

## 1、时间的修改

```
: df = pd.read_excel('数据.xlsx')
df.drop_duplicates(subset=['文本'], inplace=True)

#处理时间函数
def time_process(x):
    x1 = int(x)
    if x1 < 2009:
        return "2009以前"
    elif 2009 <= x1 <= 2015:
        return "2009-2015"
    elif 2016 <= x1 <= 2019:
        return "2016-2019"
    else:
        return "2019以后"

df['年份'] = df['年份'].apply(time_process)
time_df = df['年份'].value_counts()
time_df = time_df.sort_index()
```

把这里的相关时间进行修改即可

```
# Perform PCA and reduce the data to 2 dimensions
pca = PCA(n_components=2)
X_umap = pca.fit_transform(weight)

# umap_instance = umap.UMAP(n_components=2)
# X_umap = umap_instance.fit_transform(weight)
```

这里，对应的每一块记得都要改为下面这样

```
# Perform PCA and reduce the data to 2 dimensions
# pca = PCA(n_components=2)
# X_umap = pca.fit_transform(weight)

umap_instance = umap.UMAP(n_components=2)
X_umap = umap_instance.fit_transform(weight)
```

我这边因为版本问题就不重新配置环境了，之前已经在你的电脑上面配置好相关的环境了

## 2、图形的大小

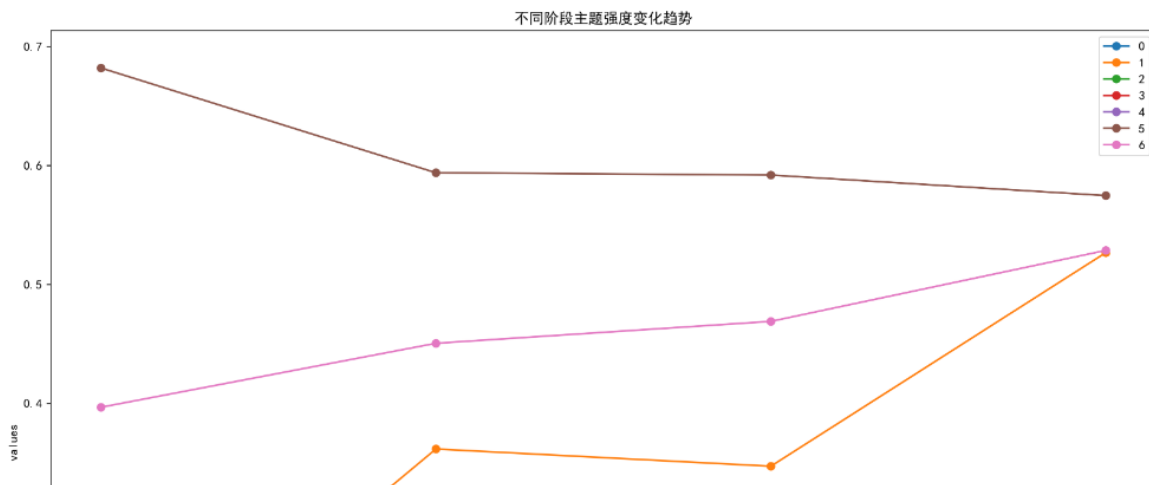
```

for t in range(len(time_date)):
    for x,y in zip(x_datal[t],y_datal[t]):
        dic[x].append(y)
fig = plt.figure(figsize=(16,12),dpi=500)
plt.title('不同阶段主题强度变化趋势')
plt.xlabel('月份')
plt.ylabel('values')
time_date = ['2008以前', '2008-2015', '2016-2019', '2019以后']

for key, values in dic.items():
    plt.plot(time_date, values, marker='o', label=key)

plt.legend(loc="upper right")
plt.savefig('不同阶段主题强度变化趋势.png')
plt.show()

```



这里是调整他的长和宽的比例

我另外保存了他们的数据

名称	修改日期	类型	大小
lda.html	2024/3/26 23:10	Chrome HTML D...	106 KB
main.ipynb	2024/3/26 23:13	IPYNB 文件	10,685 KB
policy.py	2023/12/11 22:45	JetBrains PyChar...	1 KB
stopwords_cn.txt	2023/12/11 22:45	文本文档	15 KB
word_node.txt	2024/3/26 22:39	文本文档	56 KB
不同阶段 困惑度和一致性.png	2024/3/26 23:05	PNG 文件	97 KB
不同阶段主题强度变化趋势.png	2024/3/26 23:16	PNG 文件	742 KB
不同阶段主题强度数据.csv	2024/3/26 23:16	Microsoft Excel ...	1 KB
词云图.png	2024/3/26 22:39	PNG 文件	229 KB
高频词Top80.csv	2024/3/26 22:39	Microsoft Excel ...	2 KB
共现语义.png	2024/3/26 22:39	PNG 文件	6,946 KB
环境面主题演化路径.html	2024/2/26 11:33	Chrome HTML D...	24 KB
困惑度和一致性.png	2024/3/26 22:42	PNG 文件	55 KB
数据.xlsx	2024/2/26 9:06	Microsoft Excel ...	202 KB
中国新能源汽车发展主题演化路径.html	2024/2/26 9:48	Chrome HTML D...	30 KB
主题词分布表.csv	2024/3/26 23:10	Microsoft Excel ...	3 KB

你需要用R语言作图可以根据这个数据来重新作图

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		0	1	2	3	4	5	6											
2	0	0.2584	0.127	0.0531	0.1221	0.0992	0.6821	0.3968											
3	1	0.2004	0.3617	0.1771	0.2152	0.2437	0.594	0.4506											
4	2	0.149	0.3472	0.2854	0.2152	0.1639	0.5921	0.469											
5	3	0.213	0.5268	0.3068	0.2433	0.2252	0.5748	0.5287											
6																			
7																			
8																			
9																			
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			
22																			
23																			
24																			
25																			
26																			
27																			

0 1 2 3 对应的就是['2008以前','2008-2015','2016-2019','2019以后']

3、桑基图的话，加入阈值参数的话，主要是在这里加入对应的阈值

```

links = []
for x in range(num_topics):
    for j in range(num_topics):
        # 对vec1进行线性插值
        vec1_resized = np.interp(np.linspace(0, 1, len(list_data[1][j])), np.linspace(0, 1, len(list_data[0][x])), list_data[0][x])
        hellinger_distance = hellinger(vec1_resized, list_data[1][j])
        hellinger_distance1 = round(hellinger_distance, 2)
        if float(hellinger_distance1) >= 0.5:
            d = {
                "source": "P1_Topic_{}".format(x),
                "target": "P2_Topic_{}".format(j),
                "value": hellinger_distance1
            }
            links.append(d)

# 依次计算不同时间点、不同主题之间的Hellinger距离
for x in range(num_topics):
    for j in range(num_topics):
        vec1_resized = np.interp(np.linspace(0, 1, len(list_data[2][j])), np.linspace(0, 1, len(list_data[1][x])), list_data[1][x])
        hellinger_distance = hellinger(vec1_resized, list_data[2][j])
        hellinger_distance1 = round(hellinger_distance, 2)
        if float(hellinger_distance1) >= 0.5:
            d = {
                "source": "P2_Topic_{}".format(x),
                "target": "P3_Topic_{}".format(j),
                "value": hellinger_distance1
            }
            links.append(d)

for x in range(num_topics):
    for j in range(num_topics):
        # 对vec1进行线性插值，扩展为563维
        vec1_resized = np.interp(np.linspace(0, 1, len(list_data[3][j])), np.linspace(0, 1, len(list_data[2][x])), list_data[2][x])
        hellinger_distance = hellinger(vec1_resized, list_data[3][j])
        hellinger_distance1 = round(hellinger_distance, 2)
        if float(hellinger_distance1) >= 0.5:
            d = {
                "source": "P3_Topic_{}".format(x),
                "target": "P4_Topic_{}".format(j),
                "value": hellinger_distance1
            }
            links.append(d)

```

根据你的不同的需求，要注意的是，这里一共分为三段，一段是第一阶段到第二阶段的

第二段是第二阶段到第三阶段的

第三段是第三阶段到第四阶段的，不同的段落，阈值会不一样，所以你根据你的相关需求加入不同阈值进行调整即可，不太清楚的话，你就点击相关的可视化进行查看

4、如何改名的话，主要是看这里

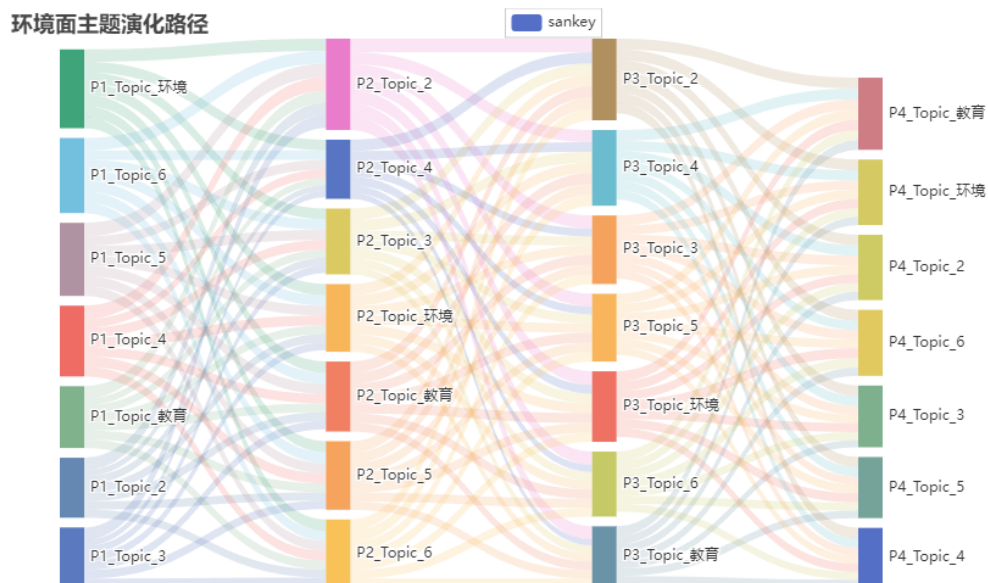
```

links.append(d)
links1 = []
for l in links:
    source = l['source']
    target = l['target']
    source1 = str(source).replace('_0', '_教育').replace('_1', '_环境')
    target1 = str(target).replace('_1', '_环境').replace('_0', '_教育')
    d = {
        "source": source1,
        "target": target1,
        "value": l['value']
    }
    links1.append(d)
nodes1 = []
for n in nodes:
    name = n['name']
    name1 = str(name).replace('_0', '_教育').replace('_1', '_环境')
    d = {
        "name": name1
    }
    nodes1.append(d)

c = (
    Sankey()
    .add(
        "sankey",
        nodes1,
        links1,
        linestyle_opts=opts.LineStyleOpts(opacity=0.2, curve=0.5, color="source"),
        label_opts=opts.LabelOpts(position="right"),
    )
    .set_global_opts(title_opts=opts.TitleOpts(title="环境面主题演化路径"))
    .render("环境面主题演化路径.html")
)
print('生成完毕')

```

如果你要修改一个名字，就要修改这三处地方，这三处地方都修改了，后面的可视化才会跟着改变



你也可以修改整个名字，例如P1\_Topic\_6 这样，看你个人