中央财经大学

Central University of Finance and Economics

课　　程　大数据抽样技术

实验名称　贷款数据抽样实验

指导老师　孙志猛

专业班级　中财 17 统计专硕班

实验日期　2017/10/21

目录

一、 研究问题与目的

1.1 研究问题

依据大数据抽样课上所讲授的抽样技术内容进行本次试验，利用样本信息量公式推导样本质量公式，并对四种抽样方式进行样本质量层面的分析比较，利用样本质量公式衡量四种抽样方式（简单随机抽样，分层抽样，整群抽样，两阶段抽样）的优劣。并利用贷款数据 loanstatC 进行实证检验，尝试对给定的样本量计算每种抽样方式所能达到的平均样本质量，并对每种抽样方式的的最优样本量进行寻找，最终找出公认的最优样本质量 95%时的每种抽样技术的样本数量。

1.2 研究目的

本次大数据抽样实验的分析目的有以下几种：

1) 熟练应用抽样信息量公式推导抽样质量公式，并利用抽样质量公式评估抽样质量。
2) 依据抽样样本质量公式计算给定的样本数量的四种抽样方法的样本质量，并比较各样本数量下的四种不同的抽样方法样本质量，并进行绘图。
3) 根据任意给定的样本质量，利用抽样样本质量公式计算四种抽样方式所需要的抽样数量。并找出样本质量为 95%时的四种抽样方法所需要的抽样数量。
4) 找出简单随机抽样，分层抽样，整群抽样，两阶段抽样四中抽样方式在贷款数据 loanstatC 中的最优样本质量。

二、 数据的获取与预处理

2.1 数据的获取与变量的选择

本次使用的是贷款数据中的 loanstatC 数据集。而本次抽样实验所选取的变量是 loanstatC 数据表中 id, member_id, loan_amnt, annual_inc, grade, issue_d 的变量。变量说明如下：

表 2-1 实验变量说明表

变量名		详细说明	取值范围	备注
键名	id	某一条贷款业务的唯一主键名，标识唯一某次贷款交易	[10000000-40000000]	此两项指标为相应的贷款业务和贷款人 id，完全是标识无单位。
	member_id	反映贷款用户在 loanstatC 贷款下的标识号	[137225-40860827]	
数量特征	loan_amnt	连续型变量 反映贷款用户在一条贷款记录中的贷款数量总额。	[1000-35000]	正向指标 贷款数量越大说明贷款额度高，贷款分级高。 单位为：美元
	annual_inc	连续型变量 反映在一年内的年收入	[3000-7500000]	正向指标 年收入越高，贷款额度越高，贷款分级越高。 单位：美元
分组信息	grade	分类型变量 反映每名顾客的信用分级	[A,B,C,D,E,F]	离散型变量 A 代表信用评级越高 F 代表信用评级最低 无单位
	issue_d	发生此条交易的时间包括年月	发生的交易时间为：2014-01 到 2014-12 为 2014 年一整年	日期型变量 记录每条交易的年份和月份

2.2 数据的预处理

```
1.  ################数据处理
2.  library(sampling)  #导入数据包
3.  library(dplyr)
4.  library(ggplot2)
5.  setwd("D:/大数据作业/抽样技术/")
6.  summary(mydata)
7.  mydata<-
8.    read.csv("LoanStats3c.csv",header = T,skip=1)%>%
9.    dplyr::select(id,member_id,loan_amnt,annual_inc,grade,issue_d)#整理实验数据
10. write.csv(mydata,"D:/大数据作业/抽样技术/mydata.csv",row.names = TRUE)  #输出
    实验总体集
11. any(is.na(mydata))  #检查数据中是否存在缺失值
12. which(is.na(mydata))  #查看缺失值位置
13. mydata<-mydata[complete.cases(mydata),]  #删除含缺失值的记录
```

本次使用的 R 包包括 sampling,dplyr 以及绘图包 ggplot2。首先是数据处理与导入问题，由于数据量较大，变量较多，所以选取几个实验用的变量重新构成实验矩阵，为了减少变量的命名使实验空间变得简单，所以采取 dplyr 包的管道处理，得到的 mydata 矩阵重新输出为 csv 文件，方便下次的读取。用 any 语句进行缺失值的查询，并用 complete.cases 语句进行缺失值的删除，最终得到实验矩阵。

2.3 数据研究方法

本次试验根据实验目的采用的试验方法为利用 R 软件进行数据统计分析实验，采用抽样技术中不同种抽样方法，包括简单随机抽样，分层抽样，整群抽样，两阶段抽样的抽样方法，并通过样本质量公式进行对抽样样本的样本相似度的估算，并进行最有样本量的确定，本绘制样本质量曲线。最后基于 loanstatC 数据上，对每一中抽样方式进行优缺点的评估。

样本质量的含义是样本结构与数据整体结构的相似性。对于离散数据，假设整体数据集为 D，包含 1 个指标取值即和为$\{x_1, x_2, x_3, \dots, x_K\}$,在点$x_i$出有$N_i$观测（$\mathrm{i} = 1,2,3, \dots, N_K$），我们可以根据 Kullback-Laible 信息量量衡量抽样数据集 S 和 D 的差异性公式如下：

$$\mathrm{I}(S, D) = \sum_{i=1}^{K} (f_{Si} - f_{Di}) log \frac{f_{Si}}{f_{Di}}$$

可以看出 Kullback-Laible 信息量基于频率衡量数据集的差异性，我们用

$$Q(S,D) = e^{-I(S,D)}$$

衡量样本质量，称为样本数据的 S 可对整体数据 D 的样本质量。Q（S,D）数值越大越接近于 1，则代表样本质量较高的样本，其分析结果越接近于整体分析的结果。

2.4 抽样存在的问题及解决方案

1．连续型变量转化为离散型变量。因为本次实验的采用的两个计算样本相似的指标 loan_amnt 和 annual_inc 均为连续性变量，一般不能直接计算样本质量 Q（S,D），此时，我们采用的是将连续型变量进行适当分组，转化成离散型变量的情况。

2．采取多次抽样平均的方式计算样本质量。因为是随机取样，因此对于同一个样本容量的不同样本的样本质量也会有差异，因此样本质量并不是样本容量的单调函数，而是多了扰动。为了减少抽样扰动的影响，采取了在确定样本质量时候采取多次取样求平均的方法消除随机误差，接下来进行每种抽样方法的讨论。

3. 采取多个指标样本质量的平均样本质量作为抽样样本的样本质量。由于本次实验采取了两个指标，具有两个样本质量，所以需要综合让两个指标的样本质量，因为两个指标的重要程度相同，赋予相同权重，即采取平均样本质量作为整体样本质量。

三、 简单随机抽样实证研究

3.1 简单随机抽样数据准备

如第一章所述，抽取了 loan_amnt 和 annual_inc 两个指标作为数据集样本相似性的度量。由于两个变量均是连续型变量，首先应当进行适当的分组。根据 loan_amnt 的数据分布特征进行将数据切分为 17 个段，并作为 factor 变量储存在 bre1 之中利用 cut 函数计算总体频率储存在 PD1 之中。

```
1.  ################简单随机抽样
2.  ##简单随机抽样提取 loan_amnt 和 annual_inc 两个指标进行样本质量的计算
3.  ##首先进行 annual_inc 指标的样本质量计算
4.  data0<-mydata[,c('loan_amnt','annual_inc')]   #提取所需样本矩阵
5.  attach(data0)
6.  N<-length(annual_inc)
7.  bre1<-c(0,10000*(1:10),150000,100000*(2:5),max(annual_inc))  #进行变量的数据切
    分分组
8.  PD1<-
9.    annual_inc %>%
10.   cut(breaks = bre1) %>%   #划分连续变量分组
11.   table()/N                 #确定总体频率
12. ##简单随机抽样 loan_amnt 指标
13. hist(loan_amnt,xlab="贷款金额",ylab="频数",main="贷款金额直方图
    ",col="lightblue") #画出直方图观察分布
14. bre2<-c(0,5000*(1:7))
15. PD2<-
16.   loan_amnt %>%
17.   cut(breaks = bre2) %>%    #划分连续变量分组
18.   table()/N                 #确定总体频率
```

计算完毕 loan_amnt 的各个分段的总体分布频率后，就该计算第二个指标 annual_inc 的总体分布频率，首先画出直方图来看该指标在总体上的分布特征，以此来划分数据的分段间隔。直方图如下所示：

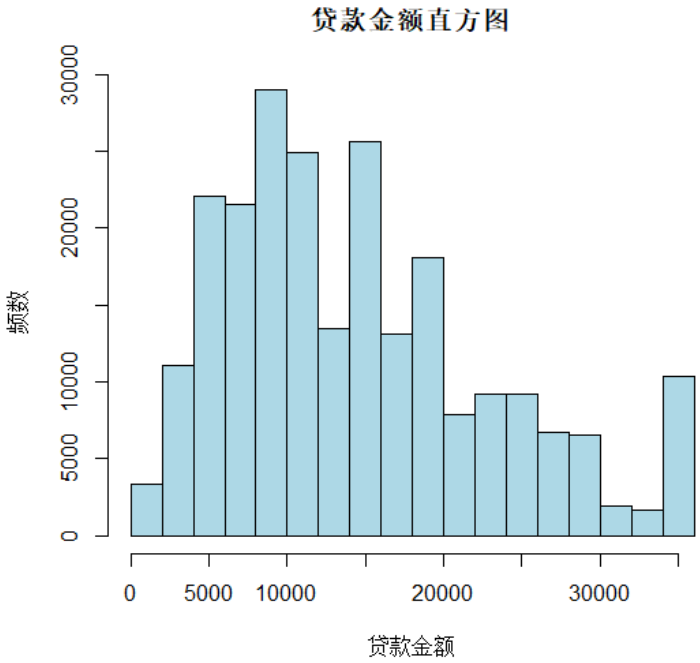

图 3-1 贷款金额直方图

由 annual_inc 指标的数量分布可以看出，大部分的年收入都分布在 5000-20000 美元之间，且成右偏分布，但大体上分布均匀，因此在切割分组时，只需要均匀地切割分段即可。根据数据分布特征每间隔 5000 美元，进行数据切分为 7 个段，并作为 factor 变量储存在 bre2 之中利用 cut 函数计算总体频率储存在 PD2 之中。

3.2 确定简单随机抽样样本数量

```
##确定总抽样个数 samp
x<-seq(6,20,by=0.1)
y<-2^(x)
samp<-round(y)[-c(1:51)][-c(69:90)]    #确定样本抽取个数
n<-length(samp)
samp<-as.matrix(samp)
```

首先生成 x 序列 6:20 每间隔 0.1 的步长，然后 2 的指数函数生成抽样数据量，最后去除部分不满足要求的样本量，是样本量尽量的符合数据采集量要求，且满足逐步递增的效果。生成的抽样样本数量储存在 samp 向量里，一共是 68 个样本容量。用于以后的计算。

3.3 定义抽样函数与信息量计算函数

```
##定义函数
###定义抽样函数
fun1<-function(i,datasam){
  sub<-sample(nrow(data0),i)
  if(datasam=="annual_inc")  p<-data0[sub,]$annual_inc
  if(datasam=="loan_amnt")  p<-data0[sub,]$loan_amnt
  p<-c(p,matrix(NA,1,samp[n]-length(p)))
  return(p)
}
###定义信息量计算函数
fun2<-function(datasam1,bre,PD){
  datasam1<-cut(na.omit(datasam1),breaks = bre)
  PS<-table(datasam1)/length(na.omit(datasam1))+0.0000000001
#防止某个分组概率为零
  J<-sum((PS-PD)*(log(PS/PD)))
  return(J)
}
```

首先定义简单随机抽样的抽样函数 fun1，内置参数 i 为抽取的样本个数，datasam 字符型变量，为抽样的指标。P 为储存抽样样本的向量，因为抽样数量有大有小,所以不够的位数由 NA 空值进行填充。随后定义信息量计算函数 fun2，datasam1 为抽样出来的样本向量，bre 参变量为分组依据，即分组间隔，PD 则是对应 loan_amnt 和 annual_inc 指标的总体分布频率。函数 fun2 的内部参变量 PS 为样本的各个分段的分布频率，在计算 PS 时注意将所有的空值进行剔除，并每个分布频率段上加上 0.000000001，以保证可能有的数据段上没有样本分布而出现零值的现象，由于加的数值较小因此不对结果造成影响。J 为计算出得出信息量。函数返回 J 信息量。

3.4 简单随机抽样实证检验

做好具体的数据和函数的准备工作后，对简单抽样函数的最优样本量，给定样本容量计算样本质量和给定样本容量阈值计算所需的样本容量三方面进行讨论，结果如下：

➢ 简单随机抽样最优样本量

传统抽样理论已经发展较为完善，从理论上保证用抽取的部分样本代替全部样本数据进行建模是把抽样误差控制在一定范围内，在牺牲较小的精度的同时换取较高的计算效率。大样本带来高精度，但同时也损失了计算效率，因此确定了一些选取最优样本的标准。即换句话说，停止增加样本量的条件：

1. 需要达到的最小样本质量 Q（S,D）为 95%，也就是最低承受的样本质量。
2. 前后两次增加样本质量并没有明显改善，设置为前后两次的取样必须达到样本质量 1%的进步才能继续增加样本。

在这样增加样本量基础之上，设定每次增加样本数量为 10 个,起始为 100 个样本。R 平台代码：

```
fun7<-function(lab=fun1){            #lab 函数默认为简单随机抽样
  a<-100
  Q<-vector(mode="numeric",length=0)
  Qm<-0
  while(Qm<0.95||(Qm-Qm1)>0.1){
    for(i in 1:5){
      mr1<-lab(a,"annual_inc")      #确定抽样方式，并在数据集中抽取变量
      j1<-fun2(mr1,bre1,PD1)
      mr2<-lab(a,"loan_amnt")       #确定抽样方式，并在数据集中抽取变量
      j2<-fun2(mr2,bre2,PD2)
      Q[i]<-exp(-(j1+j2)/2)         #计算各种抽样方式的抽样样本质量
    }
    Qm1<-Qm
    Qm<-mean(Q)
    a<-a+10                         #增加样本容量
  }
  print(Qm)
  return(a)
}
```

在 R 语言代码中，lab 为抽取样本方式的标识（即通过 lab 的函数来确定是四种抽样方式中的哪一个，），其中，fun1 为简单抽样方式，fun4 为分层抽样，fun6 为两阶段抽样。可以看出首先起始的数据样本数量为 100 个，且利用 a<-a+10 每次向其中增加 10 个新样本。停止增加样本的条件设定为“Qm<0.95||(Qm-Qm1)>0.001”即，样本质量必须达到 95%以上，且每次增加样本必须具有 0.1%的改善，为了避免抽样的随机性，设置了 for 循环，使得在每个样本数量下都抽取 5 次，取 5 次抽样的平均数，避免抽样随机性导致的品样本质量计算偏差。

简单随机抽样最优样本实验结果如下：

```
> fun7()
[1] 0.9606746
[1] 360
```

基于贷款数据集 loanstatC 数据集的双指标样本相似度的简单随机抽样最优样本量为：360 个，且能达到的样本质量为 96.06%，在这种取样数量上既可以保证样本抽样质量足够高，大于 95%，且再向其中增加样本也无法显著增加样本质量，故最优样本质量为 360 个。

➢ 给定样本容量计算样本质量

利用 3.2 节中生成的 samp 向量储存了抽样样本的数量序列，因此可以利用 samp 向量作为相应的样本容量，并结合 loan_amnt 和 annual_inc 两个指标计算 samp 中样本容量的样本质量，最后根据样本容量个数绘制出随样本数量变化的样本质量分布散点图，所使用的 R 代码如下所示：

```
######实证检验
Q1<-matrix(NA,length(samp),10)
for(i in 1:10){
  ma1<-apply(samp,1,function(x) fun1(x,"annual_inc"))#指标 annual 样本矩阵
  J1<-apply(ma1,2,function(x) fun2(x,bre1,PD1)) #计算每个抽样样本下的样本质量
  ma2<-apply(samp,1,function(x) fun1(x,"loan_amnt")) #指标 loan_amnt 样本矩阵
  J2<-apply(ma2,2,function(x) fun2(x,bre2,PD2))
  Q1[,i]<-exp(-(J1+J2)/2)      #合并两个指标的 KL 信息量
}                              #每个样本容量下产生 10 个样本数据集
Qj<-apply(Q1,1,mean)           #计算每个样本容量下的样本质量均值
huatu<-as.data.frame(cbind(as.vector(samp),as.vector(Qj)))
names(huatu)[1:2]<-c("samp","Qj")
ggplot(huatu,aes(x=samp,y=Qj))+geom_point(size=2,color="red")+ggtitle('简单抽样样本质量与样本容量散点图') +theme(plot.title = element_text(hjust = 0.5,family="myFont",size=18,color="black"),panel.background=element_rect(fill='aliceblue',color='black'))      #利用 ggpolt2 画出图像
```

首先需要生成 ma1 和 ma2 两个抽样样本数据矩阵，利用的皆为 apply 行处理函数，利用 fun1 生成两个指标的抽样样本矩阵，然后利用 apply 列处理对两个抽样矩阵的 KL 信息量进行计算，最后将 J1 和 J2 进行合并，并求解出基于两个指标的样本质量 Qj。但需要注意的是，为消除抽样的随机性影响，在此使用 10 次反复抽样的平均 KL 信息量作为每个计算每个样本容量的样本质量的信息量。并利用 ggplot2 作图包，绘制函数图形，实验结果如下图:

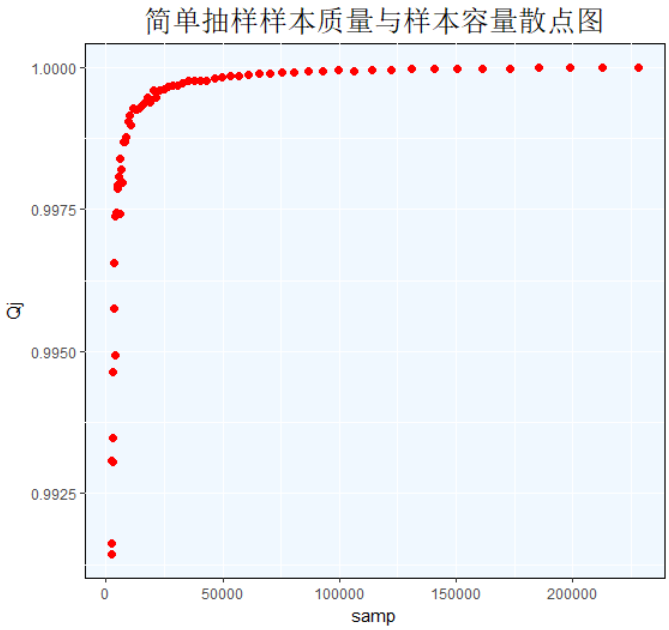

图 3-2 简单抽样样本质量与容量关系

在图中可以清楚地看到，随着样本容量的增加，样本质量也在迅速爬升，大约在简单随机抽样的样本数量增加到 25000 左右时，样本质量增加量的改变较少，并且样本质量逼近于 1。

➢ 给定样本质量阈值计算样本容量

计算给定样本质量阈值为 95%时的简单随机抽样的样本容量，构造函数 fun3 在逐步增加样本量来逼近 95%的阈值，起始的样本容量为 100 个，每次添加的步长为 5，样本质量容许的误差为 0.0005，即确定到 95%±0.5%即可停止增加样本量。对于 fun3 函数中的参变量，Pro1 为设定的阈值，lab 为所使用的抽样函数，其中，fun1 为简单抽样方式，fun4 为分层抽样，fun6 为两阶段抽样。而且为了避免抽样的随机性对实验结果造成影响，内层设置多次抽样取平均的方法来减少抽样随机性。

```
##给定样本质量 95%时，计算样本容量，设定为每次添加样本步长为 5，样本质量允许误差为 0.0005
###给定样本质量来计算样本容量函数
fun3<-function(pro1,lab=fun1){
  a<-100
  Q<-vector(mode="numeric",length=0)
  Qm<-0
  while(abs(Qm-pro1)>0.0005){          #判断 Qm 距离 Pro1 的
    for(i in 1:2){
```

```
      mr1<-lab(a,"annual_inc")
      j1<-fun2(mr1,bre1,PD1)
      mr2<-lab(a,"loan_amnt")
      j2<-fun2(mr2,bre2,PD2)
      Q[i]<-exp(-(j1+j2)/2)
    }
    if(Qm>0.9505) break      #为了防止样本质量>95.05%后进入死循环，选择跳出循环
    Qm<-mean(Q)
    a<-a+10
  }
  print(Qm)
  return(a)
}
```

简单随机抽样的阈值找样本容量的实验结果为:

```
> fun3(0.95,fun1)
[1] 0.9505057
[1] 310
```

基于贷款数据集 loanstatC 数据集的双指标样本相似度的简单随机抽样最优样本量为：310 个，且能达到的样本质量为 95.05%，在这种取样数量使得样本质量已经达到 95%，故简单随机抽样的数量为 310。

也可以利用插值函数，如下代码，分别计算 95%，96%,97%,99%的样本质量的样本容量得出下表。

```
#选择样本拟合样本量和样本质量变化曲线图
x=seq(6,20,by=0.1)
y=2^(x)
samp=round(y)[-c(1:20)]
samp=samp[-c(40:length(samp))]#选取合适数据
sa=data.frame(y=samp,x=c(1:39))
n1=length(samp)
samp=as.array(samp)
ma<-apply(samp,1,fun1)#利用随机抽样函数抽取一组样本
q1=apply(ma,2,fun2)#计算样本质量
s1=smooth.spline(q1,samp)#拟合曲线并预测样本质量在 0.95 的情况下的样本量
pr1=predict(s1,0.95)  #预测 95%的样本容量
ceiling(pr1[[2]])
 pr2=predict(s1,0.96)  #预测 96%的样本容量
ceiling(pr2[[2]])
pr3=predict(s1,0.97)  #预测 97%的样本容量
ceiling(pr3[[2]])
```

```
pr4=predict(s1,0.98)  #预测 99%的样本容量
ceiling(pr4[[2]])
```

表 3-1 不同精度下的最优样本容量表

样本质量	95%	96%	97%	99%
样本容量	310	578	1016	1875

四、 分层抽样实证研究

4.1 分层抽样数据准备

首先在生成的数据矩阵 mydata 中抽取出三个变量分别是 loan_amnt,annual_inc,grade，利用 dplyr 包中的 select 函数形成新的数据矩阵 data3 用于进行分层抽样实验。其次，利用 grade 变量进行数据的分层，并计算每层的总体分布频率储存在 PD3 中，最后利用 order 对 grade 进行分层排序。形成最终数据集 fcdata。

```
####################分层随机抽样
#首先抽取出样本矩阵
data3<-
  mydata%>%
  dplyr::select(loan_amnt,annual_inc,grade)
PD3<-table(data3$grade)/N
fcdata<-data3[order(data3[,3]),]#构造分层抽样样本数据集
```

4.2 确定分层抽样样本数量

分层抽样的样本量的确定与简单随机抽样的相同，共同使用 samp 中的 68 个样本容量。利用这 68 个容量进行数据的分层抽取与检验。

4.3 定义分层抽样函数与信息量计算函数

分层抽样的样本 KL 信息量函数 fun2 与简单随机抽样的函数相同，接下来定义分层抽样样本抽取函数 fun4

```
fun4<-function(s,datasam){
  sub=strata(fcdata,stratanames="grade",size=round(s*PD3[-
1]),method="srswor")
  if(datasam=="annual_inc")  p<-data3[sub$ID_unit,]$annual_inc
  if(datasam=="loan_amnt")  p<-data3[sub$ID_unit,]$loan_amnt
  res<-c(p,matrix(NA,1,samp[n]+5-length(p)))
  return(res)
}
```

在 fun4 中，首先利用 sampling 包中的 strata 函数根据 grade 指标进行“srswor”分层抽样，每层的抽样数量根据总体样本分布 PD3 进行确定，datasam 为字符型变量，标识指标的不同。P 为在分层抽样中，由 data3 标识的数据的抽样样本集合。函数返回值为 res，也就是生成抽样函数矩阵。

4.4 分层抽样实证检验

做好具体的数据和函数的准备工作后，对分层抽样函数的最优样本量，给定样本容量计算样本质量和给定样本容量阈值计算所需的样本容量三方面进行讨论，由于分层抽样的运算代价太高再次就不进行多次取样消除抽样误差了。实验结果如下：

➢ 分层抽样最优样本量

传统抽样理论已经发展较为完善，从理论上保证用抽取的部分样本代替全部样本数据进行建模是把抽样误差控制在一定范围内，在牺牲较小的精度的同时换取较高的计算效率。大样本带来高精度，但同时也损失了计算效率，因此确定了一些选取最优样本的标准。换句话说，停止增加样本量的条件，在分层抽样的最优样本的确定的条件上与简单抽样相同。

在这样增加样本量基础之上，设定每次增加样本数量为 10 个，起始为 100 个样本。使用的函数仍为 fun7，只是这次的 lab 的函数值为 fun3，即分层抽样的抽样函数，为了避免抽样的随机性，设置了 for 循环，使得在每个样本数量下都抽取 5 次，取 5 次抽样的平均数，避免抽样随机性导致的品样本质量计算偏差。

分层抽样寻找最优样本质量的实验结果为;

```
1.  > fun7(lab=fun4)
2.  [1] 0.9510933
3.  [1] 340
```

基于贷款数据集 loanstatC 数据集的双指标样本相似度的分层抽样最优样本量为：340 个，且能达到的样本质量为 95.10%，在这种取样数量上既可以保证样本抽样质量足够高，大于 95%，且再向其中增加样本也无法显著增加样本质量，故最优样本质量为 340 个。

➢ 给定样本容量计算样本质量

利用 3.2 节中生成的 samp 向量储存了抽样样本的数量序列，因此可以利用 samp 向量作为相应的样本容量，利用 grade 进行分层，并结合 loan_amnt 和

annual_inc 两个指标计算 samp 中样本容量的样本质量，最后根据样本容量个数绘制出随样本数量变化的样本质量分布散点图，所使用的 R 代码如下所示：

```
######实证检验
mb1<-apply(samp,1,function(x) fun4(x,"annual_inc"))
J1<-apply(mb1,2,function(x) fun2(x,bre1,PD1))
mb2<-apply(samp,1,function(x) fun4(x,"loan_amnt"))
J2<-apply(mb2,2,function(x) fun2(x,bre2,PD2))
Q2<-exp(-(J1+J2)/2)
huatu1<-as.data.frame(cbind(as.vector(samp),as.vector(Qj)))
names(huatu1)[1:2]<-c("samp","Q2")
ggplot(huatu1,aes(x=samp,y=Qj))+geom_point(size=2,color="red")+ggtitle('分层抽样样本质量与样本容量散点图') +theme(plot.title = element_text(hjust = 0.5,family="myFont",size=18,color="black"),panel.background=element_rect(fill='aliceblue',color='black'))
```

首先需要生成 mb1 和 mb2 两个抽样样本数据矩阵，利用的皆为 apply 行处理函数，利用 fun4 生成两个指标的抽样样本矩阵，然后利用 apply 列处理对两个抽样矩阵的 KL 信息量进行计算，最后将 J1 和 J2 进行合并，并求解出基于两个指标的样本质量 Q2。但需要注意的是，由于分层抽样的运算代价太高再次就不进行多次取样消除抽样误差。并利用 ggplot2 作图包，绘制函数图形，实验结果如下图:

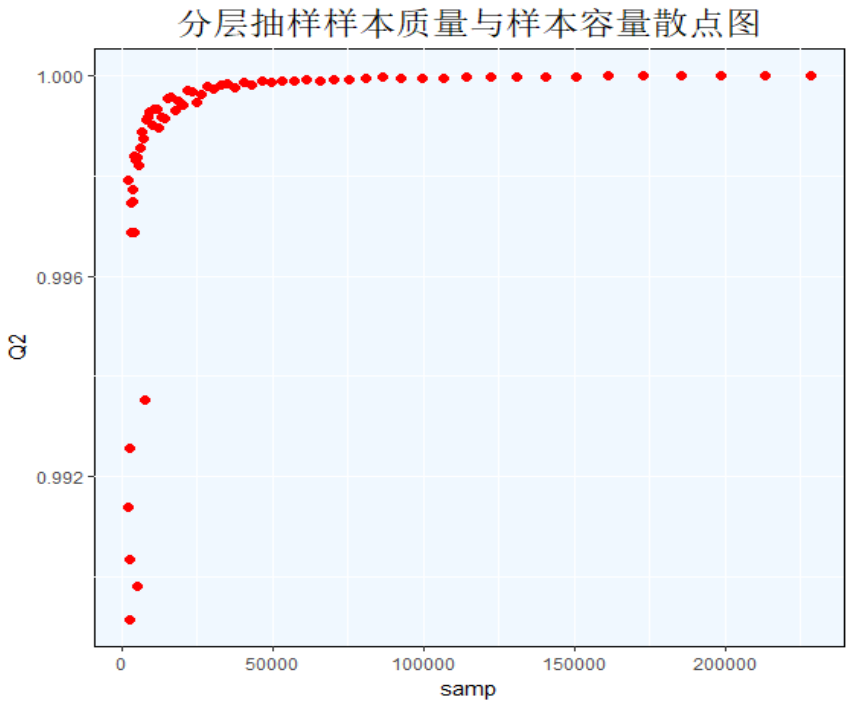

图 4-1 分层抽样样本质量与容量关系

在图中可以清楚地看到，随着分层抽样样本容量的增加，样本质量也在迅速爬升，且爬升速度较简单随机抽样来说要快很多，大约在分层抽样的样本数量增加到 13000 左右时，样本质量增加量的改变较少，并且样本质量逼近于 1。由于分层抽样利用的信息较多，达到相同样本质量时所需要的样本容量较小。

➢ 给定样本质量阈值计算样本容量

计算给定样本质量阈值为 95%时的分层抽样的样本容量，构造函数 fun3 在逐步增加样本量来逼近 95%的阈值，起始的样本容量为 100 个，每次添加的步长为 5，样本质量容许的误差为 0.0005，即确定到 95%±0.5%即可停止增加样本量。对于 fun3 函数中的参变量，Pro1 为设定的阈值，lab 为所使用的抽样函数，其中，fun1 为简单抽样方式，fun4 为分层抽样，fun6 为两阶段抽样。而且为了避免抽样的随机性对实验结果造成影响，内层设置多次抽样取平均的方法来减少抽样随机性。R 语言代码与 3.4 相同。

分层抽样的阈值找样本容量的实验结果为:

```
> fun3(0.95,fun4)      #fun4 为分层抽样
[1] 0.9457055
[1] 285
```

基于贷款数据集 loanstatC 数据集的双指标样本相似度的分层抽样最优样本量为：310 个，且能达到的样本质量为 94.57%，在这种取样数量使得样本质量已经达到 95%，故简单随机抽样的数量为 285 个。

也可以利用插值函数，如下代码，分别计算 95%，96%,97%,99%的样本质量的样本容量得出下表。

```
#选择样本拟合样本量和样本质量变化曲线图
x=seq(6,20,by=0.1)
y=2^(x)
samp=round(y)[-c(1:20)]
samp=samp[-c(40:length(samp))]#选取合适数据
sa=data.frame(y=samp,x=c(1:39))
n1=length(samp)
samp=as.array(samp)
ma<-apply(samp,1,fun1)#利用随机抽样函数抽取一组样本
q1=apply(ma,2,fun2)#计算样本质量
s1=smooth.spline(q1,samp)#拟合曲线并预测样本质量在 0.95 的情况下的样本量
pr1=predict(s1,0.95)  #预测 95%的样本容量
ceiling(pr1[[2]])
 pr2=predict(s1,0.96)  #预测 96%的样本容量
```

```
ceiling(pr2[[2]])
pr3=predict(s1,0.97)  #预测 97%的样本容量
ceiling(pr3[[2]])
pr4=predict(s1,0.98)  #预测 99%的样本容量
ceiling(pr4[[2]])
```

表 3-1 不同精度下的最优样本容量表

样本质量	95%	96%	97%	99%
样本容量	295	468	816	1575

五、 整群抽样实证研究

5.1 整群抽样数据准备

首先在生成的数据矩阵 mydata 中抽取出三个变量分别是 loan_amnt,annual_inc，issue_d，利用 dplyr 包中的 select 函数形成新的数据矩阵 data4 用于进行整群抽样实验。其次，利用 issue_d 贷款时间变量进行数据的分群。

由于数据表中只有 12 个月的记录，无法实施整群抽样，因此自己构造分群变量，也就是将 12 个月份的按每个月中的记录拆分成 1000 个一组的群，该月份最后一个无法形成 1000 个数量的群也进行保留，使数据具有完整性。拆分后的变量命名为 riqi，为新的分类变量，标识整个数据，并作为新的日期合并到表中。形成最终数据集 zqdata。

```
##整群随机抽样
data4<-mydata[,c('loan_amnt','annual_inc','issue_d')]     ##整群抽样数据集
zqdata<-data4[order(data4[,3]),]     #按日期顺序进行排序
xl<-vector(mode="character",length=0) #构造零向量按日期顺序人为切割成多个分组
YF<-as.vector(table(zqdata$issue_d)[-1])  #统计在每个月份上的样本数量
m=0
label=c('Apr','Aug','Dec','Feb','Jan','Jul','Jun','Mar','May','Nov','Oct','Sep')
for(j in 1:length(label) ){
  a=0
  for(i in 1:YF[j]){
    if(i%%1000==1){
      a=a+1
      x=sprintf("%s-%d",label[j],a)
    }
    m=m+1
    xl[m]<-x
  }
}                ##认为拆分每个月份，构造新的标签列
zqdata<-cbind(zqdata,xl)
names(zqdata)[4]<-c("riqi")    ##命名新列名日期
```

5.2 确定整群抽样样本数量

按照新生成日期变量 riqi 划分为表示不同日期的群。根据简单随机抽样和分层抽样的样本总量最高值，设定在整群抽样中最高值的样本数量为 228000 个，即最大的一次采样最多为 228 个群。因此为了是样本量分布均匀，采集的群数量从 1 开始到 228，采取步长为 4 的方式采取群数，以此来构造整群样本抽样数量集合。

```
#按日期先后顺序划分成 200 个群
summary(zqdata$riqi)
xulie=seq(from=1, to=228, by=4)    ##构造整群抽样样本数 1-228 个群
sam_count<-vector(mode="numeric",length=0)
n=0
```

R 语言的实现如上述代码所示，整群抽样数就保存在 xulie 中。以此抽样样本序列来进行接下来的整群抽样。

5.3 定义整群抽样函数与信息量计算函数

整群抽样的样本 KL 信息量函数 fun2 与简单随机抽样的函数相同，接下来定义整群抽样样本抽取函数 fun5

```
fun5<-function(s,datasam){
  sub1<-
cluster(data=zqdata,clustername="riqi",size=s,method="srswor",description=FA
LSE)
  if(datasam=="annual_inc")  p<-data4[sub1$ID_unit,]$annual_inc
  if(datasam=="loan_amnt")  p<-data4[sub1$ID_unit,]$loan_amnt
  res<-c(p,matrix(NA,1,228000-length(p)))
  return(res)
}
```

在 fun5 中，首先利用 sampling 包中的 cluster 整群抽样函数根据新划分的日期指标进行“srswor”整群抽样，每层的抽样数量根据前一节定义的储存收养群数的向量 xulie 进行确定，datasam 为字符型变量，用于标识指标的不同。p 为在整群抽样中，由 data4 标识的数据的抽样样本集合。函数返回值为 res，也就是生成抽样函数矩阵。

5.4 整群抽样实证检验

做好具体的数据和函数的准备工作后，对整群抽样函数的最优样本量，给定样本容量计算样本质量和给定样本容量阈值计算所需的样本容量三个方面进行讨论，结果如下：

➢ 整群抽样最优样本量

由于整群抽样的抽样方式的特殊性，只能一个一个增加群，所以无法通过 fun7 来进行确认，可以通过样本容量与样本质量散点图，来确定整群抽样的最有样本量，从图 5-1 中可以看出抽取 1 个群即可达到最优样本质量 95%以上，且再多增加群也无法显著地增加样本质量。

➢ 给定样本容量计算样本质量

利用 5.3 节中生成的 xulie 向量储存了抽样样本的整群数量序列，因此可以利用 xulie 向量作为相应的样本容量，利用新生成的变量 riqi 进行分层，并结合 loan_amnt 和 annual_inc 两个指标计算 xulie 整群中样本容量的样本质量，得到两个样本信息量 J1 和 J2，然后按权重 1:1 来综合信息量，最后根据抽群的整群个数绘制出随样本数量变化的样本质量分布散点图，所使用的 R 代码如下所示：

```
######实证检验
Q3<-matrix(NA,57,2)
xulie<-as.matrix(xulie)
for(i in 1:2){
  mc1<-apply(xulie,1,function(x) fun5(x,"annual_inc"))
  J1<-apply(mc1,2,function(x) fun2(x,bre1,PD1)) #annual_inc 样本数据的信息量
  mc2<-apply(xulie,1,function(x) fun5(x,"loan_amnt"))
  J2<-apply(mc2,2,function(x) fun2(x,bre2,PD2))  #loan_amnt 样本数据的信息量
  Q3[,i]<-exp(-(J1+J2)/2)          ##权重皆为 1
}
Qz<-apply(Q3,1,mean)          #计算每个样本容量下的样本质量均值、
huatu2<-as.data.frame(cbind(as.vector(samp),as.vector(Qz)))
names(huatu2)[1:2]<-c("samp","Qz")
ggplot(huatu2,aes(x=samp,y=Qz))+geom_point(size=2,color="red")+ggtitle('整群抽样样本质量与样本容量散点图') +theme(plot.title = element_text(hjust = 0.5, family="myFont",size=18,color="black"),panel.background=element_rect(fill='aliceblue',color='black'))
```

首先需要生成 mc1 和 mc2 两个抽样样本数据矩阵，利用的皆为 apply 行处理函数，利用 fun5 生成两个指标的关于整群抽样样本矩阵，然后利用 apply 列处理对两个抽样矩阵的 KL 信息量进行计算，最后将 J1 和 J2 进行合并，并求解出基于两个指标的样本质量 Q2。但需要注意的是，为了消除随机抽样所带来的抽样误差，保证整群抽样的抽样水平的客观性，在整群抽样中使用了多次抽样取平均的样本质量计算方法。利用 ggplot2 作图包，绘制函数图形，实验结果如下图:

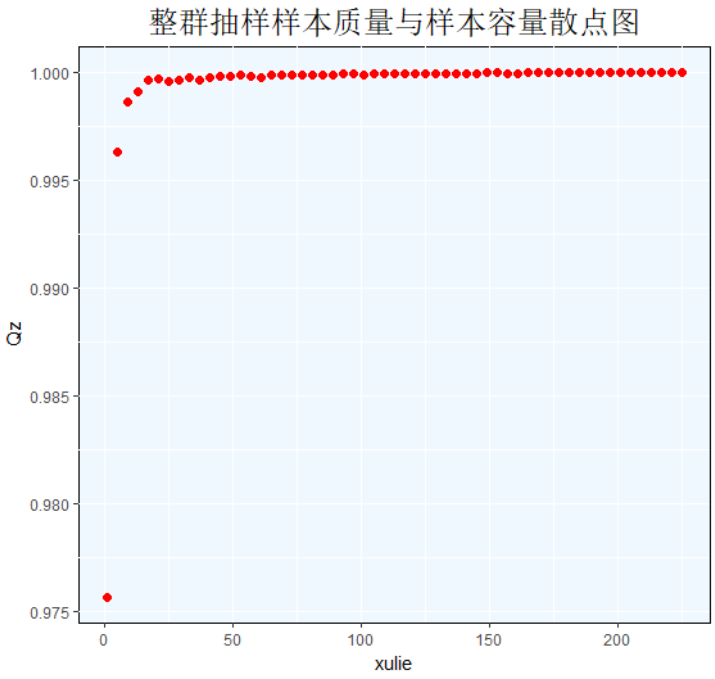

图 5-1 整群抽样样本质量与容量关系

在图中可以清楚地看到，随着整群抽样样本容量的增加，即样本个数增加（因为每个整群大约都含有相同的 1000 个样本在其中，因此可以确定的是图中的整群数量也就是相应的样本数量），样本质量也在迅速爬升，且爬升速度较简单随机抽样和分层抽样来说要快很多，大约在整群抽样的样本数量增加到 4000 左右时，样本质量增加量的改变较少，并且样本质量逼近于 1。这说明整群抽样划分的每一个群对整体的代表性极高，也就是说按月份时间的整群划分就很好的综合了整体数据的信息。

➢ 给定样本质量阈值计算样本容量

由于整群抽样的抽样方式较为特别，只能一次抽取至少一个群，而通过 riqi 所划分的群，至少含有 1000 个样本数量，而 1000 个群在实验的散点图上可以看出，1 个群就已经完全大于 95%。因此无法进行给定 95%的阈值的样本质量计算（抽取的样本数），也可以说 1 个群已让整体的样本质量达到了阈值。

六、 两阶段抽样实证研究

6.1 两阶段抽样数据准备

由于两阶段抽样相当于分层抽样和整群抽样的结合，所以首先在生成的数据矩阵 mydata 中抽取出四个变量分别是 loan_amnt,annual_inc，issue_d，grade 利用 dplyr 包中的 select 函数形成新的数据矩阵 data5 用于进行两阶段抽样实验。其次，利用在整群抽样中已经生成的 riqi 变量作为抽取的整群。

由于数据表中 riqi 变量中的整群含有较少的样本量，无法满足实施两阶段抽样的条件，因此自己删除每个月份中所包含的不满足 1000 个变量的整群，使得拥有相同 riqi 的变量的个数皆为 1000 个，因为只删除了很少的部分变量所以不影响结果，形成最终数据集 ljddata。R 代码如下：

```
##########两阶段抽样
label1=c('Apr-20','Dec-11','Feb-16','Jan-16','Aug-19 ','Jul-30','Jun-18','Mar-17','May-20','Nov-26','Sep-11','Oct-39')
xk<-vector(mode="numeric",length=0)
n=0
for(i in 1:length(xl)){
  if(xl[i] %in% label1){
    n=n+1
    xk[n]=i
  }
}            ##删除部分数据，这些数据不满足每个群含足够的的样本数量
data5<-mydata[,c('loan_amnt','annual_inc','issue_d','grade')]    ##多阶段抽样数据集
ljddata<-cbind(data5[order(data5[,3]),],xl)      #按日期顺序进行排序并构造两阶段数据集
ljddata<-ljddata[-xk,]
samp<-as.matrix(samp)
```

6.2 确定两阶段抽样样本数量

两阶段抽样由于是第一阶段是整群抽样分层抽样，因此所需要的样本量的确定与分层随机抽样的相同，共同使用 samp 中的 68 个样本容量。利用这 68 个容量进行数据的第二阶段的分层抽取与检验。而第一阶段所抽取的整群数量则是按照第二节阶段所需抽取数量的两倍以上来抽取整群，比如，某次两阶段需要抽取的样本数量为 3120 个，那么第一阶段所要抽取的个数为 3120*2=6240，因为每

一个群包含了 1000 个样本，因此确定第一阶段所抽取的整群为([6240/1000]+1)个群。如此确定两个阶段的抽样方法。

6.3 定义两阶段抽样函数与信息量计算函数

两阶段抽样的样本 KL 信息量函数 fun2 与简单随机抽样的函数相同，接下来定义两阶段抽样样本抽取函数 fun6

```
15. fun6<-function(s,datasam){
16.   qs<-round((s%/%1000)*2)   #计算关于 sampling 两倍的抽样数量
17.   mzgs<-s%/%qs              #计算两阶段第一阶段所需要的抽取的整群数
18.   yushu<-s-(qs-1)*mzgs      ##取余
19.   zngs<-c(rep(mzgs,qs-1),yushu)
20.   sub2<-
    mstage(data=ljddata,stage = c("cluster","stratified"),varnames = list("xl" ,
    "grade"),size = list(qs,zngs),method = c("srswor","srswor"),description = FA
    LSE)                       #利用 sampling 包进行两阶段抽样，并将结果储存
21.   if(datasam=="annual_inc")  p<-data5[sub2$`2`$ID_unit,]$annual_inc
22.   if(datasam=="loan_amnt")  p<-data5[sub2$`2`$ID_unit,]$loan_amnt
23.   res<-c(p,matrix(NA,1,230000-length(p)))  #储存抽样数据
24.   return(res)
25. }
```

在 fun6 中，首先确定两阶段所分别需要抽取的样本数量，第二阶段是根据 samp 的样本数量来确定，第一阶段则是根据第二阶段的数量来确定在 riqi 群中抽取的整群个数。其次利用 sampling 包中的 mstage 两阶段抽样函数根据 riqi 首先进行方法为“srswor”的整群抽样，然后根据 grade 指标进行“srswor”分层抽样，每层的抽样数量根据总体样本分布 PD3 进行确定，datasam 为字符型变量，标识指标的不同。p 为在两阶段抽样中，由 data6 标识的数据的抽样样本集合。函数返回值为 res，也就是生成抽样函数矩阵。

6.4 两阶段抽样实证检验

做好具体的数据和函数的准备工作后，对两阶段抽样函数的最优样本量，给定样本容量计算样本质量和给定样本容量阈值计算所需的样本容量三方面进行讨论，结果如下：

➢ 两阶段抽样最优样本量

从理论上保证用抽取的部分样本代替全部样本数据进行建模是把抽样误差控制在一定范围内，在牺牲较小的精度的同时换取较高的计算效率。大样本带来高精度，但同时也损失了计算效率，因此确定了一些选取最优样本的标准。换句话说，停止增加样本量的条件，在两阶段抽样的最优样本的确定的条件上与简单抽样相同。

两阶段抽样最优样本实验结果如下：

```
> fun7(lab=fun6)
[1] 0.96157746
[1] 310
```

基于贷款数据集 loanstatC 数据集的双指标样本相似度的两阶段抽样最优样本量为：310 个，且能达到的样本质量为 96.15%，在这种取样数量上既可以保证样本抽样质量足够高，大于 95%，且再向其中增加样本也无法显著增加样本质量，故最优样本质量为 310 个。

➢ 给定样本容量计算样本质量

利用 2.3 节中生成的 samp 向量储存了抽样样本的数量序列，因此可以利用 samp 向量作为相应的样本容量，利用新生成的变量 riqi 进行分群，并结合 loan_amnt 和 annual_inc 两个指标计算 samp 中相应样本容量的样本质量，得到两个样本信息量 J1 和 J2，然后按权重 1:1 来综合信息量，最后根据两阶段抽样的抽样个数绘制出随样本数量变化的样本质量分布散点图，所使用的 R 代码如下所示：

```
########实证检验
Q4<-matrix(NA,length(samp),2)
samp<-samp[1:50,]                    #由于内存不足样本容量过大因此只取前 50 个
samp<-as.matrix(samp)
for(i in 1:2){
  md1<-apply(samp,1,function(x) fun6(x,"annual_inc"))
  J1<-apply(md1,2,function(x) fun2(x,bre1,PD1)) #annual_inc 样本数据的信息量
  md2<-apply(samp,1,function(x) fun6(x,"loan_amnt"))
  J2<-apply(md2,2,function(x) fun2(x,bre2,PD2))  #loan_amnt 样本数据的信息量
  Q4[,i]<-exp(-(J1+J2)/2)            ##权重皆为 1
}
Ql<-apply(Q4,1,mean)           #计算每个样本容量下的样本质量均值
huatu3<-as.data.frame(cbind(as.vector(samp),as.vector(Ql)))
```

```
14. names(huatu3)[1:2]<-c("samp","Ql")
15. ggplot(huatu3,aes(x=samp,y=Ql))+geom_point(size=2,color="red")+ggtitle('两阶段抽样样本质量与样本容量散点图') +theme(plot.title = element_text(hjust = 0.5, family="myFont",size=18,color="black"),panel.background=element_rect(fill='aliceblue',color='black'))
```

首先需要生成 md1 和 md2 两个抽样样本数据矩阵，利用的皆为 apply 行处理函数，利用 fun6 生成两个指标的关于两阶段抽样样本矩阵，然后利用 apply 列处理对两个抽样矩阵的 KL 信息量进行计算，最后将 J1 和 J2 进行合并，并求解出基于两个指标的样本质量 Q2。但需要注意的是，为了消除随机抽样所带来的抽样误差，保证两阶段抽样的抽样水平的客观性，在两阶段抽样中使用了多次抽样取平均的样本质量计算方法。利用 ggplot2 作图包，绘制函数图形，实验结果如下图:

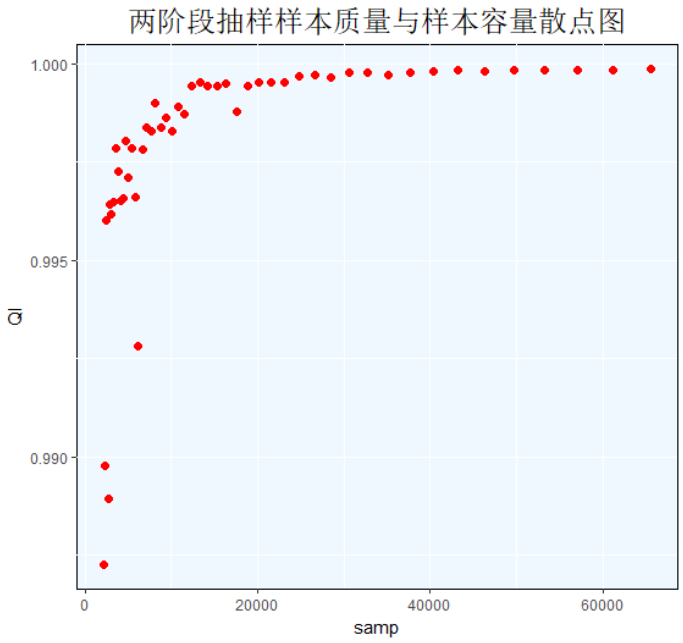

图 6-1 两阶段抽样样本质量与容量关系

在图中可以清楚地看到，随着两阶段抽样样本容量的增加，即样本个数增加，样本质量也在迅速爬升，且爬升速度较简单随机抽样，分层抽样来说要快很多，大约在两阶段随机抽样的样本数量增加到 10000 左右时，样本质量增加量的改变较少，并且样本质量逼近于 1。这说明两阶段抽样结合的信息更多，通过第一阶段的抽样抽出对整体代表性极强的整群，又结合分层抽样抽取出与总体特征更为接近的样本，很好的综合了整体数据的信息。

➢ 给定样本质量阈值计算样本容量

计算给定样本质量阈值为 95%时的两阶段抽样的样本容量，构造函数 fun3 在逐步增加样本量来逼近 95%的阈值，起始的样本容量为 100 个，每次添加的步长为 5，样本质量容许的误差为 0.0005，即确定到 95%±0.5%即可停止增加样本量。对于 fun3 函数中的参变量，Pro1 为设定的阈值，lab 为所使用的抽样函数，其中，fun1 为简单抽样方式，fun4 为分层抽样，fun6 为两阶段抽样。而且为了避免抽样的随机性对实验结果造成影响，内层设置多次抽样取平均的方法来减少抽样随机性。R 语言代码与 3.4 相同。

分层抽样的阈值找样本容量的实验结果为:

```
10. > fun3(0.95,fun6)      #fun 为两阶段抽样
11. [1] 0.9487055
12. [1] 275
```

基于贷款数据集 loanstatC 数据集的双指标样本相似度的两阶段抽样最优样本量为：310 个，且能达到的样本质量为 94.87%，在这种取样数量使得样本质量已经达到接近 95%，故简单随机抽样的数量为 275 个。

七、 大数据抽样实验总结

本次实验进行了关于四种抽样方法（简单随机抽样，分层抽样，整群抽样，两阶段抽样）基于贷款数据 loanstatC 的抽样数量的样本质量测试，所得到的结果可以整理成下表：

抽样方法	最优样本量和相应样本质量	样本质量 95%时样本量
简单随机抽样	360/96.17%	310
分层抽样	340/95.10%	285
整群抽样	1000/97.54%	1000
两阶段抽样	310/96.15	275

由表中可以总结出以下结论：

1. 简单随机抽样：利用的总体信息最少，完全的在总体中随机抽样，得到的样本质量随样本容量的变化中也可以看到，是样本质量增长最缓慢的，需要很大的样本才能更为贴近总体，大约 25000 时样本质量接近 1。但是此抽样的实施简单，便于操作。

2. 分层抽样：利用了 grade 分类变量的信息，在已知 grade 总体分布频率的基础上在每个信用等级上进行抽样，样本更贴近于总体，因此样本质量随样本容量的变化中，样本质量增加的速度比简单随机抽样速度逼近 1 的快。大约 13000 时样本质量接近 1。但是此抽样的实施相对与简单抽样较为麻烦，不太便于操作，成本较高。

3. 整群抽样：利用了总体样本日期的信息，利用日期进分群，通过每一个群来反映总体，抽取的每一个群都一定程度的反映了总体，样本更贴近于总体，因此样本质量随样本容量的变化中，样本质量增加的速度比简单随机抽样和分层抽样速度逼近 1 的快。大约 4000 时样本质量接近 1。此抽样的实施相对与简单抽样较为麻烦，不太便于操作，但比分层抽样的操作简单。

4. 两阶段抽样：是四种抽样方法中操作最为复杂的，第一阶段结合了整群抽样，第二阶段综合了分层抽样，是综合总体信息最多的抽样方式，但基于贷款数据 loanstatC 的实验的中结果并不如整群抽样，样本质量随样本容量的变化中，样本质量增加的速度比简单随机抽样和分层抽样速度逼近 1 的快。大约 10000 时样本质量接近 1。但是抽样的实施成本较高。