

## 预习作业题一

投资人小明和小白分别在拍拍贷投资 1000 元。小明投了一笔散标，评级为 AA，利率 10%，期限 12 个月；小白投了彩虹计划，利率也是 10%，期限 12 个月。请分别计算小明和小白的应收计划：

### 1. 小明和小白下一个收款日是哪一天，最后一个收款日是哪一天？

答：假设小明与小白都是 2017 年 6 月 1 日投的资金，小明投的是散标且是赔标，还款人是按照“等额本息”的方式来进行还款的。每个月还固定的金额。小明的下一个收款日是 7 月 1 日，最后一个收款日为 2018 年 6 月 1 日。小白投的是彩虹计划，下一个收款日和最后一个收款日都为 12 个月后的 1 号。

### 2. 小明和小白每一期预计能收到多少金额？

答：小明每期收到的金额满足公式：

每期还款额 = (借款金额 \* 月利率) \* power(1 + 月利率, 期限) / (power(1 + 月利率, 期限) - 1) = (1000 \* 10% / 12) \* power(1 + 10% / 12, 12) / (power(1 + 10% / 12, 12) - 1) = 87.91 元

小白彩虹计划有利息复投，所以每期收到的金额为：

一月：1000 \* 10% / 12 = 8.33 元

二月：(1000 + 8.33) \* 10% / 12 = 16.7 元

三月：(1000 + 16.7) \* 10% / 12 = 25.17 元

四月：(1000 + 25.17) \* 10% / 12 = 33.54 元

五月：(1000 + 33.54) \* 10% / 12 = 42.15 元

六月：(1000 + 42.15) \* 10% / 12 = 51.62 元

七月：(1000 + 51.62) \* 10% / 12 = 60.38 元

八月:  $(1000+60.38)*10\%/12=69.21$  元

九月:  $(1000+69.21)*10\%/12=78.12$  元

十月:  $(1000+78.12)*10\%/12=87.1$  元

十一月:  $(1000+87.1)*10\%/12=96.16$  元

十二月:  $(1000+96.16)*10\%/12=105.29$  元

(具体实现代码见附录的 homework1.py)

### 3. 他们的应收总金额一样吗?为什么?

答:他们两人收到的总金额不一样。

小明收到的总金额为:  $87.91*12=1054.92$ (元) 小白收到的总金额为:  $1000+105.29=1105.29$ (元)

用程序跑出来的结果如下: (代码见附录的 homework1.py)

```
In [12]: runfile('C:/Users/skyamz/Desktop/homework1.py', wdir='C:/Users/skyamz/Desktop')
下一个收款日为: 2017 年 7 月 1 日
最后一个收款日为: 2018 年 5 月 27 日
小明的总收益为(赔标): 1054.99
小白的总收益为(彩虹计划): 1104.71
```

### 4. 从这个场景出发考虑, 作为投资人, 应该注意什么?

答: 如果不考虑逾期未还或坏账的可能性, 那么投资人应该在投资之前计算自己的收益, 以便更好地投资。根据上面的计算结果, 显然投彩虹计划的收益比投散标 AA 的收益高。

如果考虑坏账的情况, 投赔标稳定, 平台会到期赔付。但彩虹计划就不能。所以考虑稳定性选择赔标。

由于彩虹计划是不能随时退出的, 所以在不考虑资金的流动性的情况下, 选择彩虹计划的收益高, 若是资金容易流动, 则买赔标可以随时抛售。

## 预习作业题二

基于 LC、LP 数据，统计借款成功日期在 2016 年 2 月期间的标，截止 2016 年 10 月底一共有多少 30 天(含)以上逾期金额。

答:1. 首先筛选出借款成功日期在 2016 年 2 月期间的标:

```
datelist = lc [(lc['借款成功日期']>datetime(2016,2,1)) & (lc['借款成功日期']<datetime(2016,3,1))]
```

结果如下:

	ListingId	借款金额	借款期限	借款利率	借款成功日期	初始评级	借款类型	是否首标	年龄	性别	...	户认证	视频认证	学历认证	征信认证	淘宝认证	历史成功借款次数	历史成功借款金额	总待还本金	历史正常还款期数	历史逾期还款期数
35639	7636461	6000	12	22.0	2016-02-02	D	普通	是	31	男	...	未成功认证	未成功认证	成功认证	未成功认证	未成功认证	0	0.0	0.00	0	0
36593	7775221	5000	12	24.0	2016-02-02	E	普通	是	40	男	...	未成功认证	未成功认证	未成功认证	未成功认证	未成功认证	0	0.0	0.00	0	0
36620	7779931	6000	12	22.0	2016-02-02	D	普通	是	41	男	...	未成功认证	未成功认证	未成功认证	未成功认证	未成功认证	0	0.0	0.00	0	0
36691	7787961	4500	12	24.0	2016-02-03	E	普通	是	40	女	...	未成功认证	未成功认证	未成功认证	未成功认证	未成功认证	0	0.0	0.00	0	0
48370	9135121	4200	12	22.0	2016-02-29	D	其他	否	27	男	...	成功认证	成功认证	成功认证	成功认证	成功认证	1	2500.0	1265.56	3	0
48375	9135721	5000	12	20.0	2016-02-29	C	其他	否	33	女	...	未成功认证	未成功认证	未成功认证	未成功认证	未成功认证	1	4500.0	3420.78	3	0
48379	9136001	2950	12	22.0	2016-02-29	D	其他	否	23	男	...	成功认证	成功认证	未成功认证	未成功认证	未成功认证	13	64861.0	12049.97	48	3
48382	9136421	5300	6	20.0	2016-02-29	C	其他	否	31	女	...	未成功认证	未成功认证	成功认证	未成功认证	未成功认证	2	7000.0	3694.41	4	0
48386	9137001	2200	6	20.0	2016-02-29	C	其他	否	37	男	...	未成功认证	成功认证	未成功认证	成功认证	未成功认证	1	4500.0	2278.00	3	0

8311 rows × 21 columns

根据统计结果，借款成功日期在 2016 年 2 月期间的标一共有 8311 条。

2. 将 LC 表与 LP 表内联起来:newdata = lp.merge(datelist,how = 'inner',on = 'ListingId') 结果如图:

共 81818 条标的数据。

3. 计算到 2016 年 10 月底，逾期的标的data2= newdata[newdata['到期日期']<=datetime(2016,10,31)].copy() data2['overdue\_days'] = (data2['还款日期']-data2['到期日期']).apply(lambda x:x.days)

	ListingId	期数	还款状态	应还本金	应还利息	剩余本金	剩余利息	到期日期	还款日期	recorddate	...	户认证	视频认证	学历认证	征信认证	淘宝认证	历史成功借款次数	历史成功借款金额	总待还本金	历史正常还款期数
0	7636461	1	1	451.57	109.99	0.00	0.00	2016-03-02	2016-02-28	2017-02-22	...	未成功认证	未成功认证	成功认证	未成功认证	未成功认证	0	0.0	0.00	0
1	7636461	2	1	459.84	101.72	0.00	0.00	2016-04-02	2016-03-28	2017-02-22	...	未成功认证	未成功认证	成功认证	未成功认证	未成功认证	0	0.0	0.00	0
81815	9136421	6	1	920.22	15.35	0.00	0.00	2016-08-29	2016-08-29	2017-02-22	...	成功认证	成功认证	成功认证	成功认证	成功认证	2	7000.0	3694.41	4
81816	9137001	1	1	351.69	36.66	0.00	0.00	2016-03-29	2016-03-02	2017-02-22	...	未成功认证	成功认证	未成功认证	成功认证	未成功认证	1	4500.0	2278.00	3
81817	9137001	2	3	1848.31	0.00	0.00	0.00	2016-04-29	2016-03-16	2017-02-22	...	未成功认证	成功认证	未成功认证	成功认证	未成功认证	1	4500.0	2278.00	3

81818 rows × 30 columns

	ListingId	期数	还款状态	应还本金	应还利息	剩余本金	剩余利息	到期日期	还款日期	recorddate	...	户认证	视频认证	学历认证	征信认证	淘宝认证	历史成功借款次数	历史成功借款金额	总待还本金	历史正常还款期数	历史逾期还款期数	overdue_days
0	7636461	1	1	451.57	109.99	0.0	0.0	2016-03-02	2016-02-28	2017-02-22	...	未成功认证	成功认证	未成功认证	未成功认证	未成功认证	0	0.0	0.00	0	0	-3
1	7636461	2	1	459.84	101.72	0.0	0.0	2016-04-02	2016-03-28	2017-02-22	...	未成功认证	成功认证	未成功认证	未成功认证	未成功认证	0	0.0	0.00	0	0	-5
81815	9136421	6	1	920.22	15.35	0.0	0.0	2016-08-29	2016-08-29	2017-02-22	...	成功认证	成功认证	成功认证	成功认证	成功认证	2	7000.0	3694.41	4	0	0
81816	9137001	1	1	351.69	36.66	0.0	0.0	2016-03-29	2016-03-02	2017-02-22	...	未成功认证	成功认证	未成功认证	成功认证	未成功认证	1	4500.0	2278.00	3	0	-27
81817	9137001	2	3	1848.31	0.00	0.0	0.0	2016-04-29	2016-03-16	2017-02-22	...	成功认证	成功认证	成功认证	成功认证	成功认证	1	4500.0	2278.00	3	0	-44

59567 rows × 31 columns

共有 59567 条标的

```
4. 计算出超过 30 天的标的overdue30_id = np.unique(data2[(data2['到期日期']<datetime(2016,10,31)) &
(data2['overdue_days']>=30)].ListingId) data_no2[(data_no2.ListingId.apply(lambda x: True if x in overdue30_id else False))]
```

## 预习作业题三

给出一个你想要实现的数据可视化类产品思路或 demo 答:数据可视化类基于不同的行业有不同的可视化需求,根据不同的人,可视化也有不同的喜好。总的来说,数据可视化,首先要做到好看,所以产品要有丰富的图表,不同的配色方案与布局方案;其次,还要做到能够直观迅速地反映问题,所以要有预警提醒,要定期更新;最后还要做到预测与分析,详细分析数据之间的关联并展示,能够帮助用户全方位地理解数据。

我想要实现的数据可视化类产品具有以下功能:

- 可供选择不同行业的数据可视化图表。用户根据不同的行业进入对应的操作页面,选择想要可视化的数据与可视化的形式。
- 操作简单易懂。操作界面有不同的布局,直接进行添加图标与数据即可生成一个反映问题的界面,配色可以自选。
- 可以产生不同种可视化文档,例如单个保存或全部生成 pdf。
- 可以自定义布局以及图表。要实现上述功能,需要我们调查不同行业需要具体呈现什么数据,审查图表的人的喜好也需要调查。在不同场景中的应用不同。

## 附录:

➤ homework1.py

```
from datetime import timedelta,date
```

```
import datetime
```

```

start_date = datetime.datetime(2016, 6, 1)

p1 = timedelta(days=30)

p2 = timedelta(days=30*12)

next_day = start_date + p1

last_day = start_date + p2

print "下一个收款日为:", next_day.year, '年', next_day.month, '月',
next_day.day, '日'

print "最后一个收款日为:", last_day.year, '年', last_day.month, '月',
last_day.day, '日'

amount = 1000

mon_rate = 0.1/12

period = 12

Ming_total_amt =
((amount*mon_rate)*pow(1+mon_rate, period)/(pow(1+mon_rate, period)-1))*12
Bai_total_amt = amount*pow(1+0.1/12, 12)

print '小明的总收益为(赔标):', round(Ming_total_amt, 2)

print '小白的总收益为(彩虹计划):', round(Bai_total_amt, 2)

```

➤ homework2:

```

import sys reload(sys)

sys.setdefaultencoding('utf-8')

import pandas as pd

from datetime import datetime

lc =pd.read_csv("LC.csv", parse_dates = ['借款成功日期'])

```

```
datelist = lc [(lc['借款成功日期']>=datetime(2016, 2, 1)) & (lc['借款成功日期']<datetime(2016, 3, 1))]
```

```
lp = pd.read_csv("LP.csv", parse_dates = ['还款日期', '到期日期', 'recorddate'])
```

```
newdata = lp.merge(datelist, how = 'inner', on = 'ListingId')
```

```
newdata['还款日期'] = newdata['还款日期'].apply(lambda x: datetime.strptime(x, '%Y-%m-%d') if x != '\\N' else datetime(2050, 1, 1))
```

```
data2= newdata[newdata['到期日期']<=datetime(2016, 10, 31)].copy()
```

```
data2['overdue_days'] = (data2['还款日期']-data2['到期日期']).apply(lambda x:x.days)
```

```
overdue30_id = np.unique(data2[(data2['到期日期']<datetime(2016, 10, 31)) & (data2['overdue_days']>=30)].ListingId)
```

```
data_no2[(data_no2.ListingId.apply(lambda x: True if x in overdue30_id else False))]
```