22	2236

```
#data[Date_Year].date_time.tolist()
```

年份数据图表

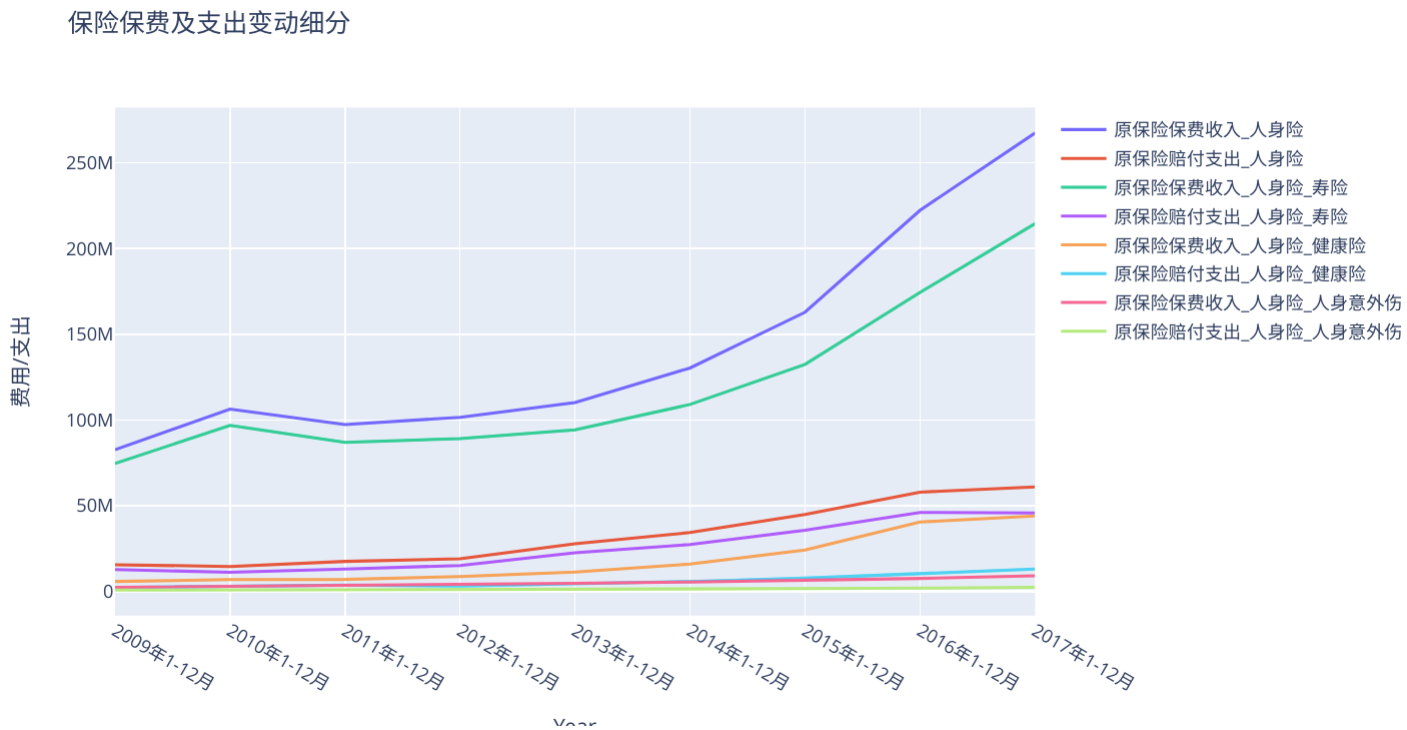
保险保费及支出变动 Line Chart

```
fig = go.Figure()
fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                         y=data[Date_Year].原保险保费收入.tolist(),
                         mode='lines',
                         name='原保险保费收入'))
fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                y=data[Date_Year].原保险赔付支出.tolist(),
                mode='lines',
                name = '原保险赔付支出'))

fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                y=data[Date_Year].原保险保费收入_财产险.tolist(),
                mode='lines',
                name = '原保险保费收入_财产险'))
fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                y=data[Date_Year].原保险保费收入_人身险.tolist(),
                mode='lines',
                name = '原保险保费收入_人身险'))
fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                y=data[Date_Year].原保险赔付支出_财产险.tolist(),
                mode='lines',
                name = '原保险赔付支出_财产险'))
fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                y=data[Date_Year].原保险赔付支出_人身险.tolist(),
                mode='lines',
                name = '原保险赔付支出_人身险'))


fig.update_layout(title = '保险保费及支出变动', xaxis_title='Year',yaxis_title='费用/支出')
#fig.show()
```



保险保费及支出变动细分 Line Chart

```
fig = go.Figure()
fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                y=data[Date_Year].原保险保费收入_人身险.tolist(),
                mode='lines',
                name = '原保险保费收入_人身险'))
fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                y=data[Date_Year].原保险赔付支出_人身险.tolist(),
                mode='lines',
                name = '原保险赔付支出_人身险'))
fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                y=data[Date_Year].原保险保费收入_人身险_寿险.tolist(),
                mode='lines',
                name = '原保险保费收入_人身险_寿险'))

fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                y=data[Date_Year].原保险赔付支出_人身险_寿险.tolist(),
                mode='lines',
                name = '原保险赔付支出_人身险_寿险'))

fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                y=data[Date_Year].原保险保费收入_人身险_健康险.tolist(),
                mode='lines',
                name = '原保险保费收入_人身险_健康险'))
fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                y=data[Date_Year].原保险赔付支出_人身险_健康险.tolist(),
                mode='lines',
                name = '原保险赔付支出_人身险_健康险'))
fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),
                y=data[Date_Year].原保险保费收入_人身险_人身意外伤害险.tolist(),
                mode='lines',
                name = '原保险保费收入_人身险_人身意外伤害险'))

fig.add_trace(go.Scatter(x=data[Date_Year].date_time.tolist(),

                y=data[Date_Year].原保险赔付支出_人身险_人身意外伤害险.tolist(),
                mode='lines',
                name = '原保险赔付支出_人身险_人身意外伤害险'))

fig.update_layout(title = '保险保费及支出变动细分', xaxis_title='Year',yaxis_title='费用/支出')
#fig.show()
```



原保险保费收入*财产险占比 Vs 原保险保费收入*人身险占比 Bar Chart

```
data_Year = data[Date_Year].copy()
```

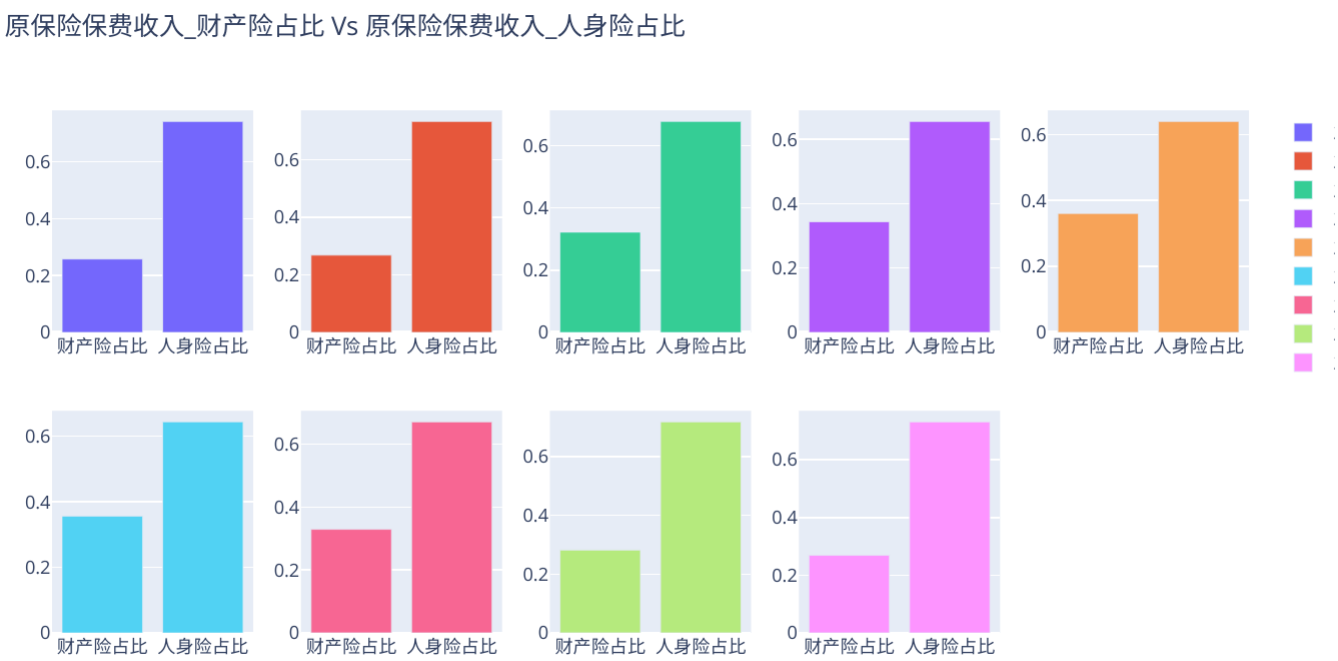
```
data_Year['原保险保费收入_财产险占比'] = data_Year['原保险保费收入_财产险']/ data_Year['原保险保费收入']
data_Year['原保险保费收入_人身险占比'] = data_Year['原保险保费收入_人身险']/ data_Year['原保险保费收入']
data_Year['原保险赔付支出_财产险占比'] = data_Year['原保险赔付支出_财产险']/ data_Year['原保险赔付支出']
data_Year['原保险赔付支出_人身险占比'] = data_Year['原保险赔付支出_人身险']/ data_Year['原保险赔付支出']
```

```
dict(zip(range(0,8),range(2009,2018)))
```

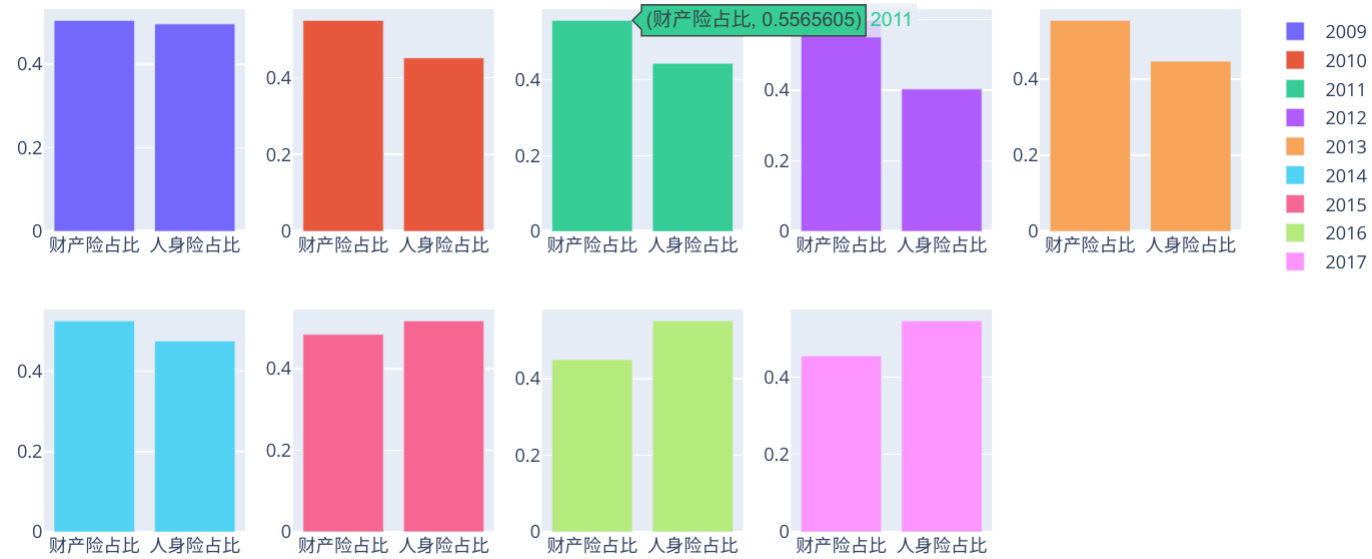
```
{0: 2009, 1: 2010, 2: 2011, 3: 2012, 4: 2013, 5: 2014, 6: 2015, 7: 2016}
```

```
Year_dict = dict(zip(range(0,9),range(2009,2018)))
'''
fig, axes = plt.subplots(2,5,figsize = (10,8))
for i in range(0,9):
    if i < 5:
        data_Year[data_Year['Year']==Year_dict[i]].loc[:,['原保险保费收入_财产险占比','原保险保费收入_人身险占比']].plot(kind='bar',ax = axes[0,i]);
        axes[0,i].legend(['Property','Individual'])
    else:
        data_Year[data_Year['Year']==Year_dict[i]].loc[:,['原保险保费收入_财产险占比','原保险保费收入_人身险占比']].plot(kind='bar',ax=axes[1,i-5]);
        axes[1,i-5].legend(['Property','Individual'])
plt.title('Recent Year Property Insurance Vs Individual Insurance')
'''
```

```
from plotly.subplots import make_subplots


fig = make_subplots(
    rows=2, cols=5,
    column_widths=[0.5]*5,
    row_heights=[0.5]*2,
    specs=[[{"type": "bar"}, {"type": "bar"},{"type": "bar"},{"type": "bar"},{"type": "bar"}],
           [{"type": "bar"},{"type": "bar"},{"type": "bar"},{"type": "bar"}, None]])
tra_ls = []
for i in range(5):
    x = ['财产险占比','人身险占比']
    y = data_Year[data_Year['Year']==Year_dict[i]].loc[:,['原保险保费收入_财产险占比','原保险保费收入_人身险占比']].squeeze(axis = 0).tolist()
    fig.add_trace(
    go.Bar(x=x,y=y, showlegend=True,name= Year_dict[i]),
    row=1, col =i+1
)

for i in range(5,9):
    x = ['财产险占比','人身险占比']
    y = data_Year[data_Year['Year']==Year_dict[i]].loc[:,['原保险保费收入_财产险占比','原保险保费收入_人身险占比']].squeeze(axis = 0).tolist()
    fig.add_trace(
    go.Bar(x=x,y=y, showlegend=True,name=Year_dict[i]
        ),
    row=2, col = i-4
    )
fig.update_layout(title = '原保险保费收入_财产险占比 Vs 原保险保费收入_人身险占比')
fig.show()
```



```
from plotly.subplots import make_subplots


fig = make_subplots(
    rows=2, cols=5,
    column_widths=[0.5]*5,
    row_heights=[0.5]*2,
    specs=[[{"type": "bar"}, {"type": "bar"},{"type": "bar"},{"type": "bar"},{"type": "bar"}],
           [{"type": "bar"},{"type": "bar"},{"type": "bar"},{"type": "bar"}, None]])
tra_ls = []
for i in range(5):
    x = ['财产险占比','人身险占比']
    y = data_Year[data_Year['Year']==Year_dict[i]].loc[:,['原保险赔付支出_财产险占比','原保险赔付支出_人身险占比']].squeeze(axis = 0).tolist()
    fig.add_trace(
    go.Bar(x=x,y=y, showlegend=True,name= Year_dict[i]),
    row=1, col =i+1
)

for i in range(5,9):
    x = ['财产险占比','人身险占比']
    y = data_Year[data_Year['Year']==Year_dict[i]].loc[:,['原保险赔付支出_财产险占比','原保险赔付支出_人身险占比']].squeeze(axis = 0).tolist()
    fig.add_trace(
    go.Bar(x=x,y=y, showlegend=True,name=Year_dict[i]
        ),
    row=2, col = i-4
    )
fig.update_layout(title = '原保险赔付支出_财产险占比 Vs 原保险赔付支出_人身险占比')
#fig.show()
```

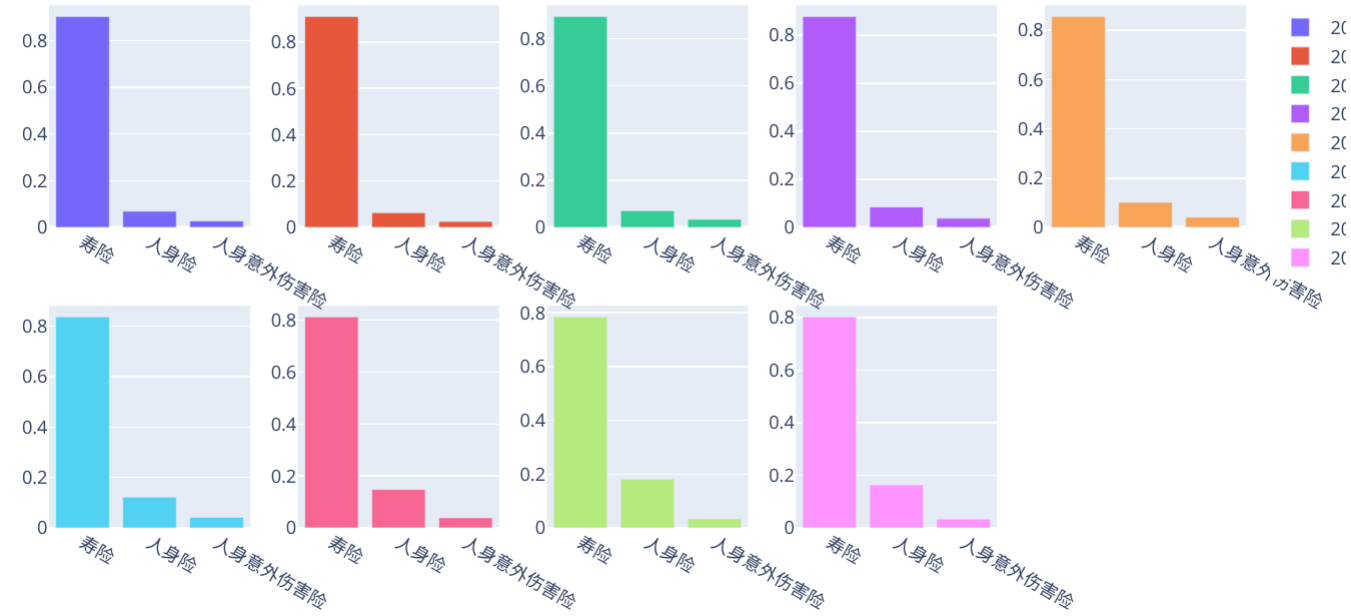


原保险保费收入-人身险细分 Bar Chart

```
data_Year['收入_寿险占比'] = data_Year['原保险保费收入_人身险_寿险']/ data_Year['原保险保费收入_人身险']
data_Year['收入_健康险占比'] = data_Year['原保险保费收入_人身险_健康险']/ data_Year['原保险保费收入_人身险']
data_Year['收入_人身意外伤害险占比'] = data_Year['原保险保费收入_人身险_人身意外伤害险']/ data_Year['原保险保费收入_人身险']
data_Year['赔付_寿险占比'] = data_Year['原保险赔付支出_人身险_寿险']/ data_Year['原保险赔付支出_人身险']
data_Year['赔付_健康险占比'] = data_Year['原保险赔付支出_人身险_健康险']/ data_Year['原保险赔付支出_人身险']
data_Year['赔付_人身险占比'] = data_Year['原保险赔付支出_人身险_人身意外伤害险']/ data_Year['原保险赔付支出_人身险']
```

```
from plotly.subplots import make_subplots


fig = make_subplots(
    rows=2, cols=5,
    column_widths=[0.5]*5,
    row_heights=[0.5]*2,
    specs=[[{"type": "bar"}, {"type": "bar"},{"type": "bar"},{"type": "bar"},{"type": "bar"}],
           [{"type": "bar"},{"type": "bar"},{"type": "bar"},{"type": "bar"}, None]])
tra_ls = []
for i in range(5):
    x = ['寿险','人身险','人身意外伤害险']
    y = data_Year[data_Year['Year']==Year_dict[i]].loc[:,['收入_寿险占比','收入_健康险占比', '收入_人身意外伤害险占比']].squeeze(axis = 0).tolist()
    fig.add_trace(
    go.Bar(x=x,y=y, showlegend=True,name= Year_dict[i]),
    row=1, col =i+1
)

for i in range(5,9):
    x = ['寿险','人身险','人身意外伤害险']
    y = data_Year[data_Year['Year']==Year_dict[i]].loc[:,['收入_寿险占比','收入_健康险占比', '收入_人身意外伤害险占比']].squeeze(axis = 0).tolist()
    fig.add_trace(
    go.Bar(x=x,y=y, showlegend=True,name=Year_dict[i]
        ),
    row=2, col = i-4
    )
fig.update_layout(title = '原保险保费收入-人身险')
#fig.show()
```

原保险保费收入-人身险

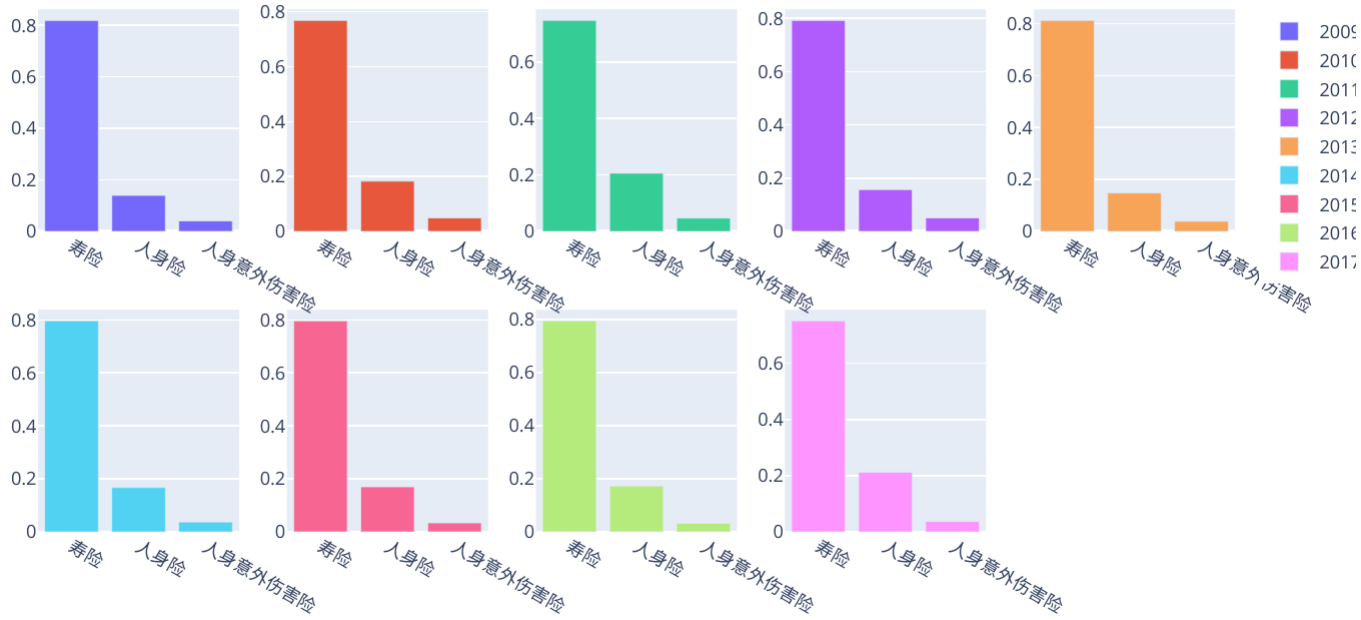


原保险赔付支出-人身险细分 Bar Chart

```
dic = {i:3 for i in ['收入_寿险占比','收入_健康险占比', '收入_人身意外伤害险占比', '赔付_寿险占比', '赔付_健康险占比', '赔付_人身险占比']}
d2 = data_Year.round(dic)
```

```
from plotly.subplots import make_subplots


fig = make_subplots(
    rows=2, cols=5,
    column_widths=[0.5]*5,
    row_heights=[0.5]*2,
    specs=[[{"type": "bar"}, {"type": "bar"},{"type": "bar"},{"type": "bar"},{"type": "bar"}],
           [{"type": "bar"},{"type": "bar"},{"type": "bar"},{"type": "bar"}, None]])

for i in range(5):
    x = ['寿险','人身险','人身意外伤害险']
    y = d2[d2['Year']==Year_dict[i]].loc[:,['赔付_寿险占比', '赔付_健康险占比', '赔付_人身险占比']].squeeze(axis = 0).tolist()
    fig.add_trace(
    go.Bar(x=x,y=y, showlegend=True,text=y,name= Year_dict[i]),
    row=1, col =i+1
)

for i in range(5,9):
    x = ['寿险','人身险','人身意外伤害险']
    y = d2[d2['Year']==Year_dict[i]].loc[:,['赔付_寿险占比', '赔付_健康险占比', '赔付_人身险占比']].squeeze(axis = 0).tolist()
    fig.add_trace(
    go.Bar(x=x,y=y, showlegend=True,text=y,name=Year_dict[i]
        ),
    row=2, col = i-4
    )
fig.update_layout(title = '原保险赔付支出-人身险')
#fig.show()
```

原保险赔付支出-人身险




```
dd.to_excel("month_data.xlsx")
```

```
#(data[data['Year']== 2009]['原保险保费收入'].reset_index(drop=True) - data[data['Year']== 2009]['原保险保费收入'].reset_index(drop=True).shift(1)).plot()
```

每月数据（对数据完整的年份）

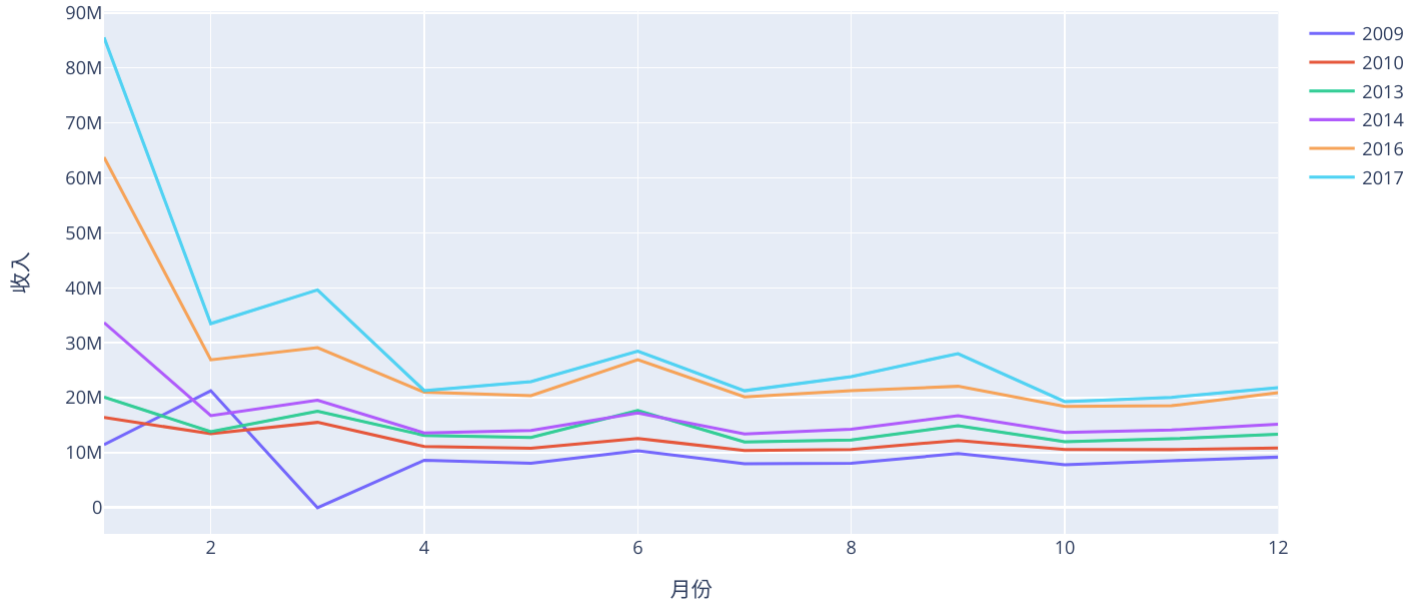
每月数据对比-->探究季节性

```
fig,ax = plt.subplots(1,1)
for i in range(2009,2018):
    if i not in [2011,2015,2018]:
        (data[data['Year']== i]['原保险保费收入'].reset_index(drop=True) - data[data['Year']== i]['原保险保费收入'].reset_index(drop=True).shift(1).fillna(0)).plot(ax=ax)
```

```
fig = go.Figure()
for i in range(2009,2018):
    if i not in [2011,2015,2018,2012]:
        x = list(range(1,13))
        y = (data[data['Year']== i]['原保险保费收入'].reset_index(drop=True) - data[data['Year']== i]['原保险保费收入'].reset_index(drop=True).shift(1).fillna(0)).tolist()
        fig.add_trace(go.Scatter(x=x,
                        y=y,
                        mode='lines',
                        name = f'{i} 年'))
fig.update_layout(title = '原保费收入（1-12月）', xaxis_title='月份',yaxis_title='收入')
#fig.show()
```

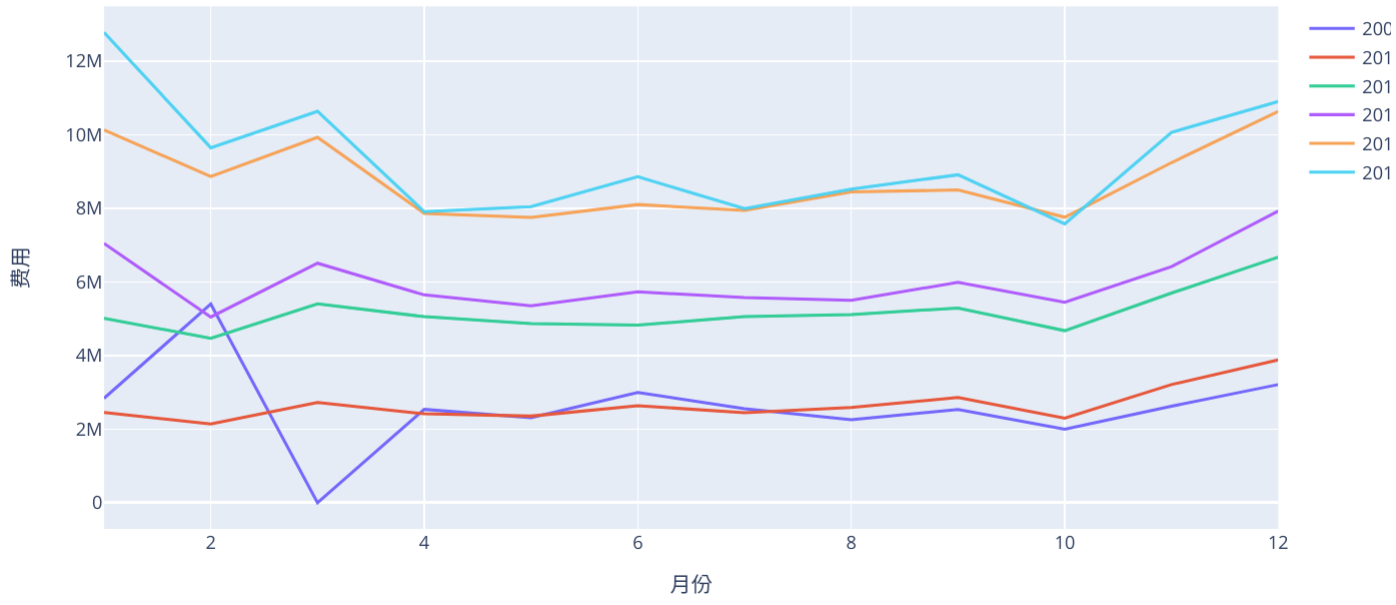
```
fig = go.Figure()
for i in range(2009,2018):
    if i not in [2011,2015,2018,2012]:
        x = list(range(1,13))
        y = (data[data['Year']== i]['原保险赔付支出'].reset_index(drop=True) - data[data['Year']== i]['原保险赔付支出'].reset_index(drop=True).shift(1).fillna(0)).tolist()
        fig.add_trace(go.Scatter(x=x,
                        y=y,
                        mode='lines',
                        name = f'{i} 年'))
fig.update_layout(title = '原赔偿支出（1-12月）', xaxis_title='月份',yaxis_title='费用')
#fig.show()
```

原保费收入（1-12月）




```
fig = make_subplots(
    rows=2, cols=1,
    column_widths=[0.8],
    row_heights=[1]*2,
    specs=[[{"type": "scatter"}],
           [{"type": "scatter"}]])

for i in range(2009,2018):
    if i not in [2011,2015,2018,2012]:
        x = list(range(1,13))
        y = (data[data['Year']== i]['原保险保费收入'].reset_index(drop=True) - data[data['Year']== i]['原保险保费收入'].reset_index(drop=True).shift(1).fillna(0)).tolist()
        fig.add_trace(go.Scatter(x=x,
                         y=y,
                         mode='lines',
                         name = f'{i} 年收入'),row=1, col = 1)
        #x = list(range(1,13))
        y = (data[data['Year']== i]['原保险赔付支出'].reset_index(drop=True) - data[data['Year']== i]['原保险赔付支出'].reset_index(drop=True).shift(1).fillna(0)).tolist()
        fig.add_trace(go.Scatter(x=x,
                         y=y,
                         mode='lines',
                         name = f'{i} 年赔偿'),row=2, col = 1)
fig.update_layout(title = '原赔偿支出/收入（1-12月）')
#fig.show()
```

原赔偿支出（1-12月）




```
#data[data['Month'] == '1-2']['原保险保费收入'].reset_index(drop = True).plot()
```

```
dd = dd.set_index(dd.date)
dd.drop('date',axis = 1, inplace = True)
dd
```

date	原保险保费收入	原保险保费收入_财产险	原保险保费收入_人身险	原保险保费收入_人身险_寿险	原保险保费收入_人身险_健康险	原保险保费收入_人身险_人身意外伤害险	人身保险公司保户投资款新增交费	人身保险公司投连险独立账户新增交费	原保险赔付支出	原保险赔付支出_财产险	原保险赔付支出_人身险	原保险赔付支出_人身险_寿险	原保险赔付支出_人身险_健康险	原保险赔付支出_人身险_人身意外伤害险	业
2009-01-01	28515180.84	6267249.25	22247931.60	18229582.85	3426596.51	591752.22	6571291.99	640352.80	9425589.20	3487974.39	5937614.80	4742044.97	1018015.41	177554.44	29
2009-02-01	68519191.78	11454663.02	57064528.75	50787454.05	5323376.13	953698.58	18563165.10	869157.37	13131087.35	4753829.05	8377258.30	6525964.78	1619936.17	231357.34	37
2009-03-01	21834240.41	10215849.49	11618390.92	8114247.06	2839168.92	664974.94	3562622.33	546535.65	10899470.02	6351780.65	4547689.37	2612003.40	1693594.41	242091.57	58
2009-04-01	20088153.48	8519367.55	11568785.93	8285150.00	2634552.47	649083.46	4595615.51	185568.97	10061710.95	5078804.34	4982906.61	3360538.07	1412272.14	210096.40	36
2009-05-01	19314506.78	7308527.70	12005979.08	8747082.26	2590382.70	668514.12	4240443.18	165568.18	7574094.70	3774005.48	3800089.22	2643289.54	995690.08	161109.60	29
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2017-07-01	8100320.57	2262270.46	5838050.11	5238381.00	425494.45	174174.66	NaN	NaN	2308018.05	1122376.76	1185641.29	987780.01	151685.45	46175.83	87
2017-08-01	8637567.71	2641230.23	5996337.48	5382843.45	423169.18	190324.86	NaN	NaN	2537911.19	1276252.03	1261659.17	1054955.93	155405.98	51297.26	90
2017-09-01	0.00	0.00	0.00	0.00	0.00	0.00	NaN	NaN	0.00	0.00	0.00	0.00	0.00	0.00	0.
2017-10-01	21290612.36	4212545.55	17078066.81	15755278.03	915913.00	406875.77	NaN	NaN	5398541.64	2262664.39	3135877.25	2684323.04	356861.36	94692.86	17
2017-11-01	11476413.23	2973274.39	8503138.84	7834667.90	488045.63	180425.31	NaN	NaN	2837258.58	1251807.76	1585450.82	1344880.23	183986.72	56583.86	83

106 rows × 18 columns

```
dd[dd.index.year == 2015]
```

date	原保险保费收入	原保险保费收入_财产险	原保险保费收入_人身险	原保险保费收入_人身险_寿险	原保险保费收入_人身险_健康险	原保险保费收入_人身险_人身意外伤害险	人身保险公司保户投资款新增交费	人身保险公司投连险独立账户新增交费	原保险赔付支出	原保险赔付支出_财产险	原保险赔付支出_人身险	原保险赔付支出_人身险_寿险	原保险赔付支出_人身险_健康险	原保险赔付支出_人身险_人身意外伤害险	业务及管理费	
2015-01-01	18917623.69	4946729.23	13970894.45	12864408.33	809343.16	297142.96	NaN	NaN	3453928.94	1714885.64	1739043.30	1454020.09	217777.90	67245.31	1542547.07	1.7
2015-02-01	10816461.36	4484941.74	6331519.62	5556105.04	563118.23	212296.35	NaN	NaN	4154407.05	2871070.74	1283336.31	932838.81	250846.62	99650.88	2746642.75	1.0
2015-03-01	10338883.11	3736251.58	6602631.54	5854019.26	510093.08	238519.20	NaN	NaN	3707313.55	2428197.77	1279115.78	967438.93	233530.80	78146.05	1698244.37	-1.5
2015-04-01	9697102.67	3206822.28	6490280.38	5729251.93	510729.20	250299.25	NaN	NaN	2698844.87	1677477.42	1021367.46	791409.26	172252.04	57706.16	1494759.91	-3.5
2015-05-01	11835256.04	3930141.80	7905114.24	6799550.62	722764.73	382798.89	NaN	NaN	3057679.52	1920200.82	1137478.69	861397.67	202906.10	73174.92	1734389.58	1.0
2015-06-01	10152993.31	3555016.50	6597976.82	5869013.13	473784.81	255178.88	NaN	NaN	3125792.68	1896942.44	1228850.24	915231.07	240392.86	73226.31	1465838.92	1.9
2015-07-01	9985763.09	3669471.09	6316292.00	5617554.78	457857.36	240879.86	NaN	NaN	2960079.50	1707286.25	1252793.25	846203.51	337940.15	68649.60	1462700.24	-1.6
2015-08-01	34394472.61	12266252.92	22128219.68	19426027.70	1810189.64	892002.34	NaN	NaN	9785627.97	5074266.13	4711361.83	3322924.14	1203457.50	184980.18	4366738.27	1.4
2015-09-01	15344748.24	4207343.16	11137405.09	9986006.62	750892.41	400506.06	NaN	NaN	3725125.93	1721634.02	2003491.93	1520265.98	417310.49	65915.46	1443853.36	1.6
2015-10-01	13657971.44	2353116.89	11304854.54	10691543.08	433266.04	180045.42	NaN	NaN	2523579.17	877919.96	1645659.20	1360097.73	237009.47	48552.00	1108486.78	9.9
2015-11-01	17168860.35	4768873.62	12399986.73	11426841.51	684517.27	288627.95	NaN	NaN	3555282.14	1694342.51	1860939.63	1491541.75	301004.18	68393.70	1302145.29	1.3
2015-12-01	10876184.49	3679288.50	7196895.99	6483080.76	536125.80	177689.43	NaN	NaN	3878188.62	2526554.72	1351633.90	938449.72	327460.83	85723.36	2217095.26	1.0

半年数据图表（1-6月 & 1-12月）数据

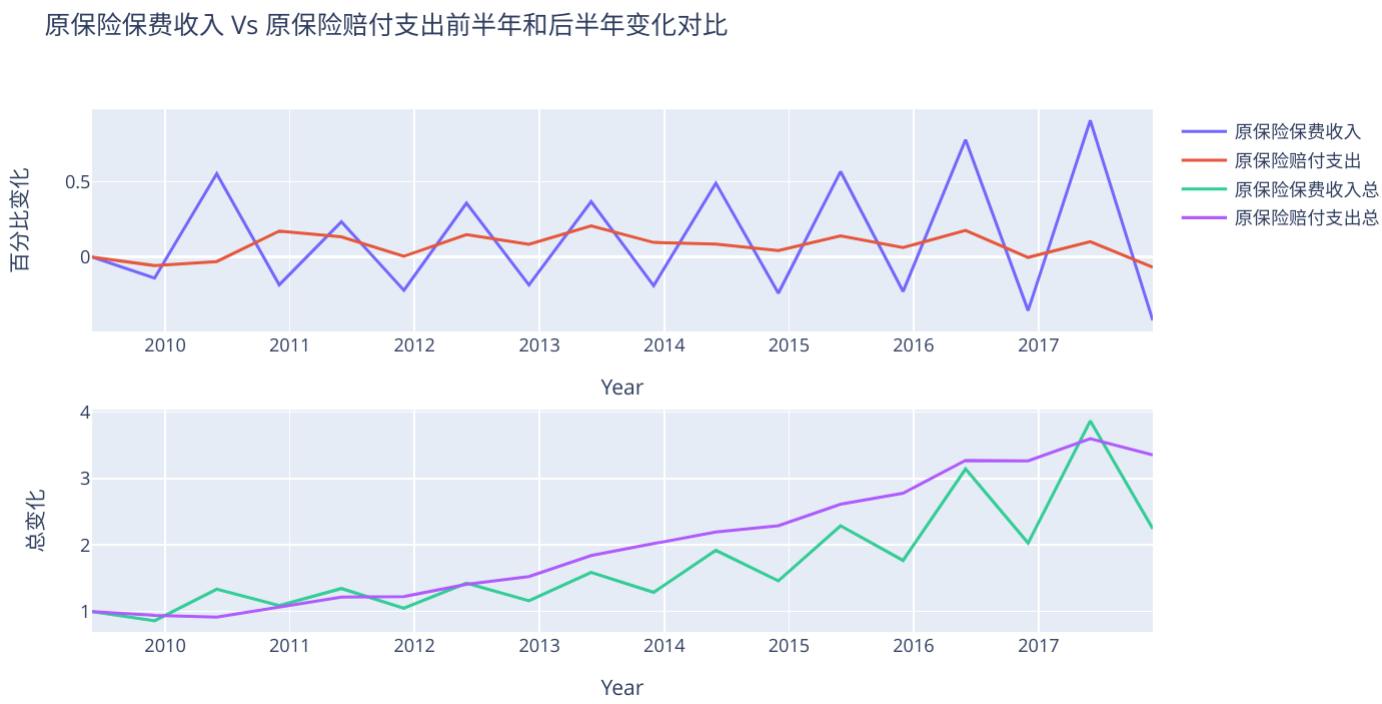
```
data_half = data.query("Month in ['1-6','1-12']").reset_index(drop = True)
```

每年上半年下半年的保费差异

```
data_half_diff = data_half.groupby('Year').apply(cal_each_month)
data_half_diff['date'] = data_half['Year'].astype(str) + data_half.Month.astype(str).apply(lambda x:'-' + x[2:])
```

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

```
tmp = data_half_diff.set_index('date')[['原保险保费收入']].pct_change().fillna(0).reset_index()
fig = make_subplots(
    rows=2, cols=1,
    column_widths=[0.8],
    row_heights=[1]*2,
    specs=[[{"type": "scatter"}],
           [{"type": "scatter"}]])

fig.add_trace(go.Scatter(x = data_half_diff['date'].tolist(),
                         y =  data_half_diff['原保险保费收入'].pct_change().fillna(0).tolist(),
                         mode = 'lines', name = '原保险保费收入'),
                         row=1, col = 1)

fig.add_trace(go.Scatter(x = data_half_diff['date'].tolist(),
                         y =  data_half_diff['原保险赔付支出'].pct_change().fillna(0).tolist(),
                         mode = 'lines', name = '原保险赔付支出'),
                         row=1, col = 1)
#fig.update_layout(title = '原保险保费收入 Vs 原保险赔付支出前半年和后半年变化对比',xaxis_title='Year',yaxis_title= '百分比变化')
fig.add_trace(go.Scatter(x = data_half_diff['date'].tolist(),
                         y = (tmp['原保险保费收入']+1).cumprod().tolist(),
                         mode = 'lines', name = '原保险保费收入总变化'),
                         row=2, col = 1)

fig.add_trace(go.Scatter(x = data_half_diff['date'].tolist(),
                         y = (data_half_diff['原保险赔付支出'].pct_change().fillna(0)+1).cumprod().tolist(),
                         mode = 'lines', name = '原保险赔付支出总变化'),
                         row=2, col = 1)

fig.update_layout(title = '原保险保费收入 Vs 原保险赔付支出前半年和后半年变化对比',
                  xaxis = dict(title='Year'),yaxis= dict(title='百分比变化'),
                  xaxis2 = dict(title='Year'),yaxis2= dict(title='总变化'))

#fig.show()
```



可以看出在前半年的保费收入和赔偿都要多于后半年，存在较强的周期性

```
#data_half_diff
```