

智能制造装备博览会运营数据分析

围绕展会的招商、运营与复盘，用 Python 采集公开数据并进行分析与可视化。

数据来源

数据	来源
中国制造业增加值（年度）	世界银行 World Bank API
中文评论语料 ChnSentiCorp	公开中文 NLP 语料
Online Retail 交易明细	UCI 机器学习库

任务模块（6 个）：市场规模分析、任务与资源分配、社交媒体情感分析、供应链管理、项目管理、财务预算。

```
In [1]: # 导入库与中文字体设置
import warnings, logging, os, json, ssl, zipfile, urllib.request
warnings.filterwarnings('ignore')
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

plt.style.use('seaborn-v0_8-whitegrid')
plt.rcParams['font.sans-serif'] = ['Microsoft YaHei', 'SimHei']
plt.rcParams['axes.unicode_minus'] = False
ssl._create_default_https_context = ssl._create_unverified_context

DATA_DIR = 'data'
os.makedirs(DATA_DIR, exist_ok=True)
UA = {'User-Agent': 'Mozilla/5.0'}

def fetch(url, fname, timeout=180):
    path = os.path.join(DATA_DIR, fname)
    if not os.path.exists(path) or os.path.getsize(path) == 0:
        print('下载', fname)
        req = urllib.request.Request(url, headers=UA)
        with urllib.request.urlopen(req, timeout=timeout) as r, open(path, 'wb') as f:
            f.write(r.read())
    return path

print('环境就绪')
```

环境就绪

一、会展行业市场规模分析

制造业景气度与会展规模相关。采集世界银行「中国制造业增加值」年度数据，用多项式回归预测未来三年。

```
In [2]: # 采集世界银行数据：中国制造业增加值
def fetch_worldbank(indicator, fname):
    path = os.path.join(DATA_DIR, fname)
    if not os.path.exists(path) or os.path.getsize(path) == 0:
        url = 'https://api.worldbank.org/v2/country/CHN/indicator/' + indicator + '
        req = urllib.request.Request(url, headers=UA)
        with urllib.request.urlopen(req, timeout=60) as r:
            js = json.loads(r.read().decode('utf-8'))
            recs = [(int(d['date']), d['value']) for d in js[1] if d['value'] is not No
            pd.DataFrame(sorted(recs), columns=['年份', '指标值']).to_csv(path, index=Fa
    return pd.read_csv(path)

mkt = fetch_worldbank('NV.IND.MANF.CD', 'worldbank_manufacturing.csv')
mkt = mkt[mkt['年份'] >= 2004].reset_index(drop=True)
mkt['制造业增加值_万亿美元'] = (mkt['指标值'] / 1e12).round(3)
print('年份范围', mkt['年份'].min(), '-', mkt['年份'].max(), '共', len(mkt), '条')
mkt[['年份', '制造业增加值_万亿美元']].tail()
```

年份范围 2004 - 2024 共 21 条

```
Out[2]:
```

	年份	制造业增加值_万亿美元
16	2020	3.861
17	2021	4.909
18	2022	4.840
19	2023	4.659
20	2024	4.661

```
In [3]: # 多项式回归拟合并预测未来三年
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score

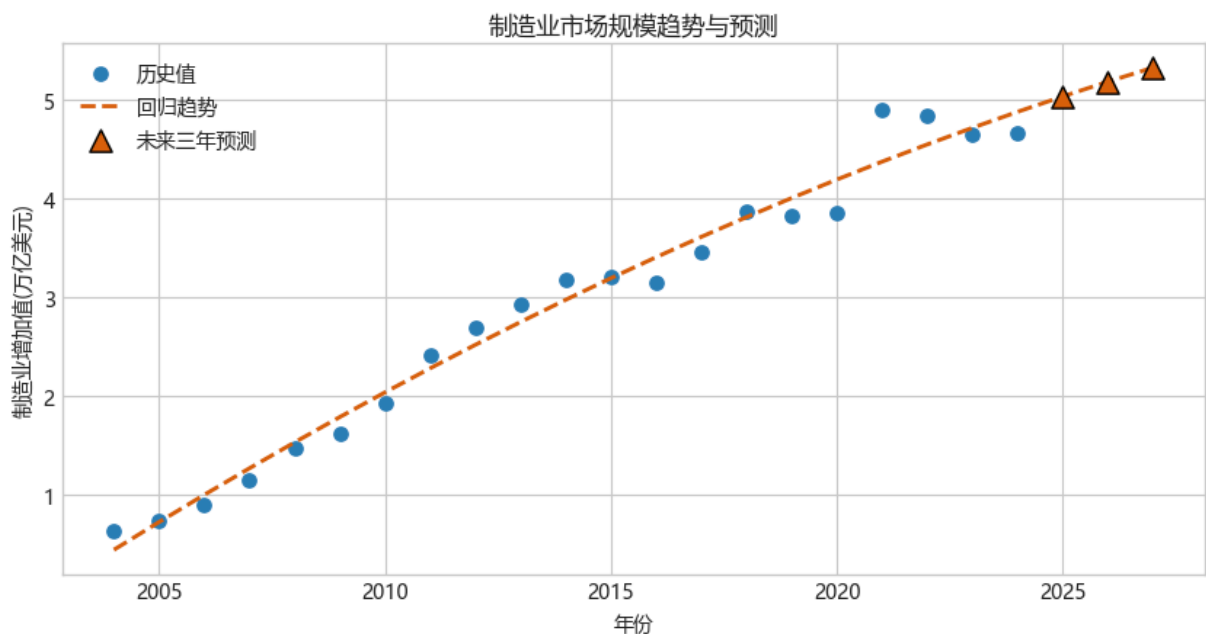
X = mkt[['年份']].values
y = mkt['制造业增加值_万亿美元'].values
model_mkt = make_pipeline(PolynomialFeatures(2), LinearRegression()).fit(X, y)
r2 = r2_score(y, model_mkt.predict(X))

last_year = int(mkt['年份'].max())
future_years = np.arange(last_year + 1, last_year + 4).reshape(-1, 1)
forecast = pd.DataFrame({'年份': future_years.ravel(),
                        '预测值_万亿美元': model_mkt.predict(future_years).round(3)})
print('拟合优度 R2 =', round(r2, 3))
print(forecast.to_string(index=False))
```

拟合优度 R2 = 0.977

年份	预测值_万亿美元
2025	5.039
2026	5.188
2027	5.332

```
In [4]: # 可视化趋势与预测
all_x = np.vstack([X, future_years])
plt.figure(figsize=(8.5, 4.6))
plt.scatter(X.ravel(), y, color='#2c7fb8', s=45, label='历史值')
plt.plot(all_x.ravel(), model_mkt.predict(all_x), '--', color='#d95f0e', lw=2, label='回归趋势')
plt.scatter(future_years.ravel(), model_mkt.predict(future_years), marker='^', s=12, color='#d95f0e', edgecolor='k', zorder=5, label='未来三年预测')
plt.xlabel('年份')
plt.ylabel('制造业增加值(万亿美元)')
plt.title('制造业市场规模趋势与预测')
plt.legend()
plt.tight_layout()
plt.show()
```



制造业增加值持续增长，回归拟合较好，预测未来三年仍将上升，可支撑扩大办展规模的判断。

二、会展运营任务与资源分配

预算有限，需在各展区间分配推广预算以最大化预期意向客户。用线性规划求解。

```
In [5]: # 展区参数
import pulp
booths = ['智能机床', '工业机器人', '3D打印', '在线检测', '工业软件']
eff = {'智能机床': 0.9, '工业机器人': 1.4, '3D打印': 1.1, '在线检测': 0.8, '工业软件': 1.0}
lo = {b: 20 for b in booths}
hi = {'智能机床': 120, '工业机器人': 150, '3D打印': 100, '在线检测': 90, '工业软件': 100}
TOTAL_BUDGET = 400
pd.DataFrame({'展区': booths, '转化效率(人/万元)': [eff[b] for b in booths],
              '预算下限': [lo[b] for b in booths], '预算上限': [hi[b] for b in booths]})
```

Out[5]:

	展区	转化效率(人/万元)	预算下限	预算上限
0	智能机床	0.9	20	120
1	工业机器人	1.4	20	150
2	3D打印	1.1	20	100
3	在线检测	0.8	20	90
4	工业软件	1.2	20	110

```
In [6]: # 线性规划求解最优分配
prob = pulp.LpProblem('alloc', pulp.LpMaximize)
x = {b: pulp.LpVariable('x' + str(i), lowBound=lo[b], upBound=hi[b]) for i, b in enumerate(booths)}
prob += pulp.lpSum(eff[b] * x[b] for b in booths)
prob += pulp.lpSum(x[b] for b in booths) <= TOTAL_BUDGET
prob.solve(pulp.PULP_CBC_CMD(msg=0))

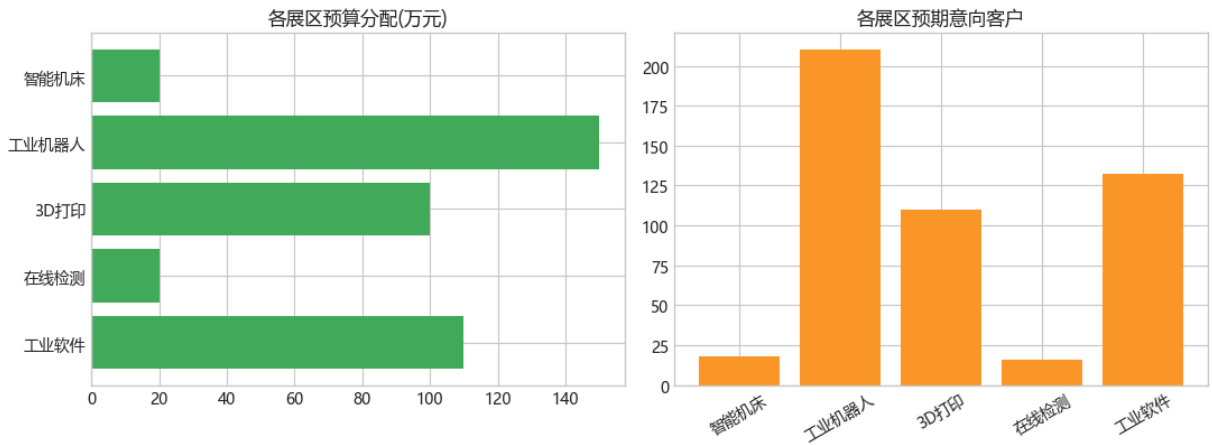
alloc = {b: round(x[b].value(), 1) for b in booths}
df_alloc = pd.DataFrame({'展区': booths, '分配预算_万元': [alloc[b] for b in booths],
                        '预期意向客户': [round(alloc[b] * eff[b], 1) for b in booths]})
print('求解状态', pulp.LpStatus[prob.status])
print('预期意向客户总数', round(pulp.value(prob.objective), 1))
df_alloc
```

求解状态 Optimal
 预期意向客户总数 486.0

Out[6]:

	展区	分配预算_万元	预期意向客户
0	智能机床	20.0	18.0
1	工业机器人	150.0	210.0
2	3D打印	100.0	110.0
3	在线检测	20.0	16.0
4	工业软件	110.0	132.0

```
In [7]: # 可视化分配结果
fig, ax = plt.subplots(1, 2, figsize=(11, 4.2))
ax[0].barh(df_alloc['展区'], df_alloc['分配预算_万元'], color='#41ab5d')
ax[0].invert_yaxis()
ax[0].set_title('各展区预算分配(万元)')
ax[1].bar(df_alloc['展区'], df_alloc['预期意向客户'], color='#fe9929')
ax[1].set_title('各展区预期意向客户')
ax[1].tick_params(axis='x', rotation=30)
plt.tight_layout()
plt.show()
```



预算优先分给转化效率高的展区，在满足各展区下限的同时让总意向客户最大。

三、会展社交媒体情感分析

采集 ChnSentiCorp 中文评论语料，用 jieba 分词加 TF-IDF，训练逻辑回归做情感分类。

```
In [8]: # 采集并清洗评论语料
import jieba
jieba.setLogLevel(logging.INFO)

senti_url = 'https://hf-mirror.com/datasets/dirtycomputer/ChnSentiCorp_htl_all/reso
senti_path = fetch(senti_url, 'ChnSentiCorp_htl_all.csv')
df_senti = pd.read_csv(senti_path).dropna(subset=['review'])
df_senti['label'] = df_senti['label'].astype(int)
print('评论条数', len(df_senti), '正面占比', round(df_senti['label'].mean() * 100, 1)

stop = set(['的', '了', '是', '我', '在', '也', '都', '和', '就', '不', '很', '有', '
def cut(t):
    return ' '.join(w for w in jieba.lcut(str(t)) if w.strip() and w not in stop an
df_senti['tokens'] = df_senti['review'].map(cut)
df_senti[['label', 'review']].head(3)
```

评论条数 7765 正面占比 68.5 %

```
Out[8]:
```

	label	review
0	1	距离川沙公路较近,但是公交指示不对,如果是"蔡陆线"的话,会非常麻烦.建议用别的路线. 房间较...
1	1	商务大床房, 房间很大, 床有2M宽, 整体感觉经济实惠不错!
2	1	早餐太差, 无论去多少人, 那边也不加食品的。酒店应该重视一下这个问题了。房间本身很好。

```
In [9]: # TF-IDF 加逻辑回归
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
```

```

vec = TfidfVectorizer(max_features=4000)
Xv = vec.fit_transform(df_senti['tokens'])
yv = df_senti['label'].values
Xtr, Xte, ytr, yte = train_test_split(Xv, yv, test_size=0.25, random_state=42, stratify=yv)
clf = LogisticRegression(max_iter=1000).fit(Xtr, ytr)
ypred = clf.predict(Xte)
print(classification_report(yte, ypred, target_names=['负面', '正面'], digits=3))

```

	precision	recall	f1-score	support
负面	0.849	0.707	0.771	611
正面	0.875	0.942	0.907	1331
accuracy			0.868	1942
macro avg	0.862	0.825	0.839	1942
weighted avg	0.867	0.868	0.865	1942

```

In [10]: # 混淆矩阵、好评率与词云
import seaborn as sns
from wordcloud import WordCloud

fig, ax = plt.subplots(1, 2, figsize=(11, 4))
cm = confusion_matrix(yte, ypred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['负面', '正面'],
            yticklabels=['负面', '正面'], ax=ax[0])
ax[0].set_xlabel('预测')
ax[0].set_ylabel('真实')
ax[0].set_title('混淆矩阵')

df_senti['评论长度'] = df_senti['review'].str.len()
df_senti['长度区间'] = pd.qcut(df_senti['评论长度'], 5, duplicates='drop')
pos_rate = df_senti.groupby('长度区间')['label'].mean()
ax[1].bar(range(len(pos_rate)), pos_rate.values, color='#756bb1')
ax[1].set_xticks(range(len(pos_rate)))
ax[1].set_xticklabels(['Q' + str(i + 1) for i in range(len(pos_rate))])
ax[1].set_title('不同评论长度的好评率')
ax[1].set_ylabel('好评率')
plt.tight_layout()
plt.show()

pos_text = ' '.join(df_senti[df_senti['label'] == 1]['tokens'].head(2000))
try:
    wc = WordCloud(font_path=r'C:\Windows\Fonts\msyh.ttc', width=700, height=320,
                  background_color='white', collocations=False).generate(pos_text)
    plt.figure(figsize=(8, 3.6))
    plt.imshow(wc)
    plt.axis('off')
    plt.title('正面评论高频词')
    plt.tight_layout()
    plt.show()
except Exception as e:
    print('词云跳过', e)

```



```

df.to_csv(csvp, index=False)
return df

retail = fetch_online_retail()
print('交易记录', len(retail), '时间', retail['InvoiceDate'].min().date(), '至', ret

top_code = retail.groupby('StockCode')['InvoiceNo'].nunique().idxmax()
sub = retail[retail['StockCode'] == top_code]
daily = sub.groupby(sub['InvoiceDate'].dt.date)['Quantity'].sum()
idx = pd.date_range(daily.index.min(), daily.index.max())
demand = daily.reindex(idx.date, fill_value=0).values
print('热销商品', top_code, '日均需求', round(demand.mean(), 1))

```

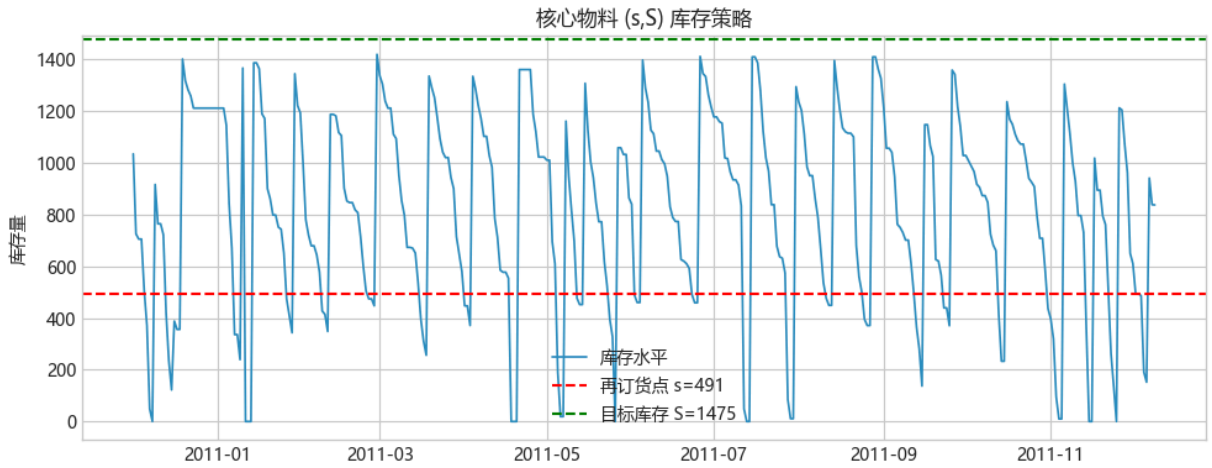
交易记录 397884 时间 2010-12-01 至 2011-12-09
热销商品 85123A 日均需求 98.3

```

In [12]: # (s,S) 库存策略
s = int(demand.mean() * 5)
S = int(demand.mean() * 15)
lead_time = 3
inv = S
track = []
orders = 0
shortage_days = 0
pipeline = {}
for t in range(len(demand)):
    if t in pipeline:
        inv += pipeline.pop(t)
    inv -= demand[t]
    if inv < 0:
        shortage_days += 1
        inv = 0
    if inv <= s and not pipeline:
        pipeline[t + lead_time] = S - inv
        orders += 1
    track.append(inv)

plt.figure(figsize=(10, 4))
plt.plot(idx, track, color='#2b8cbe', lw=1.2, label='库存水平')
plt.axhline(s, color='red', ls='--', label='再订货点 s=' + str(s))
plt.axhline(S, color='green', ls='--', label='目标库存 S=' + str(S))
plt.title('核心物料 (s,S) 库存策略')
plt.ylabel('库存量')
plt.legend()
plt.tight_layout()
plt.show()
print('补货', orders, '次, 缺货', shortage_days, '天')

```



补货 28 次, 缺货 13 天

```
In [13]: # 供应商 TOPSIS 排序
suppliers = ['搭建商甲', '搭建商乙', '物流商丙', '物流商丁', '餐饮商戊']
crit = pd.DataFrame({'报价': [85, 78, 92, 70, 88], '质量评分': [9.2, 8.5, 7.8, 8.0,
'准时率': [96, 92, 88, 90, 94], '合作年限': [6, 3, 5, 2, 4]}, i
benefit = [False, True, True, True]
weights = np.array([0.35, 0.30, 0.25, 0.10])
norm = crit / np.sqrt((crit ** 2).sum())
wmat = norm * weights
ib = np.where(benefit, wmat.max(), wmat.min())
iw = np.where(benefit, wmat.min(), wmat.max())
d_best = np.sqrt(((wmat - ib) ** 2).sum(axis=1))
d_worst = np.sqrt(((wmat - iw) ** 2).sum(axis=1))
crit['得分'] = (d_worst / (d_best + d_worst)).round(3)
crit['排名'] = crit['得分'].rank(ascending=False).astype(int)
crit.sort_values('排名')
```

Out[13]:

	报价	质量评分	准时率	合作年限	得分	排名
搭建商甲	85	9.2	96	6	0.640	1
物流商丁	70	8.0	90	2	0.471	2
搭建商乙	78	8.5	92	3	0.455	3
餐饮商戊	88	8.8	94	4	0.412	4
物流商丙	92	7.8	88	5	0.391	5

(s,S) 策略给出了清晰的补货节奏, 仿真期内多次触发补货并出现少量缺货, 可通过提高再订货点进一步降低缺货风险; TOPSIS 综合多项指标给出供应商优选顺序。

五、会展项目管理分析

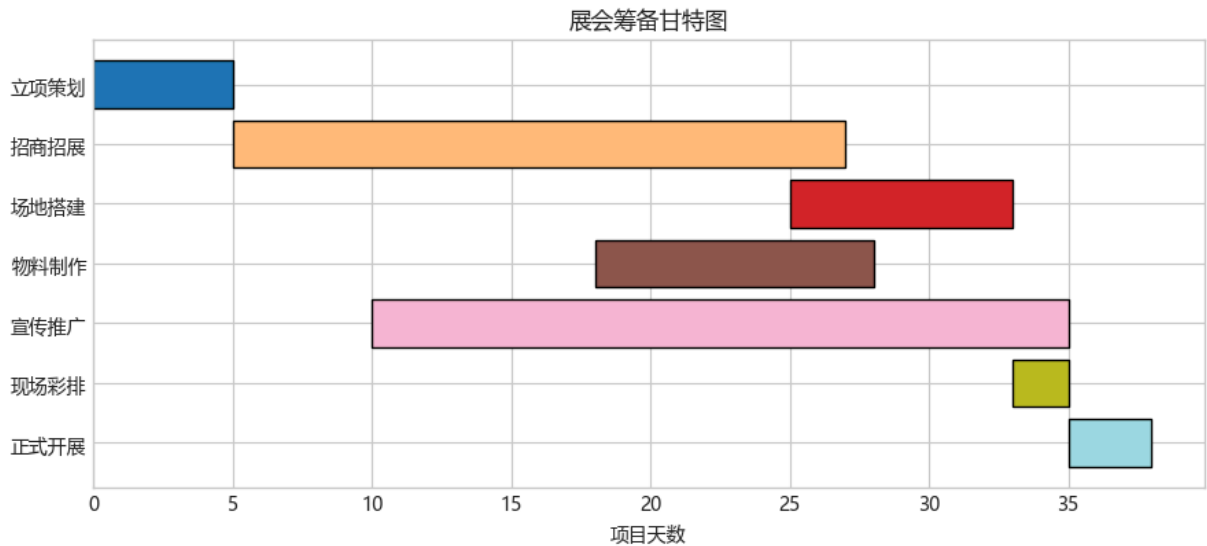
用甘特图展示筹备排期, 用 networkx 求关键路径, 并画风险矩阵。

```
In [14]: # 甘特图
import networkx as nx
tasks = pd.DataFrame({
```

```

'任务': ['立项策划', '招商招展', '场地搭建', '物料制作', '宣传推广', '现场彩排', '正
'开始日': [0, 5, 25, 18, 10, 33, 35],
'工期': [5, 22, 8, 10, 25, 2, 3],
'依赖': [None, '立项策划', '招商招展', '招商招展', '立项策划', '场地搭建', '现场彩排]
colors = plt.cm.tab20(np.linspace(0, 1, len(tasks)))
plt.figure(figsize=(9, 4.2))
for i, r in tasks.iterrows():
    plt.barh(r['任务'], r['工期'], left=r['开始日'], color=colors[i], edgecolor='k')
plt.gca().invert_yaxis()
plt.xlabel('项目天数')
plt.title('展会筹备甘特图')
plt.tight_layout()
plt.show()

```



```

In [15]: # 关键路径
G = nx.DiGraph()
dur = dict(zip(tasks['任务'], tasks['工期']))
for _, r in tasks.iterrows():
    G.add_node(r['任务'])
    if r['依赖'] is not None:
        G.add_edge(r['依赖'], r['任务'], w=dur[r['任务']])
cp = nx.dag_longest_path(G, weight='w')
print('关键路径', ' -> '.join(cp))
print('关键路径总工期', sum(dur[t] for t in cp), '天')

```

关键路径 立项策划 -> 招商招展 -> 场地搭建 -> 现场彩排 -> 正式开展
 关键路径总工期 40 天

```

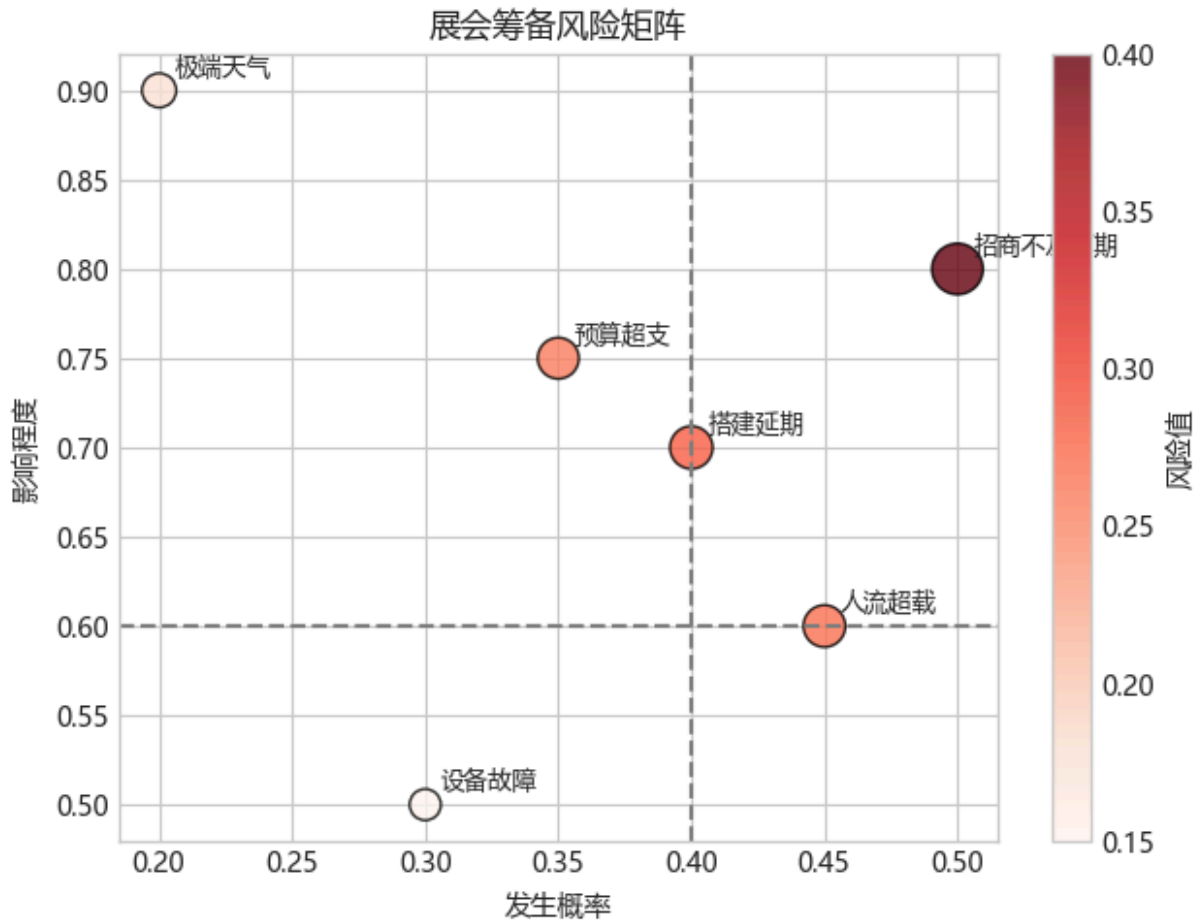
In [16]: # 风险矩阵
risks = pd.DataFrame({'风险项': ['招商不及预期', '搭建延期', '极端天气', '设备故障', '
'发生概率': [0.5, 0.4, 0.2, 0.3, 0.45, 0.35],
'影响程度': [0.8, 0.7, 0.9, 0.5, 0.6, 0.75]})
risks['风险值'] = (risks['发生概率'] * risks['影响程度']).round(2)
plt.figure(figsize=(6.5, 5))
sc = plt.scatter(risks['发生概率'], risks['影响程度'], s=risks['风险值'] * 900,
c=risks['风险值'], cmap='Reds', edgecolor='k', alpha=0.8)
for _, r in risks.iterrows():
    plt.annotate(r['风险项'], (r['发生概率'], r['影响程度']), textcoords='offset poin
plt.axvline(0.4, color='gray', ls='--')

```

```

plt.axhline(0.6, color='gray', ls='--')
plt.xlabel('发生概率')
plt.ylabel('影响程度')
plt.title('展会筹备风险矩阵')
plt.colorbar(sc, label='风险值')
plt.tight_layout()
plt.show()
risks.sort_values('风险值', ascending=False)

```



Out[16]:

	风险项	发生概率	影响程度	风险值
0	招商不及预期	0.50	0.80	0.40
1	搭建延期	0.40	0.70	0.28
4	人流超载	0.45	0.60	0.27
5	预算超支	0.35	0.75	0.26
2	极端天气	0.20	0.90	0.18
3	设备故障	0.30	0.50	0.15

关键路径是进度管控重点，风险矩阵右上角的高风险项需要优先准备预案。

六、会展财务与预算管理

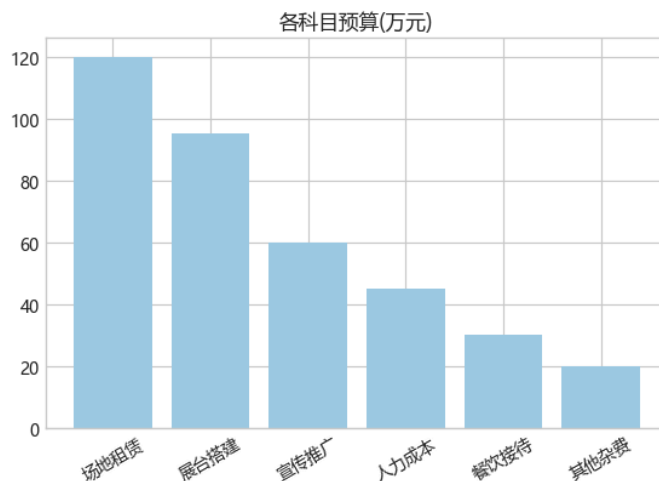
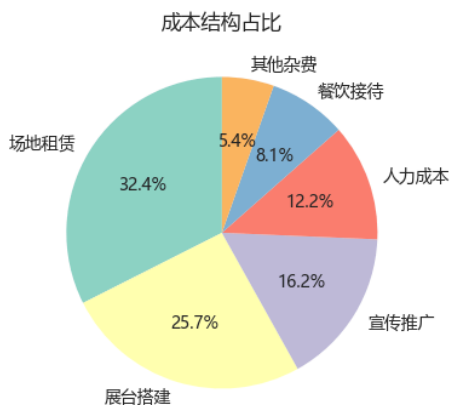
做预算编制和盈亏平衡分析，单展位收入参考交易数据的客单价来估算。

```
In [17]: # 预算编制与客单价估算
budget = pd.DataFrame({'科目': ['场地租赁', '展台搭建', '宣传推广', '人力成本', '餐饮接待', '其他杂费'],
                      '预算金额_万元': [120, 95, 60, 45, 30, 20]})

invoice_value = retail.groupby('InvoiceNo')['金额'].sum()
aov = invoice_value.median()
price_per_booth = round(aov / 1000 * 2.5, 2)
print('订单金额中位数', round(aov, 1), '估算单展位收入', price_per_booth, '万元')

fig, ax = plt.subplots(1, 2, figsize=(11, 4.2))
ax[0].pie(budget['预算金额_万元'], labels=budget['科目'], autopct='%1.1f%%', startangle=90)
ax[0].set_title('成本结构占比')
ax[1].bar(budget['科目'], budget['预算金额_万元'], color='#9ecae1')
ax[1].set_title('各科目预算(万元)')
ax[1].tick_params(axis='x', rotation=30)
plt.tight_layout()
plt.show()
budget
```

订单金额中位数 303.0 估算单展位收入 0.76 万元



```
Out[17]:
```

科目	预算金额_万元
0 场地租赁	120
1 展台搭建	95
2 宣传推广	60
3 人力成本	45
4 餐饮接待	30
5 其他杂费	20

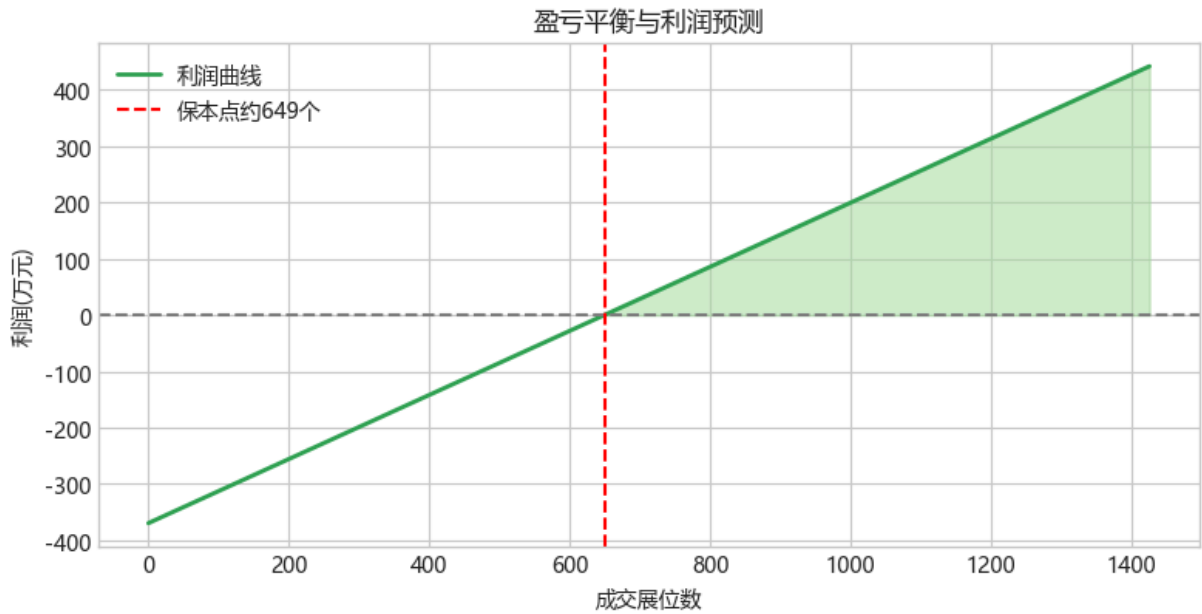
```
In [18]: # 盈亏平衡分析
fixed_cost = budget['预算金额_万元'].sum()
var_cost = round(price_per_booth * 0.25, 2)
break_even = fixed_cost / (price_per_booth - var_cost)
print('固定成本', fixed_cost, '万元, 盈亏平衡展位数约', round(break_even))
```

```

rng_x = np.arange(0, int(break_even * 2.2), 5)
profit = (price_per_booth - var_cost) * rng_x - fixed_cost
plt.figure(figsize=(8, 4.2))
plt.plot(rng_x, profit, color='#31a354', lw=2, label='利润曲线')
plt.axhline(0, color='gray', ls='--')
plt.axvline(break_even, color='red', ls='--', label='保本点约' + str(round(break_eve
plt.fill_between(rng_x, profit, 0, where=(profit >= 0), color='#a1d99b', alpha=0.5)
plt.xlabel('成交展位数')
plt.ylabel('利润(万元)')
plt.title('盈亏平衡与利润预测')
plt.legend()
plt.tight_layout()
plt.show()

```

固定成本 370 万元，盈亏平衡展位数约 649



结合客单价估算后，得到保本展位数与利润区间，招展目标应定在保本点之上并留出余量。

七、总结与分工

市场规模看涨，资源分配已优化，情感模型识别效果良好，供应链、项目管理与财务测算均给出可操作结论。

成员分工

成员	负责模块
组员A	模块一、二、三
组员B	模块四、五、六

(成员名称为占位，提交前替换为真实姓名与学号。)