

对于应收账款，作为财务的你是否在关心：谁欠钱？谁欠的多？谁欠的久？

当应收账款未能按时收回时，就会降低企业资金的使用效率，使企业效益下降。

在赊销

业务中，企业发出商品并开票确认收入后，货款却不能同步到账，这种没有货款回笼的销售收入，势必会形成没有现金流入的会计利润，企业还需垫支由此产生的相关税款。

对于应收账款的管理成本、清欠过程中的中介费用等都会增加企业的现金流出，这不仅占用了大量的流动资金，还将影响企业现金流的正常运转。

在今天的案例中，

为规避应收账款变成坏账的风险，小秘书将与大家分享跟应收账款账龄分析相关的实用工具，教大家如何借助数字化技术Python依据应收账款数据的“凭证日期”统计账龄。

在案例中，我们将运用先进先出的原则：

- 如果应收账款借方发生额已经实现回款，那么，这部分交易不影响我们对应收账款龄的统计；
- 倒序统计应收账款余额分布在哪几笔借方发生里（定义为“实际应收”），即可按实际应收借方发生的日期统计应收账款龄。

（在本次案例中，我们暂不考虑应收账款借方红字的情况）

一般来说，应收账款清账操作需要在核算系统中找到对应的应收账款借方凭证，然后对该笔凭证进行冲销。

在这种情况下，统计应收账款账龄只需要找到未冲销的应收账款，计算凭证日期/交易日期至统计日期的间隔时长就可以了。

我们使用先进先出的方法统计截止2020/9/30的应收账款龄，账龄分段格式如下：

	1-3月	3月-1年	1-2年	3年以上
客户1	*	*	*	*
客户2	*	*	*	*

1

数据准备

引入应收账款凭证数据：

```
import pandas as pd
from datetime import datetime, date
import warnings
warnings.filterwarnings("ignore")
pd.options.display.float_format = '{:,.2f}'.format
file = '?????.xlsx'
df = pd.read_excel(file, sheet_name='?????')
df.fillna(0,inplace=True)
df
```

	客户编号	总账科目	凭证编号	D/C	过账日期	本币金额	摘要	年	月
0	3100003	1221010000	100048	S	2013-12-31	33,000.00	销售货物33件	2013	12
1	3100003	1221010000	100140	H	2014-06-05	-3,000.00	客户回款	2014	6
2	3100006	1221010000	502734	S	2019-12-14	60,000.00	销售货物60件	2019	12
3	3100015	1221010000	300000	S	2019-01-12	25,000.00	销售货物25件	2019	1
4	3100035	1221010000	502144	S	2019-10-11	100,000.00	销售货物100件	2019	10
...
283	29000055	1221010000	102271	H	2020-09-30	-60.00	市场营销费	2020	9
284	29000055	1221010000	102272	H	2020-09-30	-200.00	市场营销费	2020	9
285	29000055	1221010000	102273	H	2020-09-30	-60.00	市场营销费	2020	9
286	29000055	1221010000	102274	H	2020-09-30	-100.00	市场营销费	2020	9
287	29000055	1221010000	102275	H	2020-09-30	-200.00	市场营销费	2020	9

288 rows*9 columns

2

数据处理

	客户编号	总账科目	凭证编号	D/C	过账日期	本币金额	摘要	年	月	实际应收
0	23500000	1221010000	1001003	S	2020-08-10	5,000.00	销售货物5件	2020	8	5,000.00
1	23500000	1221010000	1001251	S	2019-10-26	10,000.00	销售货物10件	2019	10	10,000.00
2	23500000	1221010000	1000814	S	2019-07-24	10,000.00	销售货物10件	2019	7	6459.37

3

数据计算

计算每笔实际应收账款截止2020/9/30的账龄，账龄以月份数据表示，不足一月的按照天数/30计算小数。

```

enddate = date(2020, 9, 30)
def totalmonth(startdate):
return round((enddate.year-startdate.year)*12 + (enddate.mon
th-startdate.month) + (enddate.day-startdate.day)/30, 2)
df3['??'] = df3['????'].map(totalmonth)
df3.loc[df3['????'] == 23500000]

```

	客户编号	总账科目	凭证编号	D/C	过账日期	本币金额	摘要	年	月	实际应收	账龄
0	23500000	1221010000	1001003	S	2020-08-10	5,000.00	销售货物5件	2020	8	5,000.00	1.67
1	23500000	1221010000	1001251	S	2019-10-26	10,000.00	销售货物10件	2019	10	10,000.00	11.13
2	23500000	1221010000	1000814	S	2019-07-24	10,000.00	销售货物10件	2019	7	6459.37	14.20

计算账龄区间：使用条件判断

- ? 当账龄小于等于3个月时，落入账龄区间 1-3月；
- ? 当账龄大于3个月小于等于12个月时，落入账龄区间 3月-1年；
- ? 当账龄大于12个月小于等于24个月时，落入账龄区间 1-2年；
- ? 当账龄大于36个月时，落入账龄区间 3年以上。

```

def age(totalmonth):
if totalmonth<=3:
age='1-3?'
elif totalmonth<=12:
age='3?-1?'
elif totalmonth<=24:
age='1-2?'
else:
age='3???'
return age

```

```
df3['????'] = df3['??'].map(age)
df3.loc[df3['????'] == 23500000]
```

	客户编号	总账科目	凭证编号	D/C	过账日期	本币金额	摘要	年	月	实际应收	账龄	账龄区间
0	23500000	1221010000	1001003	S	2020-08-10	5,000.00	销售货物5件	2020	8	5,000.00	1.67	1-3月
1	23500000	1221010000	1001251	S	2019-10-26	10,000.00	销售货物10件	2019	10	10,000.00	11.13	3月-1年
2	23500000	1221010000	1000814	S	2019-07-24	10,000.00	销售货物10件	2019	7	6459.37	14.20	1-2年

账龄统计：

```
pd.pivot_table(df3, index['????'],
columns=['????'],
values=['????'], aggfunc=
sum, fill_value=0?)
```

账龄区间	实际应收			
	1-2年	1-3月	3年以上	3月-1年
客户编号				
3100003	0.00	0.00	30000.00	0.00
3100006	0.00	0.00	0	60,000.00
3100015	25,000.00	0.00	0	0.00
3100035	0.00	0.00	0	330,000.00
3100048	0.00	4,260.00	0	0.00
3100055	0.00	0.00	0	50,000.00
3100057	0.00	9,000.00	0	0.00
3100063	0.00	0.00	0	1,250.00
4000001	0.00	9,142.48	0	63,000.00
20000007	0.00	0.00	0	0.00
20000079	0.00	80,000.00	0	0.00
23500000	6459.37	5,000.00	0	10,000.00
23500005	0.00	0.00	0	0.00
23500007	0.00	120,000.00	0	114,056.45
23500009	20,000.00	0.00	0	0.00

以上就是用Python自动计算应收账款账龄的操作步骤了。公司的应收账款在流动资产中具有举足轻重的地位，如果公司的应收账款能及时收回，那么

公司的资金使用效率将能大幅提高！

?图片版权归财码数字化所有，如对版权有异议，请联系后台议定处理。

本文由财码数字化整理发布，如需转载请务必注明以上信息。