本项目旨在通过Python编程语言对普利制药公司的财务数据进行深入分析，以识别潜在的财务造假行为。参考了叶钦华等（2022）发表在《会计研究》的财务舞弊识别框架构建——基于会计信息系统论及大数据视角。通过计算各维度下的具体指标，全面评估普利制药公司的财务状况和潜在风险。

环境检查与设置

In [3]:
```
! python --version
! pip list | findstr "numpy pandas"
```

```
Python 3.13.2
numpy                     2.2.3
pandas                    2.3.0
```

In [4]:
```
import matplotlib_inline
import pandas as pd
from IPython.core.interactiveshell import InteractiveShell
```

In [5]:
```
matplotlib_inline.backend_inline.set_matplotlib_formats("svg")
InteractiveShell.ast_node_interactivity = "all"

# 显示所有的列
pd.set_option("display.max_columns", None)
# 显示所有的行
pd.set_option("display.max_rows", None)
```

导入数据

In [71]:
```
df_t4=pd.read_excel("rawdata\FI_T4.xlsx",skiprows=[2],header=1)
df_t5=pd.read_excel("rawdata\FI_T5.xlsx",skiprows=[2],header=1)
df_t8=pd.read_excel("rawdata\FI_T8.xlsx",skiprows=[2],header=1)
df_comins=pd.read_excel("rawdata\FS_Comins.xlsx",skiprows=[2],header=1)
df_combas=pd.read_excel("rawdata\FS_Combas.xlsx",skiprows=[2],header=1)
df_comscfd=pd.read_excel("rawdata\FS_comscfd.xlsx",skiprows=[2],header=1)
df_pt=pd.read_excel("rawdata\PLED_TRDDETL.xlsx",skiprows=[2],header=1)
df_tfs=pd.read_excel("rawdata\SC_TopFiveSaleInfo.xlsx",skiprows=[2],header=1)
df_cg=pd.read_excel("rawdata\CG_Ybasic.xlsx",skiprows=[2],header=1)
```

```
<>:1: SyntaxWarning: invalid escape sequence '\F'
<>:2: SyntaxWarning: invalid escape sequence '\F'
<>:3: SyntaxWarning: invalid escape sequence '\F'
<>:4: SyntaxWarning: invalid escape sequence '\F'
<>:5: SyntaxWarning: invalid escape sequence '\F'
<>:6: SyntaxWarning: invalid escape sequence '\F'
<>:7: SyntaxWarning: invalid escape sequence '\P'
<>:8: SyntaxWarning: invalid escape sequence '\S'
<>:9: SyntaxWarning: invalid escape sequence '\C'
<>:1: SyntaxWarning: invalid escape sequence '\F'
<>:2: SyntaxWarning: invalid escape sequence '\F'
<>:3: SyntaxWarning: invalid escape sequence '\F'
<>:4: SyntaxWarning: invalid escape sequence '\F'
<>:5: SyntaxWarning: invalid escape sequence '\F'
<>:6: SyntaxWarning: invalid escape sequence '\F'
<>:7: SyntaxWarning: invalid escape sequence '\P'
<>:8: SyntaxWarning: invalid escape sequence '\S'
<>:9: SyntaxWarning: invalid escape sequence '\C'
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\4269263200.py:1: SyntaxWarnin
g: invalid escape sequence '\F'
  df_t4=pd.read_excel("rawdata\FI_T4.xlsx",skiprows=[2],header=1)
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\4269263200.py:2: SyntaxWarnin
g: invalid escape sequence '\F'
  df_t5=pd.read_excel("rawdata\FI_T5.xlsx",skiprows=[2],header=1)
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\4269263200.py:3: SyntaxWarnin
g: invalid escape sequence '\F'
  df_t8=pd.read_excel("rawdata\FI_T8.xlsx",skiprows=[2],header=1)
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\4269263200.py:4: SyntaxWarnin
g: invalid escape sequence '\F'
  df_comins=pd.read_excel("rawdata\FS_Comins.xlsx",skiprows=[2],header=1)
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\4269263200.py:5: SyntaxWarnin
g: invalid escape sequence '\F'
  df_combas=pd.read_excel("rawdata\FS_Combas.xlsx",skiprows=[2],header=1)
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\4269263200.py:6: SyntaxWarnin
g: invalid escape sequence '\F'
  df_comscfd=pd.read_excel("rawdata\FS_comscfd.xlsx",skiprows=[2],header=1)
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\4269263200.py:7: SyntaxWarnin
g: invalid escape sequence '\P'
  df_pt=pd.read_excel("rawdata\PLED_TRDDETL.xlsx",skiprows=[2],header=1)
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\4269263200.py:8: SyntaxWarnin
g: invalid escape sequence '\S'
  df_tfs=pd.read_excel("rawdata\SC_TopFiveSaleInfo.xlsx",skiprows=[2],header=1)
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\4269263200.py:9: SyntaxWarnin
g: invalid escape sequence '\C'
  df_cg=pd.read_excel("rawdata\CG_Ybasic.xlsx",skiprows=[2],header=1)
c:\Users\Lenovo\AppData\Local\Programs\Python\Python313\Lib\site-packages\openpyx
l\styles\stylesheet.py:237: UserWarning: Workbook contains no default style, appl
y openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
c:\Users\Lenovo\AppData\Local\Programs\Python\Python313\Lib\site-packages\openpyx
l\styles\stylesheet.py:237: UserWarning: Workbook contains no default style, appl
y openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
c:\Users\Lenovo\AppData\Local\Programs\Python\Python313\Lib\site-packages\openpyx
l\styles\stylesheet.py:237: UserWarning: Workbook contains no default style, appl
y openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
c:\Users\Lenovo\AppData\Local\Programs\Python\Python313\Lib\site-packages\openpyx
l\styles\stylesheet.py:237: UserWarning: Workbook contains no default style, appl
y openpyxl's default
```

```
  warn("Workbook contains no default style, apply openpyxl's default")
c:\Users\Lenovo\AppData\Local\Programs\Python\Python313\Lib\site-packages\openpyx
l\styles\stylesheet.py:237: UserWarning: Workbook contains no default style, appl
y openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
c:\Users\Lenovo\AppData\Local\Programs\Python\Python313\Lib\site-packages\openpyx
l\styles\stylesheet.py:237: UserWarning: Workbook contains no default style, appl
y openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
c:\Users\Lenovo\AppData\Local\Programs\Python\Python313\Lib\site-packages\openpyx
l\styles\stylesheet.py:237: UserWarning: Workbook contains no default style, appl
y openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
c:\Users\Lenovo\AppData\Local\Programs\Python\Python313\Lib\site-packages\openpyx
l\styles\stylesheet.py:237: UserWarning: Workbook contains no default style, appl
y openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
c:\Users\Lenovo\AppData\Local\Programs\Python\Python313\Lib\site-packages\openpyx
l\styles\stylesheet.py:237: UserWarning: Workbook contains no default style, appl
y openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
```

清洗数据

财务指标

In [72]:

```
df_t4.head()
df_t5.head()
df_t8.head()
```

Out[72]:

	股票代码	股票简称	统计截止日期	报表类型编码	公告来源	行业代码1	行业名称1	应收账款与收入比	应收账款周转率A	应收账款周转天数A	存货周转率A
0	4	国农科技	2018-12-31	A	0	C27	医药制造业	0.074327	13.454030	27.129418	NaN
1	4	国农科技	2019-12-31	A	0	C27	医药制造业	2.441356	0.409608	891.095877	NaN
2	153	丰原药业	2018-12-31	A	0	C27	医药制造业	0.212096	4.714846	77.415042	3.990717
3	153	丰原药业	2019-12-31	A	0	C27	医药制造业	0.208752	4.790373	76.194484	4.113691
4	153	丰原药业	2020-12-31	A	0	C27	医药制造业	0.204177	4.897716	74.728710	4.116402

Out[72]:

	股票代码	股票简称	统计截止日期	报表类型编码	公告来源	行业代码1	行业名称1	营业毛利率	营业成本率	销售费用率	财务费用率	研发费
0	4	国农科技	2018-12-31	A	0	C27	医药制造业	0.818792	0.181208	0.738372	-0.004109	0.0476
1	4	国农科技	2019-12-31	A	0	C27	医药制造业	0.730102	0.269898	0.767700	-0.015441	0.0094
2	153	丰原药业	2018-12-31	A	0	C27	医药制造业	0.389499	0.610501	0.268058	0.010370	0.0241
3	153	丰原药业	2019-12-31	A	0	C27	医药制造业	0.381550	0.618450	0.265346	0.011814	0.0240
4	153	丰原药业	2020-12-31	A	0	C27	医药制造业	0.346629	0.653371	0.234881	0.012248	0.0227

Out[72]:

	股票代码	股票简称	统计截止日期	报表类型编码	公告来源	行业代码1	行业名称1	总资产增长率A	营业利润增长率A	营业收入增长率A	经营活动产生的净流量增长率A
0	4	国农科技	2018-12-31	A	0	C27	医药制造业	0.306249	NaN	0.088536	NaN
1	4	国农科技	2019-12-31	A	0	C27	医药制造业	3.255655	NaN	-0.863119	NaN
2	153	丰原药业	2018-12-31	A	0	C27	医药制造业	0.086816	-0.769829	0.034883	8.805171
3	153	丰原药业	2019-12-31	A	0	C27	医药制造业	0.183383	-0.308112	0.042536	2.140629
4	153	丰原药业	2020-12-31	A	0	C27	医药制造业	0.021517	0.462325	0.054685	11.595046

列处理、筛选

In [73]:
```
df_t4["年份"]=df_t4["统计截止日期"].str.slice(0,4).astype("int64")
df_t5["年份"]=df_t5["统计截止日期"].str.slice(0,4).astype("int64")
df_t8["年份"]=df_t8["统计截止日期"].str.slice(0,4).astype("int64")

df_t4.columns
df_t5.columns
df_t8.columns
```

Out[73]:
```
Index(['股票代码', '股票简称', '统计截止日期', '报表类型编码', '公告来源', '行业代码1', '行业名称1',
       '应收账款与收入比', '应收账款周转率A', '应收账款周转天数A', '存货周转率A', '年份'],
      dtype='object')
```

Out[73]:
```
Index(['股票代码', '股票简称', '统计截止日期', '报表类型编码', '公告来源', '行业代码1', '行业名称1', '营业毛利率',
       '营业成本率', '销售费用率', '财务费用率', '研发费用率', '投资收益率', '年份'],
      dtype='object')
```

Out[73]:
```
Index(['股票代码', '股票简称', '统计截止日期', '报表类型编码', '公告来源', '行业代码1', '行业名称1', '总资产增长率A',
       '营业利润增长率A', '营业收入增长率A', '经营活动产生的净流量增长率A', '年份'],
      dtype='object')
```

In [74]:
```
df_t4.drop(['统计截止日期', '报表类型编码', '公告来源'],axis=1,inplace=True)
df_t5.drop(['股票简称','统计截止日期', '报表类型编码', '公告来源','行业代码1', '
df_t8.drop(['股票简称','统计截止日期', '报表类型编码', '公告来源','行业代码1', '
```

In [75]:
```
df_1=pd.merge(df_t4,df_t5,on=['股票代码','年份'],how='outer')
df_1=pd.merge(df_1,df_t8,on=['股票代码','年份'],how='outer')
```

In [76]:
```
df_1.head()
df_1.columns
```

Out[76]:

	股票代码	股票简称	行业代码1	行业名称1	应收账款与收入比	应收账款周转率A	应收账款周转天数A	存货周转率A	年份	营业毛利率	营
0	4	国农科技	C27	医药制造业	0.074327	13.454030	27.129418	NaN	2018	0.818792	0.
1	4	国农科技	C27	医药制造业	2.441356	0.409608	891.095877	NaN	2019	0.730102	0.
2	153	丰原药业	C27	医药制造业	0.212096	4.714846	77.415042	3.990717	2018	0.389499	0.
3	153	丰原药业	C27	医药制造业	0.208752	4.790373	76.194484	4.113691	2019	0.381550	0.
4	153	丰原药业	C27	医药制造业	0.204177	4.897716	74.728710	4.116402	2020	0.346629	0.

Out[76]:
```
Index(['股票代码', '股票简称', '行业代码1', '行业名称1', '应收账款与收入比', '应收账款周转率A', '应收账款周转天数A',
       '存货周转率A', '年份', '营业毛利率', '营业成本率', '销售费用率', '财务费用率', '研发费用率', '投资收益率',
       '总资产增长率A', '营业利润增长率A', '营业收入增长率A', '经营活动产生的净流量增长率A'],
      dtype='object')
```

In [77]:
```
# 打印出医药制造业的股票代码，以便后续表格筛选
pharmaceutical_codes = df_1[df_1['行业名称1'] == '医药制造业']['股票代码']

# 去重
unique_pharmaceutical_codes = pharmaceutical_codes.drop_duplicates().tolist()

# 打印医药制造业的股票代码
print("医药制造业的股票代码:")
print(unique_pharmaceutical_codes)
```

```
医药制造业的股票代码:
[4, 153, 403, 423, 513, 518, 534, 538, 566, 590, 597, 623, 650, 661, 739, 756, 76
6, 788, 790, 813, 908, 915, 919, 931, 952, 953, 989, 999, 1367, 2001, 2007, 2019,
2020, 2022, 2030, 2038, 2082, 2099, 2102, 2107, 2118, 2166, 2198, 2219, 2252, 226
2, 2275, 2287, 2294, 2317, 2332, 2349, 2365, 2370, 2390, 2393, 2399, 2411, 2412,
2422, 2424, 2433, 2435, 2437, 2550, 2562, 2566, 2581, 2603, 2644, 2653, 2675, 268
8, 2693, 2728, 2737, 2750, 2755, 2773, 2793, 2817, 2821, 2826, 2864, 2868, 2873,
2880, 2898, 2900, 2907, 2923, 2932, 2940, 300006, 300009, 300016, 300026, 300039,
300049, 300086, 300108, 300110, 300111, 300119, 300122, 300142, 300143, 300147, 3
00158, 300181, 300194, 300199, 300204, 300233, 300239, 300254, 300255, 300267, 30
0289, 300294, 300357, 300363, 300381, 300406, 300434, 300436, 300439, 300452, 300
463, 300482, 300485, 300497, 300501, 300519, 300534, 300558, 300573, 300583, 3005
84, 300601, 300630, 300636, 300639, 300642, 300683, 300685, 300702, 300705, 30072
3, 300832, 300841, 300871, 300878, 300942, 300966, 301075, 301089, 301093, 30111
1, 301130, 301201, 301207, 301211, 301246, 301258, 301277, 301281, 301301, 30133
1, 301507, 301509, 430017, 430047, 430478, 600056, 600062, 600079, 600080, 60008
5, 600129, 600161, 600195, 600196, 600200, 600201, 600211, 600216, 600222, 60022
7, 600252, 600267, 600276, 600285, 600299, 600329, 600332, 600351, 600380, 60038
5, 600420, 600422, 600436, 600479, 600488, 600513, 600518, 600521, 600530, 60053
5, 600557, 600566, 600568, 600572, 600594, 600613, 600624, 600664, 600671, 60075
0, 600771, 600781, 600789, 600812, 600851, 600867, 600993, 601089, 603087, 60313
9, 603168, 603207, 603222, 603229, 603351, 603367, 603387, 603392, 603439, 60345
6, 603520, 603538, 603566, 603567, 603590, 603658, 603669, 603676, 603707, 60371
8, 603811, 603858, 603880, 603896, 603963, 603976, 603998, 605116, 605177, 60519
9, 605507, 688062, 688068, 688075, 688076, 688091, 688098, 688117, 688136, 68816
3, 688166, 688176, 688177, 688180, 688185, 688189, 688192, 688193, 688197, 68821
7, 688221, 688235, 688247, 688253, 688266, 688276, 688278, 688289, 688298, 68830
2, 688317, 688319, 688321, 688331, 688336, 688338, 688363, 688366, 688373, 68838
2, 688393, 688399, 688426, 688428, 688443, 688468, 688488, 688505, 688506, 68851
3, 688520, 688526, 688553, 688566, 688575, 688578, 688606, 688626, 688656, 68865
8, 688670, 688687, 688739, 688767, 688799, 830946, 832566, 832735, 832982, 83323
0, 833266, 833575, 836433, 836547, 837344, 839729, 870656, 873167]
```

筛选利润表以合并

In [78]:
```
df_comins_filtered = df_comins[df_comins['证券代码'].isin(unique_pharmaceutical_
df_comins_filtered.columns
```

Out[78]:
```
Index(['证券代码', '证券简称', '统计截止日期', '报表类型', '营业总收入', '营业收
入', '减：所得税费用', '净利润'], dtype='object')
```

In [79]:
```
df_comins_filtered["年份"]=df_comins_filtered["统计截止日期"].str.slice(0,4).ast
df_comins_filtered = df_comins_filtered.rename(columns={"证券代码": "股票代码"})
df_comins_filtered.drop(['证券简称', '统计截止日期', '报表类型'],axis=1,inplace=
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\3426466560.py:1: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_comins_filtered["年份"]=df_comins_filtered["统计截止日期"].str.slice(0,4).as
type("int64")
```

筛选资产负债表以合并

In [80]:
```
df_combas_filtered = df_combas[df_combas['证券代码'].isin(unique_pharmaceutical_
df_combas_filtered.columns
```

Out[80]:
```
Index(['证券代码', '证券简称', '统计截止日期', '报表类型', '应收账款净额'], dtyp
e='object')
```

In [81]:
```
df_combas_filtered["年份"]=df_combas_filtered["统计截止日期"].str.slice(0,4).ast
df_combas_filtered = df_combas_filtered.rename(columns={"证券代码": "股票代码"})

df_combas_filtered['上年应收账款净额'] = df_combas_filtered.groupby('股票代码')[
df_combas_filtered['应收账款平均余额'] = (df_combas_filtered['应收账款净额'] + d
df_combas_filtered['应收账款增长率'] = (df_combas_filtered['应收账款净额'] - df_

df_combas_filtered = df_combas_filtered[df_combas_filtered['年份'] != 2017]

df_combas_filtered.drop(['证券简称', '统计截止日期', '报表类型'],axis=1,inplace=
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\718284051.py:1: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_combas_filtered["年份"]=df_combas_filtered["统计截止日期"].str.slice(0,4).as
type("int64")
```

筛选现金流量表以合并

In [82]:
```
df_comscfd_filtered = df_comscfd[df_comscfd['证券代码'].isin(unique_pharmaceutic
df_comscfd_filtered.columns
```

Out[82]:
```
Index(['证券代码', '证券简称', '统计截止日期', '报表类型', '经营活动产生的现金流
量净额', '投资活动产生的现金流量净额',
       '筹资活动产生的现金流量净额'],
      dtype='object')
```

In [83]:
```
df_comscfd_filtered["年份"]=df_comscfd_filtered["统计截止日期"].str.slice(0,4).a
df_comscfd_filtered = df_comscfd_filtered.rename(columns={"证券代码": "股票代码"

df_comscfd_filtered.drop(['证券简称', '统计截止日期', '报表类型'],axis=1,inplace
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\1432218641.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_comscfd_filtered["年份"]=df_comscfd_filtered["统计截止日期"].str.slice(0,4).astype("int64")
```

合并财务数据

In [84]:
```
df_fi=pd.merge(df_1,df_comins_filtered,on=['股票代码','年份'],how="outer")
df_fi=pd.merge(df_fi,df_combas_filtered,on=['股票代码','年份'],how="outer")
df_fi=pd.merge(df_fi,df_comscfd_filtered,on=['股票代码','年份'],how="outer")
df_fi.head()
df_fi.columns
```

Out[84]:

	股票代码	股票简称	行业代码1	行业名称1	应收账款与收入比	应收账款周转率A	应收账款周转天数A	存货周转率A	年份	营业毛利率	营
0	4	国农科技	C27	医药制造业	0.074327	13.454030	27.129418	NaN	2018	0.818792	0.1
1	4	国农科技	C27	医药制造业	2.441356	0.409608	891.095877	NaN	2019	0.730102	0.2
2	4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020	NaN	
3	4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2021	NaN	
4	4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2022	NaN	

Out[84]:
```
Index(['股票代码', '股票简称', '行业代码1', '行业名称1', '应收账款与收入比', '应收账款周转率A', '应收账款周转天数A',
       '存货周转率A', '年份', '营业毛利率', '营业成本率', '销售费用率', '财务费用率', '研发费用率', '投资收益率',
       '总资产增长率A', '营业利润增长率A', '营业收入增长率A', '经营活动产生的净流量增长率A', '营业总收入', '营业收入',
       '减：所得税费用', '净利润', '应收账款净额', '上年应收账款净额', '应收账款平均余额', '应收账款增长率',
       '经营活动产生的现金流量净额', '投资活动产生的现金流量净额', '筹资活动产生的现金流量净额'],
      dtype='object')
```

其他数据

In [85]:
```
# 质押
df_pt.head()
# 前五大客户
df_tfs.head()
```

```
# 员工总数
df_cg.head()
```

Out[85]:

	事件ID	进展序号	证券代码	上市公司ID	证券简称	变动日期	出质方	初始数量	数量增减	剩余质押数量
0	522017002724	2	2	101775	万科A	2019-01-08	深圳市钜盛华股份有限公司	182000000	-80768434	101231566
1	522019062808	1	2	101775	万科A	2019-01-09	深圳市钜盛华股份有限公司	0	64860000	64860000
2	522017002750	2	2	101775	万科A	2019-01-10	深圳市钜盛华股份有限公司	81600000	-81600000	0
3	522019062809	1	2	101775	万科A	2019-01-11	深圳市钜盛华股份有限公司	0	72090000	72090000
4	522017002748	2	2	101775	万科A	2019-01-14	深圳市钜	91000000	-91000000	0

	事件ID	进展序号	证券代码	上市公司ID	证券简称	变动日期	出质方	初始数量	数量增减	剩余质押数量
							盛华股份有限公司			

Out[85]:

	股票代码	统计截止日期	报表类型	排名	是否上市公司关联公司	客户销售额	客户销售额占比	币种
0	2	2019-12-31	1	6	NaN	2.020000e+09	0.6	人民币元
1	2	2020-12-31	1	6	NaN	5.750000e+09	1.4	人民币元
2	2	2021-12-31	1	1	NaN	2.660000e+09	0.6	人民币元
3	2	2021-12-31	1	6	NaN	7.770000e+09	1.7	人民币元
4	2	2022-12-31	1	1	NaN	1.400000e+09	0.3	人民币元

Out[85]:

	证券代码	统计截止日期	员工人数
0	1	2018-12-31	34626.0
1	1	2019-12-31	34253.0
2	1	2020-12-31	38097.0
3	1	2021-12-31	40651.0
4	1	2022-12-31	44207.0

筛选、列处理

In [86]:

```
df_pt_filtered = df_pt[df_pt['证券代码'].isin(unique_pharmaceutical_codes)]
df_tfs_filtered = df_tfs[df_tfs['股票代码'].isin(unique_pharmaceutical_codes)]
df_cg_filtered = df_cg[df_cg['证券代码'].isin(unique_pharmaceutical_codes)]


df_pt_filtered.head()
df_tfs_filtered.head()
df_cg_filtered.head()
```

Out[86]:

	事件ID	进展序号	证券代码	上市公司ID	证券简称	变动日期	出质方	初始数量	数量增减	剩余质押数量
39	522018062813	2	4	102294	国农科技	2019-12-24	深圳中农大科技投资有限公司	15216069	0	15216069
40	522018010796	3	4	102294	国农科技	2019-12-25	中科汇通(深圳)股权投资基金有限公司	5397600	-5397600	0
41	522018062813	3	4	102294	国农科技	2020-06-17	深圳中农大科技投资有限公司	15216069	-5000000	10216069
42	522018062813	4	4	102294	国农科技	2020-06-23	深圳中农大科技投资有限	10216069	0	10216069

	事件ID	进展序号	证券代码	上市公司ID	证券简称	变动日期	出质方	初始数量	数量增减	剩余质押数量
							公司			
43	522020086057	1	4	102294	国农科技	2020-06-24	彭瀛	0	1970000	1970000

Out[86]:

	股票代码	统计截止日期	报表类型	排名	是否上市公司关联公司	客户销售额	客户销售额占比	币种
6	4	2018-12-31	1	1	NaN	20401291.04	5.58	人民币元
7	4	2018-12-31	1	2	3	6081418.35	1.66	人民币元
8	4	2018-12-31	1	3	NaN	5462871.85	1.49	人民币元
9	4	2018-12-31	1	4	NaN	4825717.80	1.32	人民币元
10	4	2018-12-31	1	5	NaN	4634457.56	1.27	人民币元

Out[86]:

	证券代码	统计截止日期	员工人数
10	4	2018-12-31	210.0
11	4	2019-12-31	251.0
12	4	2020-12-31	264.0
13	4	2021-12-31	309.0
14	4	2022-12-31	296.0

In [87]:
```
df_tfs_filtered["年份"]=df_tfs_filtered["统计截止日期"].str.slice(0,4).astype("i
df_cg_filtered["年份"]=df_cg_filtered["统计截止日期"].str.slice(0,4).astype("int
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\4071305145.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_tfs_filtered["年份"]=df_tfs_filtered["统计截止日期"].str.slice(0,4).astype("int64")
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\4071305145.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_cg_filtered["年份"]=df_cg_filtered["统计截止日期"].str.slice(0,4).astype("int64")
```

In [88]:
```
df_tfs_filtered.head()
df_tfs_filtered.columns

df_tfs_filtered.drop(['统计截止日期', '报表类型'],axis=1,inplace=True)
```

Out[88]:

	股票代码	统计截止日期	报表类型	排名	是否上市公司关联公司	客户销售额	客户销售额占比	币种	年份
6	4	2018-12-31	1	1	NaN	20401291.04	5.58	人民币元	2018
7	4	2018-12-31	1	2	3	6081418.35	1.66	人民币元	2018
8	4	2018-12-31	1	3	NaN	5462871.85	1.49	人民币元	2018
9	4	2018-12-31	1	4	NaN	4825717.80	1.32	人民币元	2018
10	4	2018-12-31	1	5	NaN	4634457.56	1.27	人民币元	2018

Out[88]:
```
Index(['股票代码', '统计截止日期', '报表类型', '排名', '是否上市公司关联公司',
'客户销售额', '客户销售额占比', '币种',
       '年份'],
      dtype='object')
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\1245656129.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_tfs_filtered.drop(['统计截止日期', '报表类型'],axis=1,inplace=True)
```

五大客户、员工人数进一步处理

五大客户

In [89]:
```
df_tfs_filtered.columns
```

Out[89]:
```
Index(['股票代码', '排名', '是否上市公司关联公司', '客户销售额', '客户销售额占比', '币种', '年份'], dtype='object')
```

In [90]:
```
df_tfs_filtered = df_tfs_filtered.rename(columns={"排名": "五大客户排名"})
df_tfs_filtered = df_tfs_filtered.rename(columns={"客户销售额": "五大客户销售额"
df_tfs_filtered = df_tfs_filtered.rename(columns={"客户销售额占比": "五大客户销售
```

In [91]:
```
grouped = df_tfs_filtered.groupby(['股票代码', '年份'])

# 计算每家公司的前五大客户的销售额和销售额占比和
df_tfs_result = grouped.apply(lambda x: x.nlargest(5, '五大客户销售额').agg({
    '五大客户销售额': 'sum',
    '五大客户销售额占比': 'sum'
})).reset_index()

df_tfs_result.columns = ['股票代码', '年份', '前五大客户销售额和', '前五大客户销售

df_tfs_result[['前五大客户销售额和', '前五大客户销售额占比和']] = df_tfs_result[[
df_tfs_result.head()
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\3325749258.py:4: FutureWarnin
g: DataFrameGroupBy.apply operated on the grouping columns. This behavior is depr
ecated, and in a future version of pandas the grouping columns will be excluded f
rom the operation. Either pass `include_groups=False` to exclude the groupings or
explicitly select the grouping columns after groupby to silence this warning.
  df_tfs_result = grouped.apply(lambda x: x.nlargest(5, '五大客户销售额').agg({
```

Out[91]:

	股票代码	年份	前五大客户销售额和	前五大客户销售额占比和
0	4	2018	7.817706e+07	21.38
1	4	2019	3.164380e+07	29.23
2	4	2020	8.502910e+07	27.28
3	4	2021	4.850428e+08	70.18
4	4	2022	1.270443e+08	62.72

员工人数

In [92]:
```
df_cg_filtered.columns
```

Out[92]:
```
Index(['证券代码', '统计截止日期', '员工人数', '年份'], dtype='object')
```

In [93]:
```
df_cg_filtered.drop(['统计截止日期'],axis=1,inplace=True)
df_cg_filtered = df_cg_filtered.rename(columns={"证券代码": "股票代码"})
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_17304\733057480.py:1: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_cg_filtered.drop(['统计截止日期'],axis=1,inplace=True)
```

合并

In [94]:
```
df=pd.merge(df_fi,df_tfs_result,on=['股票代码','年份'],how="outer")
df=pd.merge(df,df_cg_filtered,on=['股票代码','年份'],how="outer")
```

缺失值处理

In [95]:
```
df.shape
```

Out[95]:
```
(1674, 33)
```

In [96]:
```
df.isnull().sum()
```

Out[96]:
```
股票代码                    0
股票简称                   75
行业代码1                  75
行业名称1                  75
应收账款与收入比               75
应收账款周转率A               99
应收账款周转天数A              99
存货周转率A                 87
年份                      0
营业毛利率                  77
营业成本率                  77
销售费用率                  75
财务费用率                  75
研发费用率                  76
投资收益率                 406
总资产增长率A                276
营业利润增长率A               483
营业收入增长率A               322
经营活动产生的净流量增长率A          588
营业总收入                  251
营业收入                   251
减：所得税费用                272
净利润                    248
应收账款净额                 263
上年应收账款净额               355
应收账款平均余额               359
应收账款增长率                359
经营活动产生的现金流量净额          248
投资活动产生的现金流量净额          248
筹资活动产生的现金流量净额          253
前五大客户销售额和              286
前五大客户销售额占比和             286
员工人数                   308
dtype: int64
```

In [97]:
```
df.columns
```

Out[97]:
```
Index(['股票代码', '股票简称', '行业代码1', '行业名称1', '应收账款与收入比', '应收账款周转率A', '应收账款周转天数A',
       '存货周转率A', '年份', '营业毛利率', '营业成本率', '销售费用率', '财务费用率', '研发费用率', '投资收益率',
       '总资产增长率A', '营业利润增长率A', '营业收入增长率A', '经营活动产生的净流量增长率A', '营业总收入', '营业收入',
       '减：所得税费用', '净利润', '应收账款净额', '上年应收账款净额', '应收账款平均余额', '应收账款增长率',
       '经营活动产生的现金流量净额', '投资活动产生的现金流量净额', '筹资活动产生的现金流量净额', '前五大客户销售额和',
       '前五大客户销售额占比和', '员工人数'],
      dtype='object')
```

In [98]:
```
# 剔除缺失较多列
df.drop(['投资收益率','营业利润增长率A','经营活动产生的净流量增长率A'],axis=1,inp
```

In [99]:
```
df=df.dropna()
```

In [100…
```
df.isnull().sum()
df.shape
```

Out[100…
```
股票代码                0
股票简称                0
行业代码1               0
行业名称1               0
应收账款与收入比            0
应收账款周转率A           0
应收账款周转天数A          0
存货周转率A             0
年份                  0
营业毛利率               0
营业成本率               0
销售费用率               0
财务费用率               0
研发费用率               0
总资产增长率A            0
营业收入增长率A           0
营业总收入               0
营业收入                0
减：所得税费用             0
净利润                 0
应收账款净额              0
上年应收账款净额            0
应收账款平均余额            0
应收账款增长率             0
经营活动产生的现金流量净额     0
投资活动产生的现金流量净额     0
筹资活动产生的现金流量净额     0
前五大客户销售额和          0
前五大客户销售额占比和        0
员工人数                0
dtype: int64
```

Out[100…
```
(1208, 30)
```

导出数据

In [101…
```
# 除质押数据
df.to_excel("data/df.xlsx",index=False)
# 股权质押
df_pt_filtered.to_excel("data/df_Equity_Pledge.xlsx",index=False)
```

维度分析

In [13]:
```
df=pd.read_excel('data\df.xlsx')
```

```
<>:1: SyntaxWarning: invalid escape sequence '\d'
<>:1: SyntaxWarning: invalid escape sequence '\d'
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_26784\3396205958.py:1: SyntaxWarnin
g: invalid escape sequence '\d'
  df=pd.read_excel('data\df.xlsx')
```

In [14]:
```
df.columns
```

Out[14]:
```
Index(['股票代码', '股票简称', '行业代码1', '行业名称1', '应收账款与收入比', '应
收账款周转率A', '应收账款周转天数A',
       '存货周转率A', '年份', '营业毛利率', '营业成本率', '销售费用率', '财务费
用率', '研发费用率', '总资产增长率A',
       '营业收入增长率A', '营业总收入', '营业收入', '减：所得税费用', '净利润',
'应收账款净额', '上年应收账款净额',
       '应收账款平均余额', '应收账款增长率', '经营活动产生的现金流量净额', '投资
活动产生的现金流量净额',
       '筹资活动产生的现金流量净额', '前五大客户销售额和', '前五大客户销售额占比
和', '员工人数'],
      dtype='object')
```

In [15]:
```
df.dtypes
```

Out[15]:
```
股票代码                 int64
股票简称                object
行业代码1               object
行业名称1               object
应收账款与收入比           float64
应收账款周转率A           float64
应收账款周转天数A          float64
存货周转率A             float64
年份                   int64
营业毛利率              float64
营业成本率              float64
销售费用率              float64
财务费用率              float64
研发费用率              float64
总资产增长率A            float64
营业收入增长率A           float64
营业总收入              float64
营业收入                float64
减：所得税费用            float64
净利润                 float64
应收账款净额             float64
上年应收账款净额           float64
应收账款平均余额           float64
应收账款增长率            float64
经营活动产生的现金流量净额     float64
投资活动产生的现金流量净额     float64
筹资活动产生的现金流量净额     float64
前五大客户销售额和          float64
前五大客户销售额占比和        float64
员工人数                int64
dtype: object
```

财务税务维度

In [16]:
```
df_fx=df[['股票代码', '股票简称','年份', '行业代码1', '行业名称1',
          '营业收入增长率A','应收账款增长率','应收账款净额','销售费用率',
          '营业毛利率','存货周转率A','减：所得税费用', '净利润',
          '研发费用率','营业收入','员工人数','应收账款与收入比','应收账款平均余额
          '前五大客户销售额和', '前五大客户销售额占比和','营业总收入',
          '经营活动产生的现金流量净额', '投资活动产生的现金流量净额',
      '筹资活动产生的现金流量净额']]
```

In [17]:
```
df_fx.head()
```

Out[17]:

	股票代码	股票简称	年份	行业代码1	行业名称1	营业收入增长率A	应收账款增长率	应收账款净额	销售费用率	营业毛利率
0	153	丰原药业	2018	C27	医药制造业	0.034883	0.016598	6.391291e+08	0.268058	0.389499
1	153	丰原药业	2019	C27	医药制造业	0.042536	0.057615	6.759525e+08	0.265346	0.381550
2	153	丰原药业	2020	C27	医药制造业	0.054685	0.003189	6.781084e+08	0.234881	0.346629
3	153	丰原药业	2021	C27	医药制造业	0.006766	-0.036313	6.534843e+08	0.186691	0.301077
4	153	丰原药业	2022	C27	医药制造业	0.176302	0.059459	6.923398e+08	0.169947	0.282284

In [18]:
```
df_fx.to_excel("data/df_fx.xlsx",index=False)
```

In [19]:
```
df=pd.read_excel('data\df_fx.xlsx')
```

```
<>:1: SyntaxWarning: invalid escape sequence '\d'
<>:1: SyntaxWarning: invalid escape sequence '\d'
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_26784\174752311.py:1: SyntaxWarning:
invalid escape sequence '\d'
  df=pd.read_excel('data\df_fx.xlsx')
```

In [20]:
```
df.columns
```

Out[20]:
```
Index(['股票代码', '股票简称', '年份', '行业代码1', '行业名称1', '营业收入增长率
A', '应收账款增长率', '应收账款净额',
       '销售费用率', '营业毛利率', '存货周转率A', '减：所得税费用', '净利润',
'研发费用率', '营业收入', '员工人数',
       '应收账款与收入比', '应收账款平均余额', '前五大客户销售额和', '前五大客户
销售额占比和', '营业总收入',
       '经营活动产生的现金流量净额', '投资活动产生的现金流量净额', '筹资活动产生
的现金流量净额'],
      dtype='object')
```

营业收入与应收账款联动分析

In [21]:
```
# 所需指标营业收入增长率，应收账款增长率
df["营业收入行业均值"]=df.groupby('年份')['营业收入'].transform('mean')
df["应收账款净额行业均值"]=df.groupby('年份')['应收账款净额'].transform('mean')


df["营业收入增长率A行业均值"]=df.groupby('年份')['营业收入增长率A'].transform('me
df["应收账款增长率行业均值"]=df.groupby('年份')['应收账款增长率'].transform('mean
```

In [32]:
```
# 检查2018年的应收账款增长率行业均值
df_2018 = df[df['年份'] == 2018]

# 计算2018年行业应收账款增长率的均值
mean_2018 = df_2018['应收账款增长率'].mean()

# 输出2018年的行业均值用于验证
print(f"2018年应收账款增长率行业均值: {mean_2018}")

# 检查2018年的应收账款增长率行业均值
df_2019 = df[df['年份'] == 2019]

# 计算2018年行业应收账款增长率的均值
mean_2019 = df_2019['应收账款增长率'].mean()

# 输出2018年的行业均值用于验证
print(f"2018年应收账款增长率行业均值: {mean_2019}")
```

```
2018年应收账款增长率行业均值: 1.4280140225219737
2018年应收账款增长率行业均值: 0.38383354276189247
```

销售费用率与营收增长联动分析

In [23]:
```
# 所需指标销售费用率，营业收入增长率
df['销售费用率变动'] = df.groupby('股票代码')['销售费用率'].diff()
df["销售费用率行业均值"]=df.groupby('年份')['销售费用率'].transform('mean')
```

毛利率与存货联动分析

In [24]:
```
# 所需指标毛利率，存货周转率
df['毛利率变动'] = df.groupby('股票代码')['营业毛利率'].diff()
df['存货周转率变动'] = df.groupby('股票代码')['存货周转率A'].diff()
```

税费分析

In [25]:
```
# 所需指标所得税费用，净利润
df['每元净利润支付税费']=df['减：所得税费用']/df['净利润']
df["每元净利润支付税费行业均值"]=df.groupby('年份')['每元净利润支付税费'].transfo
```

行业业务维度

研发费用比较

In [26]:
```
# 所需指标研发费用
df["研发费用率行业均值"]=df.groupby('年份')['研发费用率'].transform('mean')
```

人均产值

In [27]:
```
# 所需指标营业收入，员工人数
df["员工人数行业均值"]=df.groupby('年份')['员工人数'].transform('mean')
df['人均产值']=df['营业收入']/df['员工人数']
df["人均产值行业均值"]=df.groupby('年份')['人均产值'].transform('mean')
```

销售回款周期

In [28]:
```
# 所需指标应收账款与收入比
df['日均销售收入'] = df['营业收入'] / 365
df['应收账款平均账龄'] = df['应收账款平均余额'] / df['日均销售收入']
```

客户集中度

In [29]:
```
# 所需指标五大客户销售额、总收入
df["前五大客户销售额和行业均值"]=df.groupby('年份')['前五大客户销售额和'].transfo
df['客户集中度'] = df['前五大客户销售额和'] / df['营业总收入']
```

In [30]:
```
df.head()
```

Out[30]:

	股票代码	股票简称	年份	行业代码1	行业名称1	营业收入增长率A	应收账款增长率	应收账款净额	销售费用率	营业毛利率
0	153	丰原药业	2018	C27	医药制造业	0.034883	0.016598	6.391291e+08	0.268058	0.389499
1	153	丰原药业	2019	C27	医药制造业	0.042536	0.057615	6.759525e+08	0.265346	0.381550
2	153	丰原药业	2020	C27	医药制造业	0.054685	0.003189	6.781084e+08	0.234881	0.346629
3	153	丰原药业	2021	C27	医药制造业	0.006766	-0.036313	6.534843e+08	0.186691	0.301077
4	153	丰原药业	2022	C27	医药制造业	0.176302	0.059459	6.923398e+08	0.169947	0.282284

In [31]:
```
df.to_excel("data/df_fx.xlsx",index=False)
```