# 2017－2018 年第一学期

# 《大数据统计基础》试题

**说明：**

1. **试题共含五道大题，总计 100 分。请仔细阅读题目，按照要求完成，并在指定时间内提交。**
2. **所有分析应填写在答题纸内的相应位置上。**
3. **在分析正文中，要求用统计图、统计表、截图等展示分析结果，并通过文字加以阐述；整个分析过程中的程序语句请加上注释后附后。答卷以 word 方式提交（除了第三大题有特殊的要求）。**

一、抽样

在 loan data 的 2013-2014 年数据集中，对变量 loan_amnt，使用简单随机抽样分别抽取样本容量为 100、1000、5000、10000 的样本，给出相应的样本质量，并且计算最优样本容量。（20 分）

二、数据预处理

1、（10 分）UCI 鲍鱼数据含有 4177 只鲍鱼的观测。其中，被解释变量是 Rings（环的个数），鲍鱼年龄是环的个数加 1.5。其他变量为预测变量。以下是获取数据和数据信息的方式：

> library(AppliedPredictiveModeling)

> data(abalone)

> head(abalone)

> ?abalone

1) 对数据作图估计预测变量和被解释变量之间的函数关系。
2) 用散点图和相关系数图解释预测变量之间的相关性。
3) 对预测变量估计重要性得分。找到一种筛选方法得到预测变量子集，该集合不含冗余变量。
4) 对连续型预测变量应用主成分分析，决定多少个不相关的主成分能够代

表数据中的信息？

2、（10 分）使用下面的非线性函数来模拟数据$(y,x_1,x_2,x_3,x_4,x_5)$：

$$y=10\sin(\rho x_1x_2)+20(x_3-0.5)^2+10x_4+5x_5+e$$

其中预测变量$x_1$到$x_5$服从均匀分布，误差 e 服从标准正态分布

1) 写一个 R 函数从该模型中模拟数据。
2) 随机模拟一个数据集，样本量是 500，绘制图形研究预测变量和被解释变量之间的关系。
3) 使用线性回归中的向前法、向后法和逐步回归等变量选择方法，最终模型选择了哪些变量？
4) 应用不同的过滤法，逐个评估变量。一些过滤法同时评估多个变量（如 ReliefF 算法），两个有交互效应的预测变量$x_1$和$x_2$否被选中了？是否倾向于选择其中某一个变量？

参考书籍：

[1] Max Kuhn, Kjell Johnson. Applied Predictive Modeling. Springer,2013.

（题目所涉及的方法参考本书第 18 章、第 19 章）

三、数据探索性分析

为了评估英超球员的价值，研究人员希望构建一个预测模型，通过球员在当前赛季的场上表现来预测其在下一赛季的进球数。已经搜集到了英超 16 支球队 166 名球员在 2012 到 2013 赛季的场上表现情况、下一赛季的进球数。请你帮助研究人员开展如下数据分析工作：

1、研究人员发现，用于描述当前赛季场上表现的变量众多，若直接用它们对下一赛季进球数进行回归建模，建模效果并不好。应该如何解决这一问题？请为研究人员提供一个可行的解决方案，简要说明你的思路。（5 分）

2、按照你的想法，使用“英超数据.csv”中提供的数据，通过 R 语言编程来解决前述问题，并对结果进行适当分析。（15 分）

要求：使用 R Markdown 记录你为研究人员提供的解决思路（文字简述）、R code、分析结果与相关解读。输出 html 文件作为附件来提交，用“学校＋姓名”

的形式来命名 html 文件。

四、数据可视化

1、在 loan data 中，自选合适的变量，绘制以下图形：

1) 分面的风玫瑰图，玫瑰叶片的颜色至少四种；（2 分）

2) 某一个连续型变量的分布直方图，并加入拟合分布线，直方图的组距和组数自己设定（不要使用默认的），并且每个柱子里面填上相应的组的频数，整个图片加上一个黑色的外框，并且图的底色为浅色，柱子为深色，在密度最高的部分加上文字标注“此处密度最大”；（3 分）

3) 某两个连续型变量的密度图，并且在图中找出一个部分加上一个方框与其他部分区别开来（比如说，密度最高或者最低的区域，用一个矩形的方框加以标示）；（3 分）

4) 挑选多个连续型变量，进行聚类，并且绘制相应的热图，并进行美化（可以不用全部样本）。（2 分）

2、使用 province 数据中合适的变量，绘制两幅不同的图，进行空间数据的展示。（10 分）

要求：变量选择方面尽量使整个分析具有一定的逻辑性。以上所有图表都要求有图表名称、图例、行标题、列标题等要素。

五、空间统计

1、空间自相关原理及应用领域（10 分）

2、时空扫描统计原理及应用（10 分）

# 2017－2018 年第一学期

# 《大数据统计基础》试题　答题纸

学校__中央财经大学__学号_2017210761__姓名___王思雨____成绩_____

## 一、抽样

### 1. 数据来源及变量说明

（1）**数据来源：**表格 LoanStats3c 数据集提供了 2013 年到 2014 年成功在 Lending Club 上成功申请到贷款的人的信息。

（2）**数据变量说明：**LoanStats3c 表格中包含的信息变量有贷款编号、贷款人标号、贷款人申请的贷款数量、投资者提供的贷款数量、贷款人的基本信息（年龄、房屋的所有权、贷款量、借款的目的、借款人的信用评级）、贷款人的还款情况（每月还款额、是否还清、上一次还款发生在什么时间以及还款的数额等）、贷款人的信用记录（开过多少张信用卡、首次开设信用卡的时间、以及信用卡还款延期次数）、借款得利息、最终偿付得总金额等。变量说明表如下表：

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<br>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<br>贷款数量越大说明贷款额度高，贷款分级高。<br>单位为：美元 |

### 2. 数据变量的预处理

本次抽样实验主要对贷款数据中的贷款量（loan_amnt）这一变量进行分析，首先利用 summary 函数查看贷款量的基本统计量信息，汇总可以得到表 1-1。

表 1-1 loan_amnt 变量基本统计量信息表

| 统计量 | 最小值 | 上四分位数 | 中位数 | 均值 | 下四分位数 | 最大值 | NA |
|---|---|---|---|---|---|---|---|
| 取值 | 1000 | 8325 | 13000 | 14870 | 20000 | 35000 | 2 |

从上述基本统计量信息表中可以看出，贷款量的取值范围为 1000-35000 元，跨度较大，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。利用 plot 画出贷款量 loan_amnt 的频数分布的直方图如图 1-1。

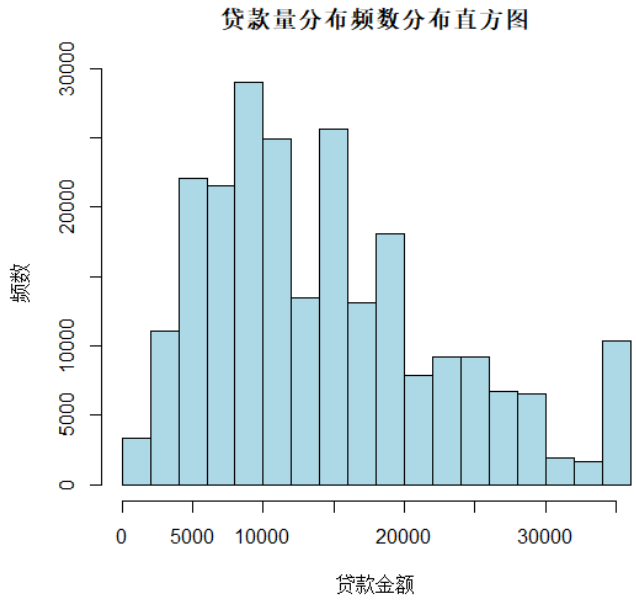


图 1-1 贷款量分布频数分布直方图

从图 1-1 直方图可以看出，贷款量 loan_amnt 大部分的都分布在 5000-20000 美元之间，且成右偏分布，但大体上分布均匀，结合下四分位数为 20000 可以看出，样本中的贷款金额大都分布在 20000 美元以下。

根据上述对贷款量的分布情况的分析，若直接对所有样本进行简单随机抽样，会导致抽样结果出现偏差。其次，loan_amnt 为数据集样本相似性的度量变量，是连续型变量，应当进行适当的分组离散化处理

由表 1-1 可看出，贷款量变量存在 2 个缺失值，首先删去缺失值，计算无缺失的总样本数量 n 为 235629。根据贷款量的四分位数等信息并结合统计图，选取 17 个分段点，由此将无缺失的变量数据分成 18 组，将贷款量 loan_amnt 这一连续型变量处理为分段取值的离散变量，得到贷款量的分组频数表，如表 1-2。

表 1-2 loan_amnt 变量分组频数表

| 组别 | 分组 | 频数 | 频率 |
|---|---|---|---|
| 1 | (0,2e+03) | 3332 | 0.141409 |
| 2 | (2e+03,4e+03) | 11035 | 0.046831 |
| 3 | (4e+03,6e+03) | 22100 | 0.093791 |
| 4 | (6e+03,8e+03) | 21556 | 0.091482 |
| 5 | (8e+03,1e+04) | 28975 | 0.122967 |
| 6 | (1e+04,1.2e+04) | 24875 | 0.105567 |
| 7 | (1.2e+04,1.4e+04) | 13460 | 0.057123 |
| 8 | (1.4e+04,1.6e+04) | 25662 | 0.108907 |
| 9 | (1.6e+04,1.8e+04) | 13144 | 0.055782 |
| 10 | (1.8e+04,2e+04) | 18036 | 0.076543 |
| 11 | (2e+04,2.2e+04) | 7905 | 0.033548 |
| 12 | (2.2e+04,2.4e+04) | 9155 | 0.038853 |
| 13 | (2.4e+04,2.6e+04) | 9178 | 0.038950 |
| 14 | (2.6e+04,2.8e+04) | 6698 | 0.028425 |
| 15 | (2.8e+04,3e+04) | 6535 | 0.027734 |
| 16 | (3e+04,3.2e+04) | 1956 | 0.008301 |
| 17 | (3.2e+04,3.4e+04) | 1684 | 0.007146 |
| 18 | (3.4e+04,3.6e+04) | 10343 | 0.043894 |

画出分组后的贷款量 loan_amnt 频数条形图如图 1-2，由于是通过平均间隔分组，所以分组之后的贷款量但存在一定程度的右偏，但整体较为正态。所以分组数量不均匀，单纯利用利用直方图的均匀分组情况并不理想。尝试根据不均匀间隔分组。

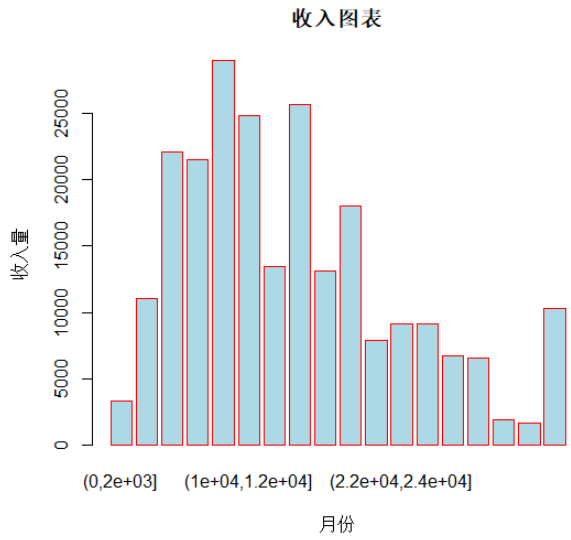


图 1-2 loan_amnt 频数分布条形图

仍然选取选取 17 个不均匀的分段点，由此将无缺失的变量数据分成 18 组，将贷款量 loan_amnt 这一连续型变量处理为分段取值的离散变量，得到贷款量的分组频数表：

表 1-3 loan_amnt 变量分组频数表

| 组别 | 分组 | 频数 | 频率 |
|---|---|---|---|
| 1 | (0,1200) | 3332 | 0.003637 |
| 2 | (1200,2400) | 11035 | 0.014484 |
| 3 | (2400,3600) | 22100 | 0.028786 |
| 4 | (3600,4800) | 21556 | 0.029745 |
| 5 | (4800,6000) | 28975 | 0.078109 |
| 6 | (6000,7200) | 24875 | 0.044926 |
| 7 | (7200,8400) | 13460 | 0.055909 |
| 8 | (8400,9600) | 25662 | 0.039107 |
| 9 | (9600,10800) | 13144 | 0.094898 |
| 10 | (10800,14000) | 18036 | 0.142298 |
| 11 | (14000,18000) | 7905 | 0.164689 |
| 12 | (18000,23000) | 9155 | 0.120595 |
| 13 | (23000,26400) | 9178 | 0.069630 |
| 14 | (26400,28800) | 6698 | 0.029631 |
| 15 | (28800,33600) | 6535 | 0.027415 |
| 16 | (3e+04,31200) | 1956 | 0.011029 |
| 17 | (31200,34500) | 1684 | 0.002054 |
| 18 | (34500,35000) | 10343 | 0.043041 |

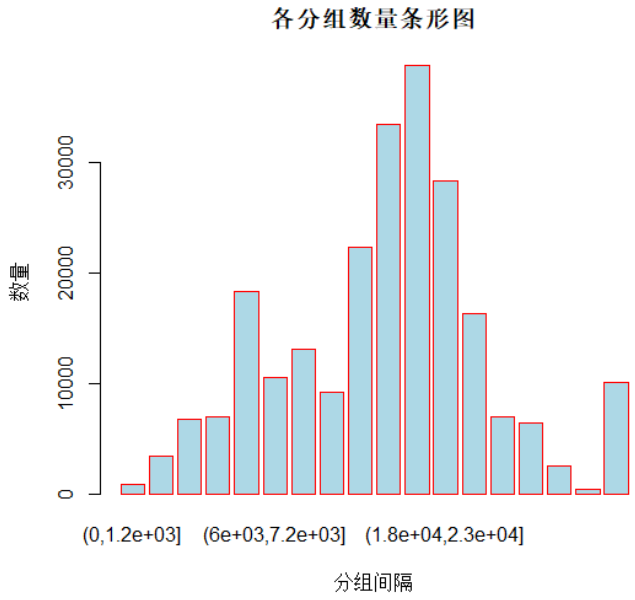


图 1-3 loan_amnt 重新分组频数分布条形图

可以发现，人工分组后，重新得到的分组频数分布的贷款量较之原始数据的均匀分组更接近正态型，但稍微存在一点程度的右偏。对比之后发现，重新分组之后的贷款量数据更适合进行抽样研究，因此接下来对分组之后的数据进行处理。

### 3. 确定抽样实验的样本容量

样本容量指样本数据中包含的从总体抽取的观测值个数。平均意义而言，样本容量越大，其包含的整体数据信息就越多，样本和整体的相似度也就更为接近，样本质量也越高，当样本容量和整体数据观测个数相等时，样本与总体一致，样本质量达到最高。

但是，根据实证研究，随着样本容量的增长，样本质量的提高并不是线性的。因此在数据分析过程中，考虑到样本的收集成本以及计算效率，不能无限的增加样本数量，故需要确定达到一定样本质量要求的最小样本容量，即最优样本容量。

可以注意到，样本量越多，样本质量越接近 1。首先需要生成一系列的样本容量，由于贷款量的样本容量为 235629，要产生这么大的数，因此考虑使用指数分布在较小的区间内生成一些列跨度较大的样本容量供选择，再从中抽取一系列在样本量范围内的样本量。生成的指数分布如图 1-4 所示：

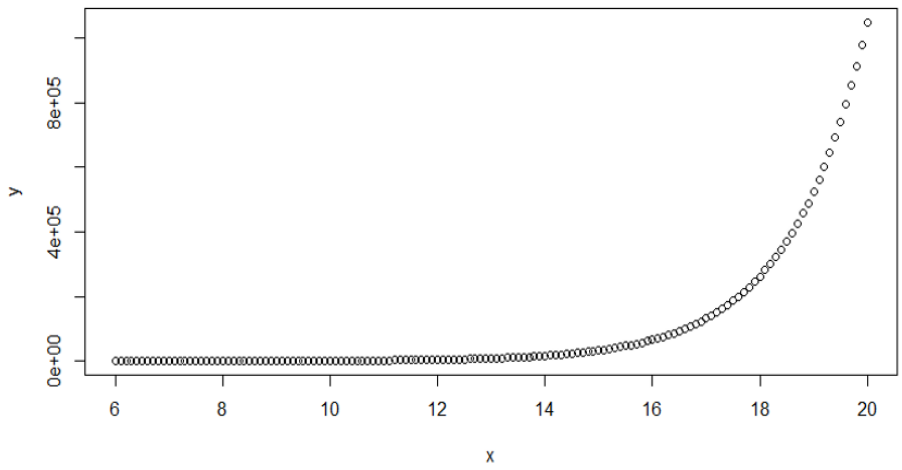


**图 1-4 指数样本增加量分布示意图**

因此，采取以 2 为底数指数分布生成了的 68 组样本容量储存在 sam 变量中，具体取值如表 1-4。另外根据题意，产生了样本量为 100，500，1000，5000，

10000 的变量 sam0。

表 1-4　样本容量表

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2195 | 2353 | 2521 | 2702 | 2896 | 3104 | 3327 | 3566 |
| 3822 | 4096 | 4390 | 4705 | 5043 | 5405 | 5793 | 6208 |
| 6654 | 7132 | 7643 | 8192 | 8780 | 9410 | 10086 | 10809 |
| 11585 | 12417 | 13308 | 14263 | 15287 | 16384 | 17560 | 18820 |
| 20171 | 21619 | 23170 | 24834 | 26616 | 28526 | 30574 | 32768 |
| 35120 | 37641 | 40342 | 43238 | 46341 | 49667 | 53232 | 57052 |
| 61147 | 65536 | 70240 | 75281 | 80684 | 86475 | 92682 | 99334 |
| 106464 | 114105 | 122295 | 131072 | 140479 | 150562 | 161369 | 172951 |
| 185364 | 198668 | 212927 | 228210 | | | | |

### 4. 简单随机抽样原理

根据简单随机抽样的原理，只需要根据不同的样本容量，从总体中直接随机抽取相应的样本数量即可。利用 matrix 函数将每个样本量里面的具体样本数放在一个矩阵里。通过并行计算，利用 apply 函数可以提高效率，减少工作量。

在编写简单随机抽样函数时，按照给定样本数进行抽样，由于需要保证返回值长度相同（矩阵需要其所有列向量长度相同），因而缺少的部分利用一维的缺失值矩阵进行补充，缺失值矩阵的长度为抽样样本数的最大值和样本长度的差值。

### 5. 计算样本质量

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

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

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。

根据样本质量的公式编写出的 quality 函数，计算出简单随机抽样在样本量

分别为 100，500，1000，5000，10000 时的样本质量如表 1-5 所示。但需要注意的是，为消除抽样的随机性影响，在此使用 100 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。

**表 1-5　4 种给定样本量的样本质量表**

| 样本容量 | 100 | 1000 | 5000 | 10000 |
|---|---|---|---|---|
| 样本质量 | 0.65428 | 0.98308 | 0.99627 | 0.99845 |

可以看出，样本质量随着样本容量的增加在逐渐增加，在样本量达 1000 时，已经获得了超过 98%的样本质量。将四种样本下的样本质量画成样本质量曲线如图 4 所示。

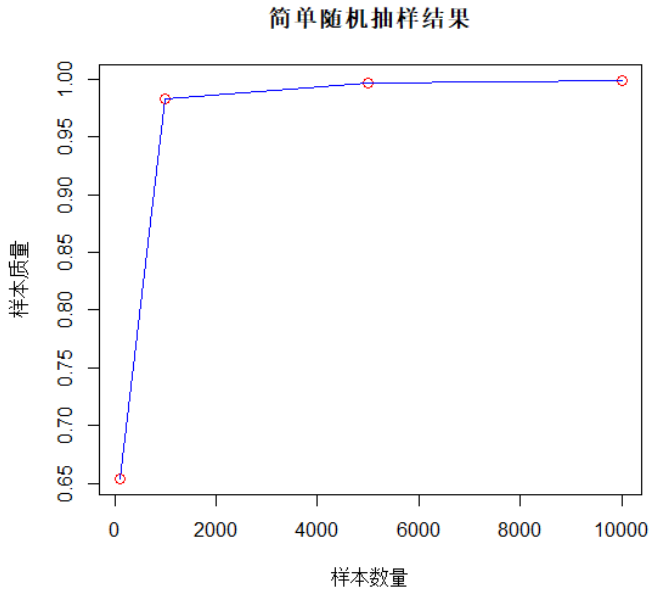


**图 1-5　4 个样本量下的样本质量曲线**

再计算 68 组样本容量下的样本质量。再画出简单随机抽样的样本质量随样本容量变化的曲线如图 1-6 所示

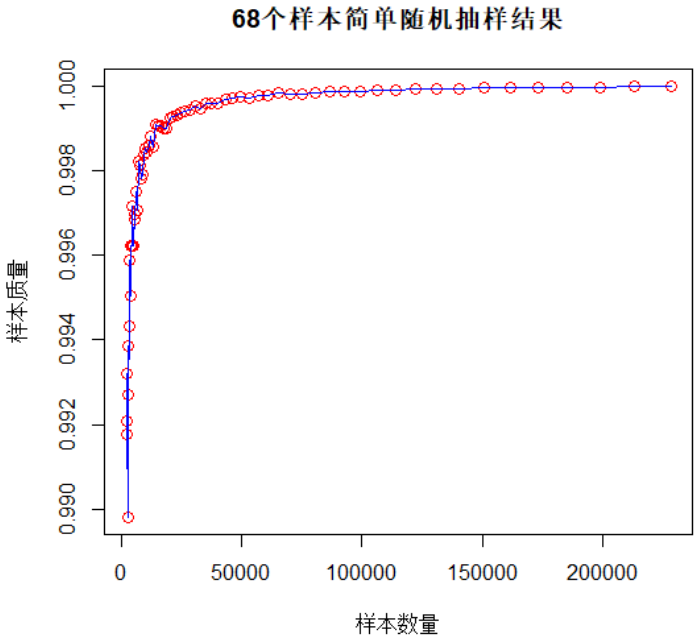


图 1-6　68 个样本量下的样本质量曲线

## 6. 样本质量曲线拟合与预测

对 68 个样本曲线进行样条插值拟合与预测，将对比图线画出如图 1-7 所示。可以看出，样本的拟合效果较好，真实值基本都落在样本质量拟合曲线上。

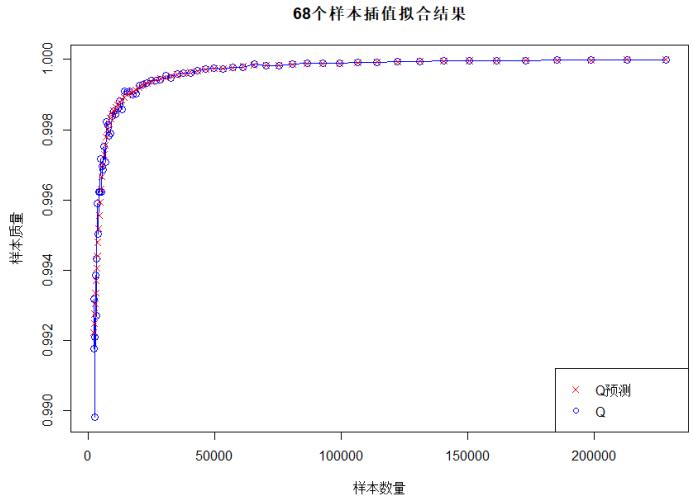


图 1-7　68 个样本插值拟合结果

## 7. 预测不同样本质量下的样本容量

由图 1-6 和图 1-7 可以看出，样本量为 2195，即第一个样本量时，样本质量就超过 0.99，接近于 1。由于总体样本量为 235629，因此当样本容量为总体

样本的接近 1%左右时，就可以快速得到精确的样本抽样结果，这在很大程度上提高计算的效率，降低了计算成本。根据样条插值预测的，不同精度下的不同样本数量，如下表 1-6 所示。

**表 1-6 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 | 527 | 878 | 1316 | 2975 |

实际操作中，需要先对要研究的情况不同，灵活选用抽样方法，根据精度需求，选择最优样本容量以达到尽可能高的样本质量，保证抽样的有效性。

### 8. 最优样本质量的确定

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 0.1%的进步才能继续增加样本。

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：

```
> print(Qm)
[1] 0.954445
> count
[1] 700
```

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：700 个，且能达到的样本质量为 95.4445%，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 700 个。

## 9. R 语言代码

```
###读取数据
mydata<-read.csv("D:/大数据作业/大数据统计基础考试/2017《大数据统计基础》考试题/2017《大数据统计基础》考试题/LoanStats3c.csv",header=TRUE,skip=1)
#删除缺失值
mydata0<-mydata[,'loan_amnt']
N<-length(mydata0)
summary(mydata0)
mydata0<-na.omit(mydata0)
#离散化数据
plot<-hist(mydata0,xlab="贷款金额",ylab="频数", main = "贷款量分布频数分布直方图",col="lightblue")
databreaks<-plot$breaks
mydata1<-
  mydata0 %>%
  cut(breaks=databreaks)
freq<-table(mydata1)/N
barplot(table(mydata1),xlab = "分组间隔",ylab = "数量",col = "lightblue",
        main = "各分组数量条形图",border = "red")
#重新生成分组间隔
databreaks=c(0,1200*(1:9),14000,18000,23000,2400*(11:14),34500,max(mydata0))
mydata2=cut(mydata0,breaks = databreaks)
barplot(table(mydata2),xlab = "分组间隔",ylab = "数量",col = "lightblue",
        main = "各分组数量条形图",border = "red")
freq1<-table(mydata2)/N    #计算再次分组的分组频率
##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)
#给定样本量
sam0<-c(100,1000,5000,10000)
n<-length(sam0)

###简单随机抽样
#定义简单随机抽样的抽样函数 fun1
fun1<-function(i){
  p<-sample(mydata0,i)
  p<-c(p,matrix(NA,1,samp[n]-length(p)))
  return(p)
}

```

```
###定义简单随机抽样样本质量函数
fun2<-function(datasam){
  datasam1<-cut(na.omit(datasam),breaks=databreaks)
  frequ<-table(datasam1)/length(na.omit(datasam))+0.0000000001  #防止某个分组概率为零
  J<-sum((frequ-freq1)*(log(frequ/freq1)))
  q<-exp(-J)
  return(q)
}
###10 次循环计算平均样本质量
sam0<-as.matrix(sam0)
Q1<-matrix(NA,length(sam0),10)
for(i in 1:10){
  ma<-apply(sam0,1,fun1)    #抽样指标 loan_amnt 抽样样本矩阵
  Q1[,i]<-apply(ma,2,fun2)     #计算每个抽样样本下的样本质量
}
Qj<-apply(Q1,1,mean)
###绘制样本质量图
plot(sam0,Qj,xlab = "样本数量",ylab = "样本质量",main="简单随机抽样结果",col = "red",bg="lightblue",cex = 1.2)
lines(sam0,Qj,col='blue')
###10 次循环求 68 个样本的平均样本质量
samp<-as.matrix(samp)
n<-length(samp)
Q1<-matrix(NA,length(samp),10)
for(i in 1:10){
  ma<-apply(samp,1,fun1)    #抽样指标 loan_amnt 抽样样本矩阵
  Q1[,i]<-apply(ma,2,fun2)     #计算每个抽样样本下的样本质量
}
Qj<-apply(Q1,1,mean)
###绘制样本质量图
plot(samp,Qj,xlab = "样本数量",ylab = "样本质量",main="68 个样本插值拟合结果",col = "blue",cex = 1.2)
lines(samp,Qj,col='blue')
###绘制预测样本质量图
s<-smooth.spline(samp,Qj)
pr<-predict(s,samp)
points(pr$x,pr$y,pch=4,col='red')
legend("bottomright", c("Q 预测", "Q"), pch = c(4, 1), col = c("red", "blue"), cex = 1)
#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]])

#选取最优样本
##计算最优样本量，停止增加是样本量的条件
##1、最小样本质量为 95%
##2、前后两次增加样本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 0.1%的进步
##每次增加样本步长为 25 个,起始为 300
  count<-300
  Q<-vector(mode="numeric",length=0)
  Qm<-0
  while(Qm<0.95||(Qm-Qm1)>0.001){
    for(i in 1:5){
      ma<-fun1(a)
      Q[i]<-fun2(ma)
    }
    Qm1<-Qm
    Qm<-mean(Q)
    count<-count+25
  }
  print(Qm)
  count
```

## 二、数据预处理

1、（10 分）UCI 鲍鱼数据

**数据说明：**abalone 数据是 AppliedPredictiveModeling 包中自带的数据集，该数据集是关于 UCI 鲍鱼样本的一系列变量，总共包含 9 个变量，共 4177 个样本。因变量是鲍鱼年龄（鲍鱼环数加 1.5），鲍鱼的年龄从 2.5 到 30.5 不等。

**数据处理：**

```
#加载所需程序包
library(ggplot2)
library(AppliedPredictiveModeling)
library(ggplot2)
library(dummies)
library(lars)
library(glmnet)
library(grid)
library(GGally)
library(MASS)
library(car)
library(caret)
library(lattice)
library(minerva)
##1.abalone 鲍鱼数据处理

#读取 abalone 数据，并查看数据
data(abalone)
any(is.na(abalone))  #检查 abalone 数据是否有缺失值
head(abalone)
str(abalone)
summary(abalone)
#因变量是 Rings，是 int 类型，属于数值因变量
#其他变量都是解释变量
```

1）对数据作图估计预测变量和被解释变量之间的函数关系。

R 语言代码：

```
####(1)对数据作图估计预测变量和被解释变量之间的函数关系。
vp <- function(x, y) {
  viewport(layout.pos.row = x, layout.pos.col = y)
}
grid.newpage()
pushViewport(viewport(layout = grid.layout(4, 2))) #切割分面作图
#设置自己的主题模板
mytheme =theme(plot.title = element_text(hjust = 0.5,family="myFont",size=18,color="gold3"),panel.background=element_rect(fill='aliceblue'))
plot1=ggplot(abalone,aes(x=LongestShell,y=Rings))+geom_point(alpha=0.2)+mytheme+geom_smooth(color='red')
plot2=ggplot(abalone,aes(x=Diameter,y=Rings))+geom_point(alpha=0.2)+mytheme+geom_smooth(color='red')
plot3=ggplot(abalone,aes(x=Height,y=Rings))+geom_point(alpha=0.2)+mytheme+geom_smooth(color='red')
plot4=ggplot(abalone,aes(x=WholeWeight,y=Rings))+geom_point(alpha=0.2)+mytheme+geom_smooth(color='red')
plot5=ggplot(abalone,aes(x=ShuckedWeight,y=Rings))+geom_point(alpha=0.2)+mytheme+geom_smooth(color='red')
plot6=ggplot(abalone,aes(x=VisceraWeight,y=Rings))+geom_point(alpha=0.2)+mytheme+geom_smooth(color='red')
plot7=ggplot(abalone,aes(x=ShellWeight,y=Rings))+geom_point(alpha=0.2)+mytheme+geom_smooth(color='red')
plot8=ggplot(abalone,aes(x=factor(Type),y=Rings,fill = I("lightblue")))+geom_boxplot()+xlab("鲍鱼种类")+ylab("数量")+mytheme
print(plot1,vp = vp(1,1))
print(plot2,vp = vp(1,2))
print(plot3,vp = vp(2,1))
print(plot4,vp = vp(2,2))
print(plot5,vp = vp(3,1))
print(plot6,vp = vp(3,2))
print(plot7,vp = vp(4,1))
print(plot8,vp = vp(4,2))
```

通过以上代码生成 abalone 数据集因变量与自变量的关系图，如图 2-1 所示，前七个图是数值变量与 rings 也就是鲍鱼寿命之间的关系，下图中可以看到一条红色回归直线，几个数值因变量都和鲍鱼的寿命有着貌似正的相关关系，而唯一的分类变量鲍鱼种类与存活寿命有着较强的关系，从箱线图中可以看出，三

种鲍鱼平均寿命相差不大，但明显 1 鲍鱼的寿命较短些。

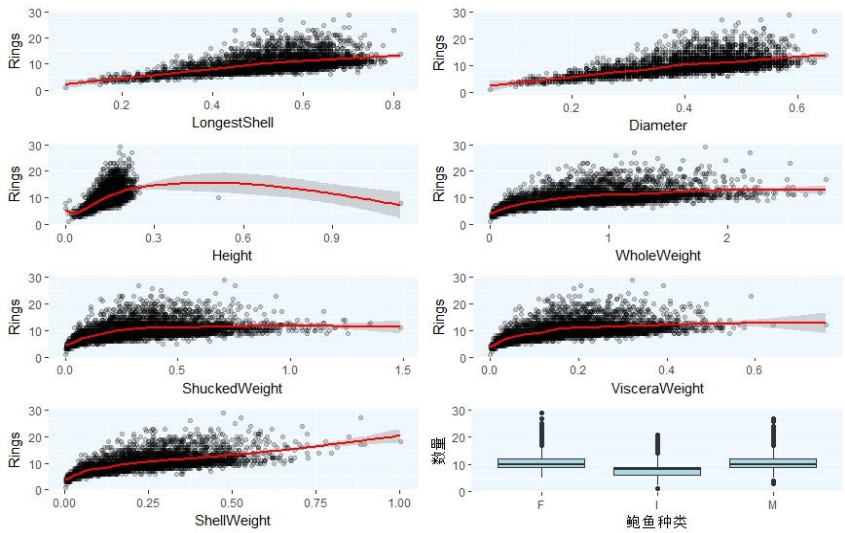


**图 2-1 abalone 数据集因变量与自变量的关系**

**2）用散点图和相关系数图解释预测变量之间的相关性。**

**R 语言代码：**

```
####(2)用散点图和相关系数图解释预测变量之间的相关性。
names(abalone)
ggpairs(abalone[,setdiff(names(abalone),c('Rings','Type'))])
library(corrplot)
#求解相关矩阵
cormatrix=cor(abalone[,setdiff(names(abalone),c('Rings','Type'))])
#画出相关矩阵图，并采用聚类算法对变量进行了聚类
#变量间相关性很高
corrplot(cormatrix,order = "hclust",addrect = 3,rect.col = "black",tl.cex =
    0.7,tl.col = "black")
```

通过以上代码生成预测变量两两之间的散点图以及相关系数图矩阵图，如图 2-2 和图 2-3，从两张图中可以看出预测变量之间的两两关系比较明显，相关系数都较大。

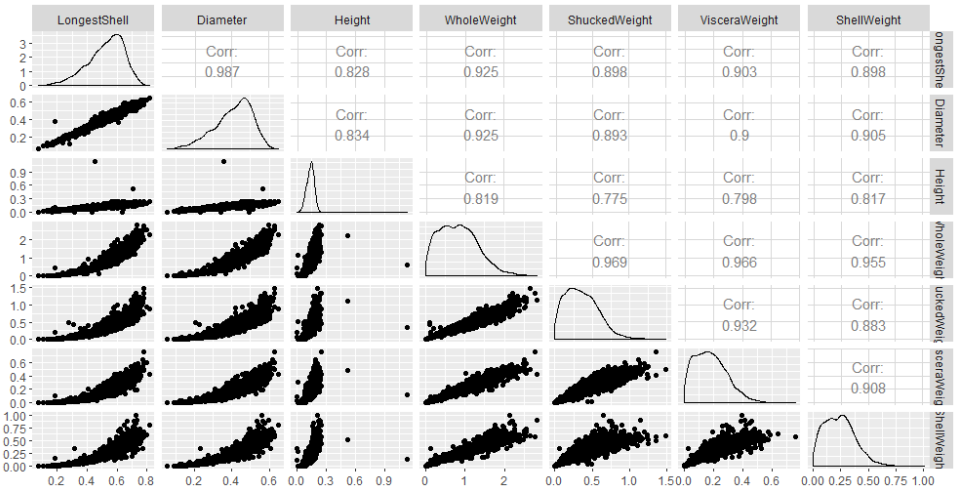


图 2-2 相关系数散点图

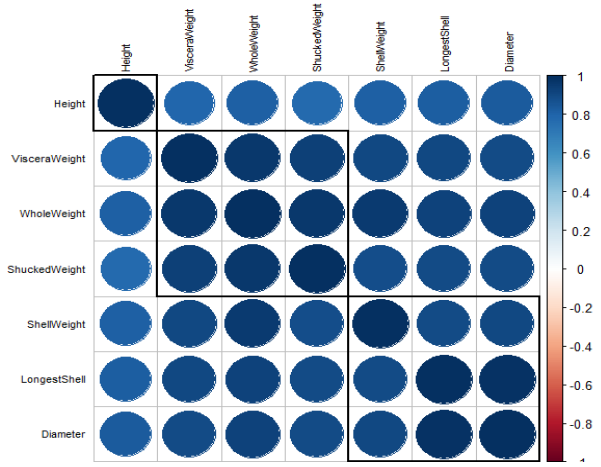


图 2-3 相关系数聚类图

3）对预测变量估计重要性得分。找一种筛选方法得到预测变量子集，该集合不含冗余变量。

R 语言代码：

```
####(3)对预测变量估计重要性得分。找到一种筛选方法得到预测变量子集，该集合不含冗余变量。
#计算数值型变量得分的
#方法 1
#filterVarImp 可以建立 LOESS 模型，确定数值变量和因变量之间的关系，nonpara 参数表示是非参数回归
loess <- filterVarImp(x = abalone[,2:8],y = abalone[,9],nonpara = TRUE)
loess
```

```
#方法 2
#尝试使用 MIC(最大信息数)来筛选预测变量
micvalue<- mine(abalone[,2:8],abalone[,9])#计算结果包含很多个统计量
micvalues$MIC
#方法 3，
#计算各预测变量和结果变量的相关系数
corr <- apply(abalone[,2:8],MARGIN = 2,FUN = function(x,y) cor(x,y),y = abal
one$Rings)
corr
#分类预测变量重要性得分
#使用三个水平方差分析
aovResult <- aov(abalone$Rings~abalone$Type)
summary(aovResult)#结果显示不同 Type 之间存在显著差异

##lasso 筛选变量子集
onehot_data=dummy.data.frame(abalone,names=c('Type'),sep="_")
x=as.matrix(onehot_data[,setdiff(names(onehot_data),c('Rings'))])
y=as.matrix(onehot_data[,c('Rings')])
model_fit= lars(x,y)
# 10 折交叉验证选择模型
model_cv=cv.lars(x,y,K=10)
select_model=model_cv$index[which.min(model_cv$cv)]
my_coef=coef.lars(model_fit,mode='fraction',s=select_model)
# 使用 cp 选择模型，可以发现需要剔除构造的 TYPE_F 和 LongestShell 变量
cp_select=which.min(model_fit$Cp)
my_coef_cp=coef.lars(model_fit,mode='step',s=cp_select)
# glmnet 包实现
cv.out=cv.glmnet(x,y,alpha = 1)
plot(cv.out)
bestlam=cv.out$lambda.min
fit_model = glmnet(x, y, alpha = 1, nlambda = 40)
lasso_coefficient=predict(fit_model,type = "coefficients",s = bestlam)
```

通过以上代码生成预测变量，可以得到 3 种评价预测变量重要性的得分，分别是 loess，corr，micvalues$MIC，绘制成如下表格：

表 2-1 预测变量估计重要性

| 变量重要性 | LongestShell | Diameter | Height | Whole | Shucked | Viscera | Shell |
|---|---|---|---|---|---|---|---|
| loess | 0.3099 | 0.33023 | 0.3107 | 0.3245 | 0.2691 | 0.3248 | 0.4224 |
| corr | 0.3546 | 0.3652 | 0.3449 | 0.3539 | 0.3153 | 0.3510 | 0.3866 |
| MIC | 0.5567 | 0.5746 | 0.5574 | 0.5403 | 0.4208 | 0.5038 | 0.6275 |

分类变量的重要性得分：

```
> summary(aovResult)#结果显示不同Type之间存在显著差异
                Df Sum Sq Mean Sq F value Pr(>F)
abalone$Type     2   8381    4191   499.3 <2e-16 ***
Residuals     4174  35030       8
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

结果显示鲍鱼类型的对寿命的影响很明显。

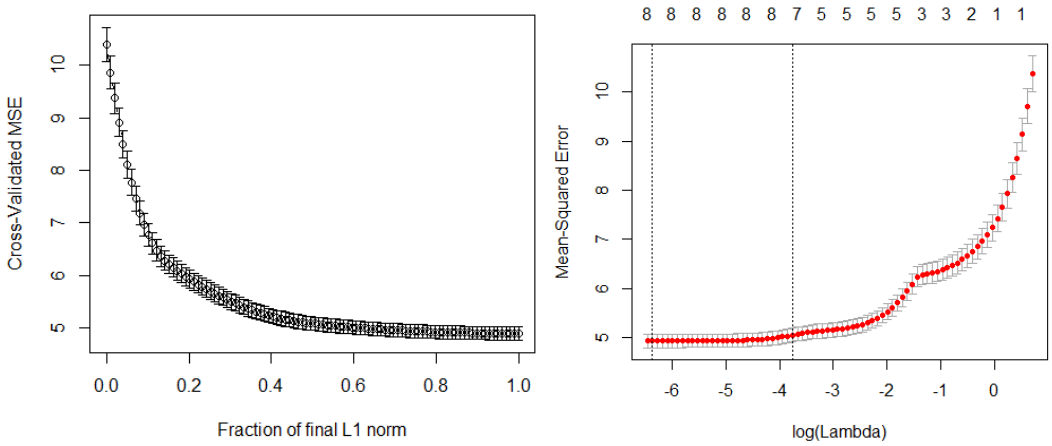


图 2-4 lasso 模型十折交叉检验效果图

利用 lasso 筛选冗余变量，将回归系数小的变量进行 lasso 压缩估计可以发现需要剔除构造的 TYPE_F 和 LongestShell 变量。

**4)对连续型预测变量应用主成分分析，决定多少个不相关的主成分能够代表数据中的信息？**

**R 语言代码：**

```
####(4)对连续型预测变量应用主成分分析，决定多少个不相关的主成分能够代表数据中的信息？
pcafit <- prcomp(abalone[,setdiff(names(abalone),c('Rings','Type'))],center = TRUE, scale = TRUE)#对变量进行中心化和标准化
summary(pcafit) #展示方差贡献率,一个主成分即能贡献 90%以上的方差
# 根据碎石图发现选取第一个变量就足以代表原始数据了
screeplot(pcafit,type="lines",main="碎石图",col="blue")
```

画出碎石图如下图所示：

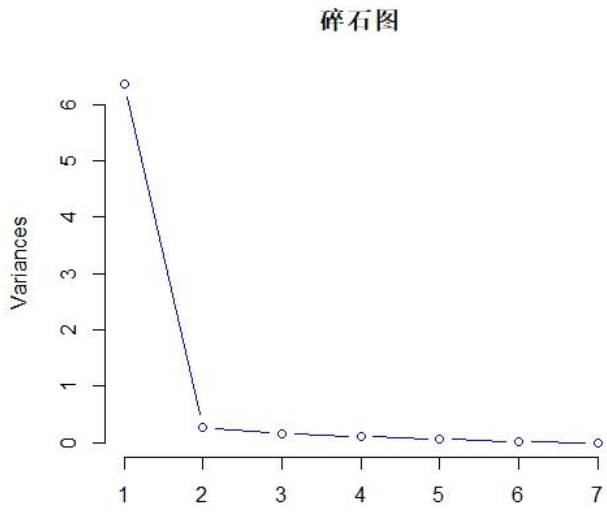


**图 2-5 碎石图**

一个主成分即能贡献 90%以上的方差，选取第一个主成分就足以代表原始数据。

2、（10 分）使用下面的非线性函数来模拟数据$(y,x_1,x_2,x_3,x_4,x_5)$：

$$y=10\sin(\rho x_1x_2)+20(x_3-0.5)^2+10x_4+5x_5+e$$

其中预测变量$x_1$到$x_5$服从均匀分布，误差 e 服从标准正态分布

**1) 写一个 R 函数从该模型中模拟数据。**

```
##数据预处理考试第二题##
#(1)写一个 R 函数从该模型中模拟数据。
Generatordata <- function (n, sd = 1) #这个函数是 R 包里的源代码，直接 copy 过来的(稍作了修改)
{                                     #原始函数是 mlbench 包里的 mlbench.friedman1
  x <- matrix(runif(5 * n), ncol = 5)
  y <- 10 * sin(pi * x[, 1] * x[, 2])
  y <- y + 20 * (x[, 3] - 0.5)^2 + 10 * x[, 4] + 5 * x[, 5]+rnorm(n, sd = sd)
  list(x = x, y = y)
}
```

**2) 随机模拟一个数据集，样本量是 500，绘制图形研究预测变量和被解释变量之间的关系。**

```
#(2)随机模拟一个数据集，样本量是 500，绘制图形研究预测变量和被解释变量之间的关系。
#模拟数据集
set.seed(1119)
data500 = Generatordata(500)#生成的是一个列表，保存了生成的 x 和 y
xdata = data500$x
ydata = data500$y
dim(xdata)
lm_Data <- data.frame(cbind(ydata,xdata))
names(lm_Data)=c("y","x1","x2","x3","x4","x5")
attach(lm_Data)
p1=ggplot(lm_Data,aes(x=x1,y=y))+geom_point(alpha=0.2)+mytheme+geom_smooth(color='red')
p2=ggplot(lm_Data,aes(x=x2,y=y))+geom_point(alpha=0.2)+mytheme+geom_smooth(color='red')
p3=ggplot(lm_Data,aes(x=x3,y=y))+geom_point(alpha=0.2)+mytheme+geom_smooth(color='red')
p4=ggplot(lm_Data,aes(x=x4,y=y))+geom_point(alpha=0.2)+mytheme+geom_smooth(color='red')
p5=ggplot(lm_Data,aes(x=x5,y=y))+geom_point(alpha=0.2)+mytheme+geom_smooth(color='red')
grid.newpage()
pushViewport(viewport(layout = grid.layout(3, 2)))
print(p1,vp = vp(1,1))
print(p2,vp = vp(1,2))
print(p3,vp = vp(2,1))
print(p4,vp = vp(2,2))
print(p5,vp = vp(3,1))
ggpairs(lm_Data)
cormatrix=cor(lm_Data)
corrplot(cormatrix)
```

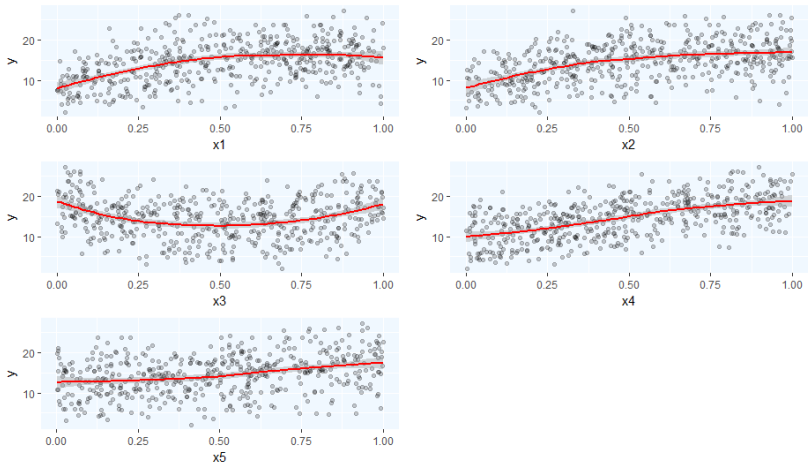


图 2-4 解释变量与被解释变量关系

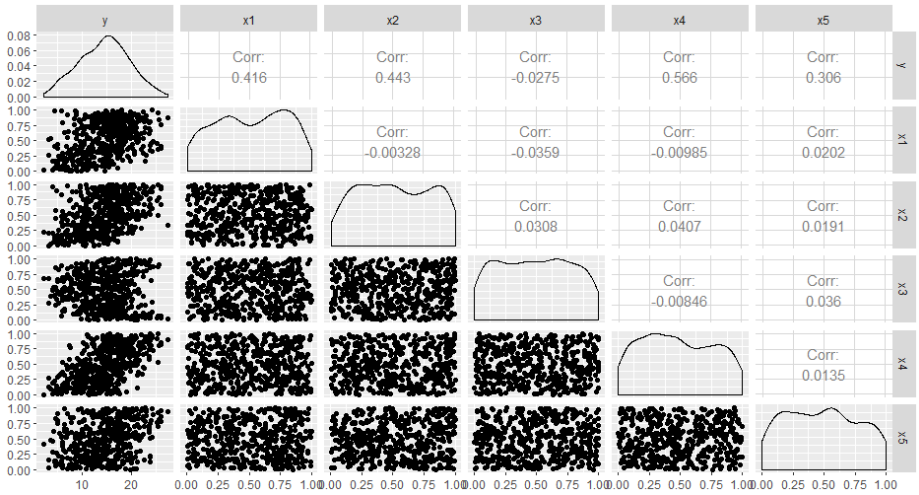


图 2-4 解释变量与被解释变量关系

**3) 使用线性回归中的向前法、向后法和逐步回归等变量选择方法，最终模型选择了哪些变量？**

R 语言代码：

```
#(3)使用线性回归中的向前法、向后法和逐步回归等变量选择方法，最终模型选择了哪些变量
Model <- lm(y~.,data = lm_Data)
summary(Model)#在 summary 里可以看到 v4 的 t 检验不显著

lm_for=stepAIC(Model,direction = "forward")#向前
lm_back=stepAIC(Model,direction = "backward")#向后
```

```
lm_both=stepAIC(Model,direction = "both")#逐步选择
summary(lm_for)# 全部选择
summary(lm_back)# 选择 x1,x2,x3,x4
summary(lm_both)# 选择 x1,x2,x3,x4
```

向前法回归结果：

```
> summary(lm_for)#  选择x1,x2,x4,x5

Call:
lm(formula = y ~ x1 + x2 + x3 + x4 + x5, data = lm_Data)

Residuals:
    Min      1Q  Median      3Q     Max
-9.7603 -1.5394 -0.0014  1.6901  6.3939

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2279     0.4378   0.520    0.603
x1            7.3828     0.3947  18.704   <2e-16 ***
x2            7.2490     0.3863  18.766   <2e-16 ***
x3           -0.5354     0.3848  -1.392    0.165
x4            9.6371     0.3908  24.663   <2e-16 ***
x5            4.9420     0.3886  12.719   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.49 on 494 degrees of freedom
Multiple R-squared:  0.756,     Adjusted R-squared:  0.7535
F-statistic: 306.1 on 5 and 494 DF,  p-value: < 2.2e-16
```

向后法回归结果：

```
> summary(lm_back)# 选择x1,x2,x4,x5

Call:
lm(formula = y ~ x1 + x2 + x4 + x5, data = lm_Data)

Residuals:
     Min       1Q   Median       3Q      Max
-10.0317  -1.5170   0.0374   1.6982   6.1242

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.02954    0.39714  -0.074    0.941
x1           7.40291    0.39483  18.750   <2e-16 ***
x2           7.23261    0.38647  18.715   <2e-16 ***
x4           9.64287    0.39110  24.656   <2e-16 ***
x5           4.92243    0.38866  12.665   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.492 on 495 degrees of freedom
Multiple R-squared:  0.755,     Adjusted R-squared:  0.753
F-statistic: 381.4 on 4 and 495 DF,  p-value: < 2.2e-16
```

向逐步回归结果：

```
> summary(lm_both)# 选择x1,x2,x4,x5

Call:
lm(formula = y ~ x1 + x2 + x4 + x5, data = lm_Data)

Residuals:
     Min       1Q   Median       3Q      Max
-10.0317  -1.5170   0.0374   1.6982   6.1242

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.02954    0.39714  -0.074    0.941
x1           7.40291    0.39483  18.750   <2e-16 ***
x2           7.23261    0.38647  18.715   <2e-16 ***
x4           9.64287    0.39110  24.656   <2e-16 ***
x5           4.92243    0.38866  12.665   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.492 on 495 degrees of freedom
Multiple R-squared:  0.755,     Adjusted R-squared:  0.753
F-statistic: 381.4 on 4 and 495 DF,  p-value: < 2.2e-16
```

**4)应用不同的过滤法，逐个评估变量。一些过滤法同时评估多个变量（如 ReliefF 算法），两个有交互效应的预测变量 $x_1$ 和 $x_2$ 否被选中了？是否倾向于选择其中某一个变量？**

**R 代码：**

```
#方法 1，使用相关系数进行筛选
#定义计算相关系数的函数
calc_cor <- function(x,y){
  corValue = cor(x,y)
  corValue
}
#定义筛选相关系数的标准
#大于 0.2 的输出为 TRUE
corSelection <- function(corValue,x,y){
  corSelected <- (corValue >0.2 & corValue <0.75 )
  corSelected
}
#lmSBF 表示选择的是 lm(线性)的 SBF
corFunction <- lmSBF
corFunction$score <- calc_cor
corFunction$summary <- defaultSummary
corFunction$filter <- corSelection
```

```
corCtrl <- sbfControl(method = "repeatedcv",
                      repeats = 5,
                      verbose = TRUE,
                      functions = corFunction )
corFilter <- sbf(lmData[,2:6],
                 lmData$Y,
                 tol = 1.0e-12,
                 sbfControl = corCtrl)
corFilter #结果显示，删除了变量 x3

#方法 2，使用 MIC 统计量

#定义计算 MIC 的函数
library(minerva)
calc_MIC <- function(x,y){
  MicValue = mine(x,y)
  MicValue$MIC
}

#定义筛选 MIC 的标准
micSelection <- function(MicValue,x,y){
  micSelected <- (MicValue > 0.2)
  micSelected
}

#lmSBF 表示选择的是 lm(线性)的 SBF
micFunction <- lmSBF
#将上面定义的函数传入到筛选器里面去
micFunction$score <- calc_MIC
micFunction$summary <- defaultSummary
micFunction$filter <- micSelection
micCtrl <- sbfControl(method = "repeatedcv",
                      repeats = 5,
                      verbose = TRUE,
                      functions = micFunction )
micFilter <- sbf(lmData[,2:6],
                 lmData$Y,
                 tol = 1.0e-12,
                 sbfControl = micCtrl)
micFilter #结果显示删除了 x3,x5
```

过滤法结果如下：

```
> corFilter #结果显示，删除了变量x3

Selection By Filter

Outer resampling method: Cross-Validated (10 fold, repeated 5 times)

Resampling performance:

 RMSE Rsquared   MAE RMSESD RsquaredSD  MAESD
 2.49   0.7555 1.959 0.2672    0.05908 0.2068

Using the training set, 4 variables were selected:
   x1, x2, x4, x5.

During resampling, the top 4 selected variables (out of a possible 4):
   x1 (100%), x2 (100%), x4 (100%), x5 (100%)

On average, 4 variables were selected (min = 4, max = 4)
```

重抽样的结果 x1,x2,x5,x4,百分百被选中，x3 则是 98%选中，所以 micFilter 结果全部选中。

```
Selection By Filter

Outer resampling method: Cross-Validated (10 fold, repeated 5 times)

Resampling performance:

  RMSE Rsquared   MAE RMSESD RsquaredSD  MAESD
 2.496   0.7573 1.966 0.2577    0.06149 0.2173

Using the training set, 5 variables were selected:
   x1, x2, x3, x4, x5.

During resampling, the top 5 selected variables (out of a possible 5):
   x1 (100%), x2 (100%), x4 (100%), x5 (100%), x3 (98%)

On average, 5 variables were selected (min = 4, max = 5)
```

结论：x1 和 x2 在两种过滤法中都被选中了,二者之间倾向性不明显。

```
#relieff 选择
library(caret)
library(CORElearn)
reliefValues=attrEval(y ~ .,data = result,estimator = "ReliefFequalK",ReliefIterations = 50)
reliefValues1=attrEval(y ~ .,data = result,estimator = "ReliefFexpRank",ReliefIterations = 50)
#也可以看出 x1,x2 有相同的倾向性，由于重要性相近，很难进行选择。
reliefValues
reliefValues1

```

```
#非过滤法，loess 选择,但可以看出 x1,x2 基本由相同的倾向性
loessresult=filterVarImp(x=result[,-6],y=result[,6],nonpara=TRUE)
loessresult
loessresult=loessresult$Overall[loessresult$Overall>=0.05]
loessresult
```

运行得到的结果如下：

```
> #也可以看出x1,x2有相同的倾向性，由于重要性相近，很难进行选择。
> reliefValues
       x1        x2        x3        x4        x5
0.8622889 0.8640325 0.8643866 0.8651041 0.8681325
> reliefValues1
       x1        x2        x3        x4        x5
0.8650783 0.8637393 0.8675084 0.8648948 0.8602004
```

```
> #非过滤法，loess选择,但可以看出x1,x2基本由相同的倾向性
> loessresult=filterVarImp(x=lm_Data[,-6],y=lm_Data[,6],nonpara=TRUE)
> loessresult
      Overall
y  0.102846293
x1 0.004385644
x2 0.012433650
x3 0.011520710
x4 0.014963622
> loessresult=loessresult$Overall[loessresult$Overall>=0.05]
> loessresult
[1] 0.1028463
```

## 三、数据探索性分析

请见附件 Rmarkdown

## 四、数据可视化

### 1. 数据来源及变量说明

（1）**数据来源：**表格 LoanStats3c 数据集提供了 2013 年到 2014 年成功在 Lending Club 上成功申请到贷款的人的信息。

（2）**数据变量说明：**LoanStats3c 表格中包含的信息变量有贷款编号、贷款人标号、贷款人申请的贷款数量、投资者提供的贷款数量、贷款人的基本信息（年龄、房屋的所有权、贷款量、借款的目的、借款人的信用评级）、贷款人的还款情况（每月还款额、是否还清、上一次还款发生在什么时间以及还款的数额等）、贷款人的信用记录（开过多少张信用卡、首次开设信用卡的时间、以及信用卡还款延期次数）、借款得利息、最终偿付得总金额等。

（3）读取数据 LoanStats3c，通过 read.csv 函数读取数据，由于原数据的第一行为数据来源说明，所以需跳过，因而设置参数 skip＝1。最后得到的样本量为 N=235633，变量个数 M=52。

（4）由于数据中存在大量缺失值，使用 na.omit()删去缺失值得到 loan 数据集，样本量 n=13252。同时，提取需要数据的所在列，得到实验变量个数 m=52。可视化实验变量说明表如下表：

表 4-1 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<br>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<br>贷款数量越大说明贷款额度高，贷款分级高。<br>单位为：美元 |

**2. 分面的风玫瑰图，玫瑰叶片的颜色至少四种；（4 分）**

使用信用等级（grade）,工作时间（emp_length）两个变量画风玫瑰图，使用房屋使用权（home_ownership）对图形进行分面，最终得到分面的风玫瑰图如图 4-1 所示。

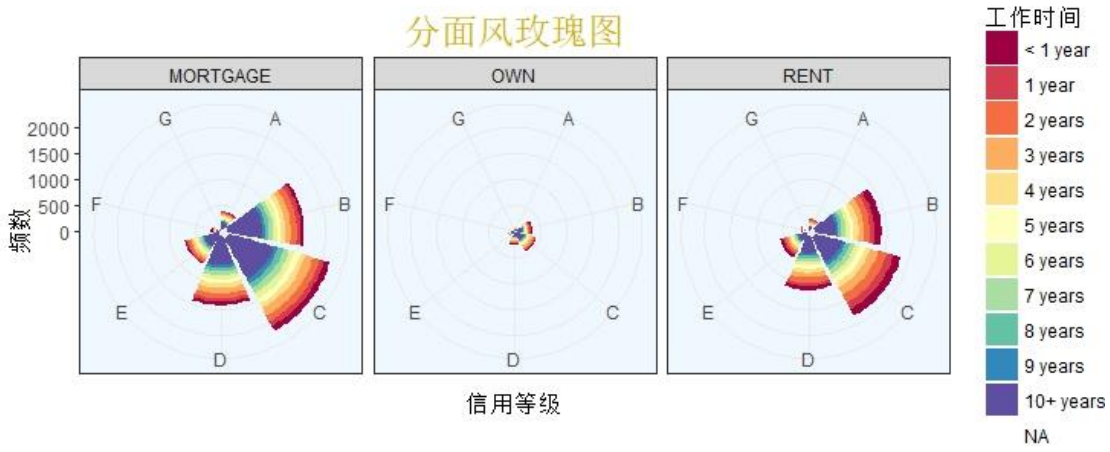


**图 4-1 分面风玫瑰图**

从分面风玫瑰图中可以看出，房屋产权归自有的数量非常少，因此 own 玫瑰图显得非常小，抵押 Mortgage 的房子最多，玫瑰图尺寸最大， 租赁房比抵押房略少，说明大部分人都是租房或者抵押房子，拥有自己的房子的人较少。可以明显看出，工作时间在 10 年以上的频数最高，其他年数的人数相差并不明显。其次，三个风玫瑰图的形状大抵相同，说明房屋的产权对信用评级和工作时间都没有明显的影响，通过抵押房来看信用等级和工作时间的关系。另外，信用等级为 C 的人数最多，其次是 B 和 D。最后，值得注意的是，在租赁房子的人当中，信用等级为 A 的最多，因为这部分人可能房屋做抵押，十分注重个人信誉。

**3.某一个连续型变量的分布直方图，并加入拟合分布线，直方图的组距和组数自己设定（不要使用默认的），并且每个柱子里面填上相应的组的频数，整个图片加上一个黑色的外框，并且图的底色为浅色，柱子为深色，在密度最高的部分加上文字标注“此处密度最大”；**

利用 R 中自带的直方图工具来绘图。利用的变量为贷款总额、年收入等多个连续变量的分布情况。在 loanstatC 数据集中，贷款总额、承诺贷款总额两个变

量的分布较为均匀且数值相同。本次作图选择承诺贷款总额变量，即承诺贷款总额进行绘图。根据分布情况，将变量分为 17 组，每组宽恰好为 2000，这也为之后标注各组频数提供了便利。同时，绘制出以该变量的均值和标准差为参数的正态密度分布曲线和原始数据的核密度曲线，标注出密度最大位置。绘制图形如下：

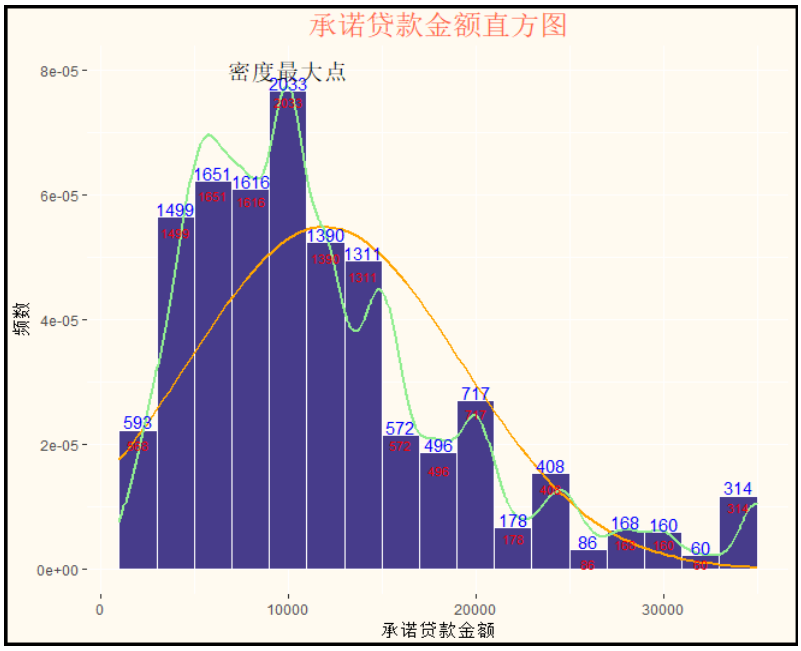


**图 4-2 承诺贷款总额直方图**

由直方图可知，承诺贷款总额分布整体呈现右偏，说明存在极大值，经查为 35000 美元。从图中，还能看出承诺贷款金额主要集中在［0，22000］这个区间内，贷款 10000 美元的客户最多。

**4. 某两个连续型变量的密度图，并且在图中找出一个部分加上一个方框与其他部分区别开来（比如说，密度最高或者最低的区域，用一个矩形的方框加以标示）**

使用贷款金额（loan_amnt）和年收入（annual_inc）两个变量画密度图，由于年收入取值范围较大，为 7450-980000，作图时对年收入取 log，横轴变量选择贷款总额，纵轴变量选择 log（年收入）最终得到密度图如图 4-3 所示。

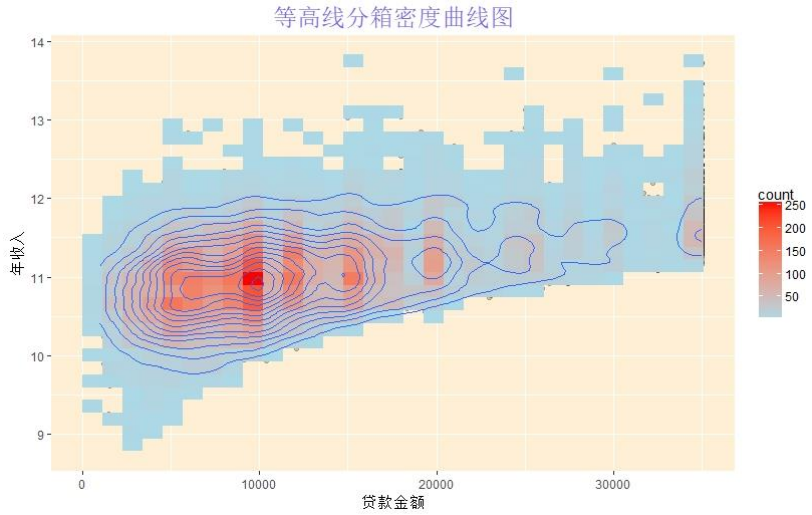


**图 4-3 等高线分箱密度曲线图**

横轴变量选择贷款总额，纵轴变量选择年收入，绘制密度图。绘制图形如下：

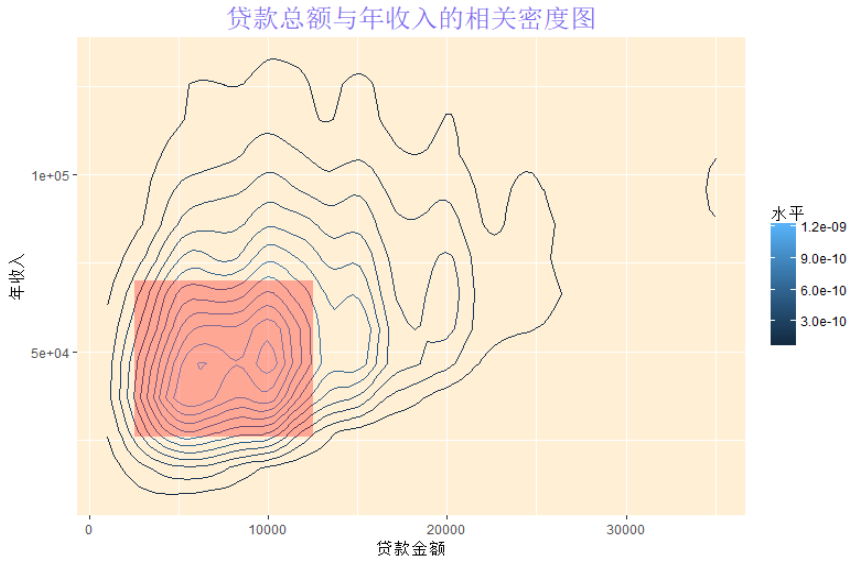


**图 4-4 贷款总额与年收入相关密度图**

从密度图可以看出，贷款人群集中于年收入在［2.6e4，7e4］美元，贷款总额在［2500，12500］美元的范围内。

**5.挑选多个连续型变量，进行聚类，并且绘制相应的热图，并进行美化（可以不用全部样本）。**

使用贷款金额（loan_amnt），每月还款额度（dti），分期付款（installment），年收入（annual_inc），总支付额（total_pymnt）这五个变量从总体中随机抽取100 个样本构造数据集 dat，首先进行标准化，再计算欧式距离，进行聚类并画出热图，最终得到热图如图 4-5 所示。

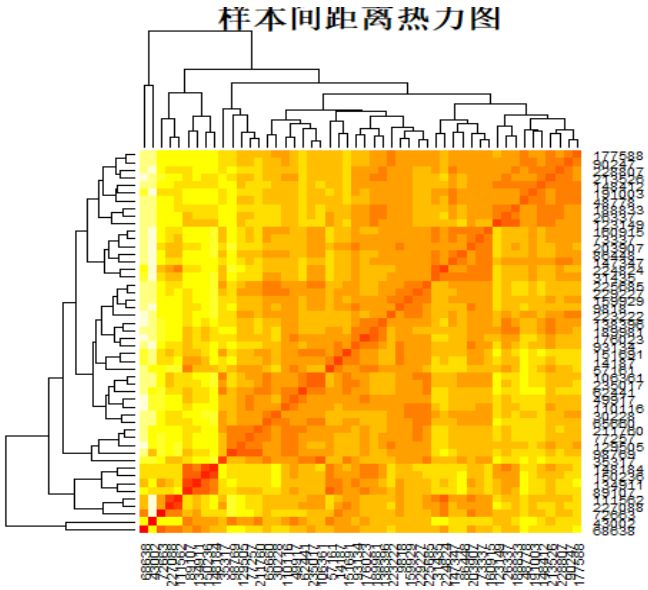


图 4-5　样本间距热力图

如图 4-6 显示了对热力图的美化效果，为热力图增加了明显的样本间分割线以及颜色正态分布图例，如下图所示：

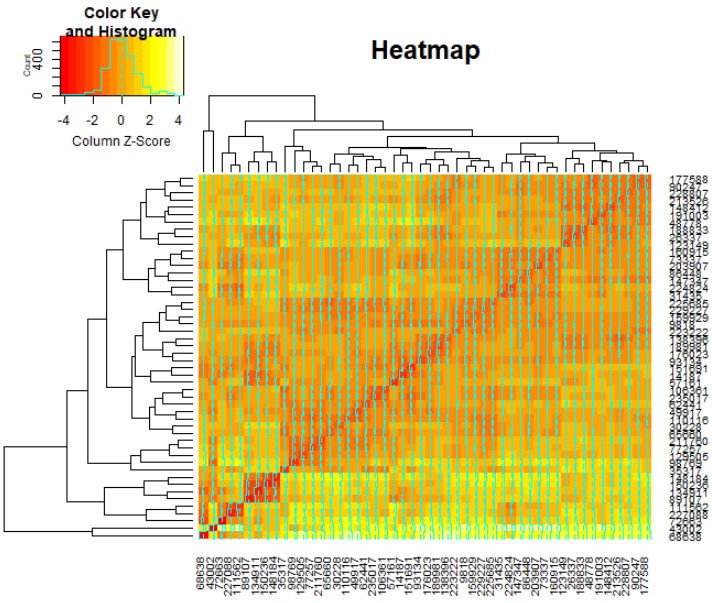


图 4-6　美化后样本间距热力图

**2、使用 province 数据中合适的变量，绘制两幅不同的图，进行空间数据的展示。（10 分）**

**1）对中国 2016 年各省市人口分布情况的空间数据展示**

首先利用 R 语言自带的 plot 绘图函数进行地图的绘制，使用的地图数据为 bou2_4p.shp，1997 年的 gis 中国地理数据。利用中国 2016 年的 province 数据中的年末人口进行空间统计展示，有图例所示，黄色向黑黄色的转变体现了人口密度的增长。如图 4-6 所示：

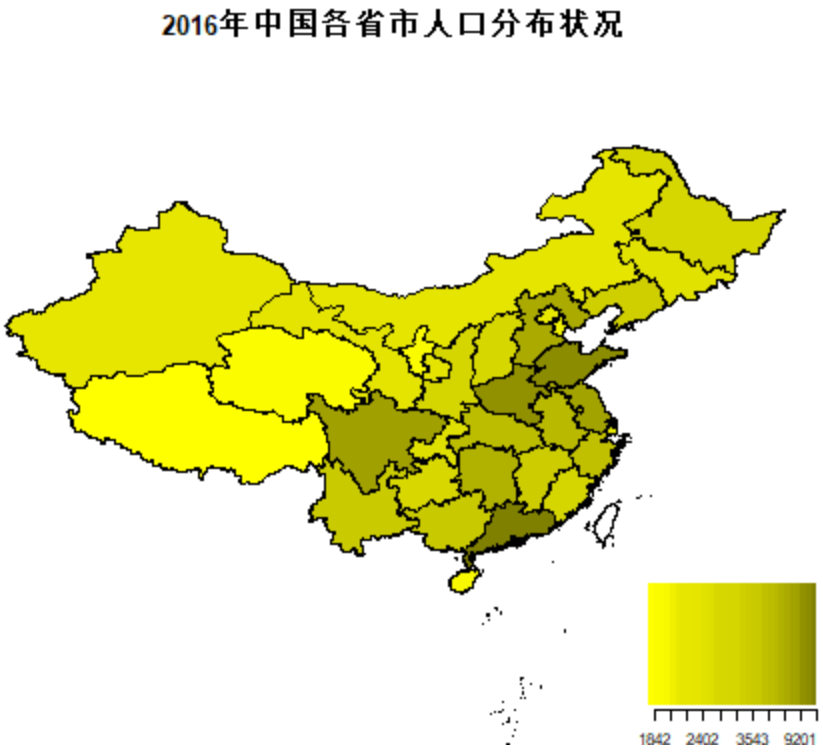


**图 4-7　2016 年中国各省市人口分布状况**

在图中各省市的不同颜色中不难发现，中国的人口分布仍然是东部人口较为密集，西部人口比较稀疏，先对来说河南，河北，山东，四川，广东及各省市的人口较为密集。

**2）对 2016 年中国各省市人口分布情况和地区生产总值的空间数据展示**

利用 ggplot 的地图绘制包进行空间展示，由于 ggplot 绘制地图比 R 语言自带的绘制工具高端，可以在一张地图上显示多个变量，因此本次试验在地图上除了显示各省市人口以外，还进行各省市地域名称展示，以及地区生产总值的展示，多项变量对比分析，如图 4-8 所示：

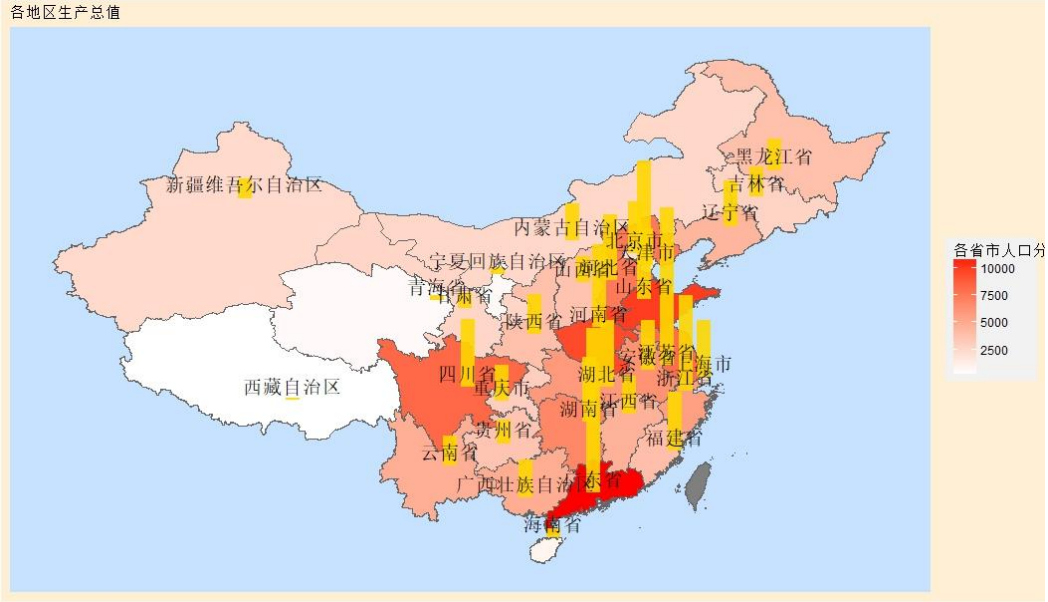


**图 4-8　2016 年中国各省市生产总值分布状况**

在 ggplot 地图中可以看出，首先人口较为多的城市地区生产总值自然就高，其次不难发现，东部的地区生产总值要明显高于西部，而四大直辖市的地区生产总值也比较高。山东省和江苏省的地区生产总值也较高。

代码：

```
###读入数据
setwd("D:/大数据作业/大数据统计基础考试/2017《大数据统计基础》考试题/2017《大数据统计基础》考试题/")
mydata<-read.csv("D:/大数据作业/大数据统计基础考试/2017《大数据统计基础》考试题/2017《大数据统计基础》考试题/LoanStats3c.csv",header=T,skip=1)
###删除缺失值
mydata0<-na.omit(mydata)
###加载包
library(ggplot2)
library(gplots)
library(plyr)
```

```
library(dplyr)
library(showtext)  #使作图的字体更加丰富
library(RColorBrewer)  #增加调色板
library(maps)
library(mapdata)
library(maptools)
library(graphics)
library(REmap)
showtext_auto(enable = T)
showtext_begin()

###question1:分面的风玫瑰图
label<-
c("< 1 year","1 year","2 years","3 years","4 years","5 years","6 years","7 y
ears","8 years","9 years","10+ years")
mydata0$emp_length<- ordered(mydata0$emp_length, levels = label)
ggplot(mydata0,aes(x=grade,fill=emp_length))+
  geom_bar()+coord_polar(theta = 'x')+
  scale_fill_brewer(palette='Spectral')+facet_wrap(~home_ownership)+theme_bw
()+
  labs(x="信用等级",y="频数",fill="工作时间",title='分面风玫瑰图
')+scale_x_discrete()+coord_polar(theta="x")+
  theme(plot.title = element_text(hjust = 0.5,family="myFont",size=18,color=
"gold3"),panel.background=element_rect(fill='aliceblue'))



###question2:直方图
binsize=diff(range(mydata0$funded_amnt))/17
mydata2<-
cut(mydata0$funded_amnt,breaks=c(0,2000*(1:15),max(mydata0$funded_amnt)))
p<-ggplot(mydata0,aes(x=funded_amnt))+geom_histogram(
  aes(y=..density..),binwidth=binsize,fill='slateblue4',color='white')+
  #stat_function(fun=dnorm,..)#统计函数，dnorm 正态分布曲线
  stat_function(fun=dnorm,args=list(mean(mydata0$funded_amnt),sd(mydata0$fun
ded_amnt)),size=1,color='orange')+
  #geom_density 时核密度估计曲线
  geom_line(stat='density',color='lightgreen',size=1)+expand_limits(y=0)+
  theme(plot.background=element_rect(fill="floralwhite",colour="black",size=
2),panel.background=element_rect(fill=NA),plot.title = element_text(hjust =
0.5,family="myFont",size=18,color="tomato1"))+
  labs(x='承诺贷款金额',y='频数',title='承诺贷款金额直方图')
   data3<-as.data.frame(ggplot_build(p)$data[1])
```

```
  p+geom_text(data = data3,aes(x=x, y= density,label = count),color='blue',vjust=-0.1)+
  annotate(geom="text",x=1e4,y=8e-5,label="密度最大点",size=5,colour="black")+
  annotate(geom="text",x=2e3,y=2e-5,label='568',size=3,colour="red")+
  annotate(geom="text",x=4e3,y=5.4e-5,label='1499',size=3,colour="red")+
  annotate(geom="text",x=6e3,y=6e-5,label='1651',size=3,colour="red")+
  annotate(geom="text",x=8e3,y=5.9e-5,label='1616',size=3,colour="red")+
  annotate(geom="text",x=1e4,y=7.5e-5,label='2033',size=3,colour="red")+
  annotate(geom="text",x=1.2e4,y=5e-5,label='1390',size=3,colour="red")+
  annotate(geom="text",x=1.4e4,y=4.7e-5,label='1311',size=3,colour="red")+
  annotate(geom="text",x=1.6e4,y=2e-5,label='572',size=3,colour="red")+
  annotate(geom="text",x=1.8e4,y=1.6e-5,label='496',size=3,colour="red")+
  annotate(geom="text",x=2e4,y=2.5e-5,label='717',size=3,colour="red")+
  annotate(geom="text",x=2.2e4,y=5e-6,label='178',size=3,colour="red")+
  annotate(geom="text",x=2.4e4,y=1.3e-5,label='408',size=3,colour="red")+
  annotate(geom="text",x=2.6e4,y=1e-6,label='86',size=3,colour="red")+
  annotate(geom="text",x=2.8e4,y=4e-6,label='168',size=3,colour="red")+
  annotate(geom="text",x=3e4,y=4e-6,label='160',size=3,colour="red")+
  annotate(geom="text",x=3.2e4,y=1e-6,label='60',size=3,colour="red")+
  annotate(geom="text",x=3.4e4,y=1e-5,label='314',size=3,colour="red")

###question3:密度图
#生成几何对象
p=ggplot(mydata0,aes(x=loan_amnt,y=log(annual_inc)))
#默认等高线图进行分箱化处理
p+geom_point(alpha=0.2)+stat_bin2d()+scale_fill_gradient(low="lightblue",high="red")+stat_density2d()+
  theme(plot.title = element_text(hjust = 0.5,family="myFont",size=18,color="slateblue2"),panel.background=element_rect(fill='papayawhip'))+
  labs(x='贷款金额',y='年收入',title='等高线分箱密度曲线图')

#将 height 映射给线条颜色
p+stat_density2d(aes(colour=..level..))+
  theme(plot.title = element_text(hjust = 0.5,family="myFont",size=18,color="slateblue2"),panel.background=element_rect(fill='papayawhip'))+
  annotate("rect",xmin=2500,xmax=12500,ymin=2.6e4,ymax=7e4,alpha=0.3,fill="red")+labs(x='贷款金额',y='年收入',title='贷款总额与年收入的相关密度图',color='水平')

###question4:热图
mydata3<-
  mydata0 %>%
  subset(select=c(loan_amnt,installment,annual_inc,dti,total_acc))
```

```r
#标准化处理
mydata3<-scale(mydata3)
#删除有缺失值的行
mydata3<-na.exclude(mydata3)
obsnum<-rownames(mydata3)
n<-sample(obsnum,50)
mydata3=mydata3[n,]
dis1<-dist(mydata3,method="euclidean")
dis1<-as.matrix(dis1)
heatmap(dis1,main="样本间距离热力图")
#美化热力图
heatmap.2(dis1,main="美化热力图",scale="column")

#画地图
data1<-read.csv("D:/大数据作业/大数据统计基础考试/2017《大数据统计基础》考试题/2017《大数据统计基础》考试题/province.csv",header=T)
data1<-data1[-c(1,2),]
colnames(data1)[1] <-'省市'
attach(data1)
map("china")
x=maptools::readShapePoly('D:/大数据作业/大数据统计基础考试/2017《大数据统计基础》考试题/2017《大数据统计基础》考试题/bou2_4p.shp')
plot(x)
provname<-c("北京市", "天津市", "河北省", "山西省", "内蒙古自治区",
            "辽宁省", "吉林省", "黑龙江省", "上海市", "江苏省",
            "浙江省", "安徽省", "福建省", "江西省", "山东省",
            "河南省", "湖北省", "湖南省", "广东省",
            "广西壮族自治区", "海南省", "重庆市", "四川省", "贵州省",
            "云南省", "西藏自治区", "陕西省", "甘肃省", "青海省",
            "宁夏回族自治区", "新疆维吾尔自治区")

pop<-data1$年末人口数
#整理数据
pop<-as.numeric(as.character(pop))
pop <-pop-min(pop)
getColor=function(mapdata,provname,provcol,othercol)
{
  f=function(x,y) ifelse(x %in% y,which(y==x),0);
  colIndex=sapply(mapdata@data$NAME,f,provname);
  col=c(othercol,provcol)[colIndex+1];
  return(col);
}

##构建图例位置
```

```
nf <- layout(matrix(c(1,1,1,1,1,2,1,1,1),3,3,byrow=TRUE), c(3,1), c(3,1), TRUE)
layout.show(nf)
provcol=rgb(red=1-pop/max(pop)/2,green=1-pop/max(pop)/2,blue=0)
plot(x,col=getColor(x,provname,provcol,"white"),xlab="",ylab="",main='2016年中国各省市人口分布状况')

# 添加图例
par(mar=c(0,0,0,0))
par(mar=c(1,1,2,0),cex=0.5)
barplot(as.matrix(rep(1,31)),col=sort(provcol,dec=T),horiz=T,axes=F,border = NA )
axis(1,seq(1,32,by=3),sort(pop[seq(1,32,by=3)]))

###再画地图
data1$省市<-provname
province <- data.frame(get_geo_position (provname))
str(province)
names(province)[3] <- c("NAME")
colnames(data1)[1] <-'NAME'
china_data_REmap <- join(data1,province,type="full",by="NAME")

china_map <- readShapePoly('D:/大数据作业/大数据统计基础考试/2017《大数据统计基础》考试题/2017《大数据统计基础》考试题/bou2_4p.shp')
china_map1 <- china_map@data
china_map1<-data.frame(china_map1,id=seq(0:924)-1)
china_map2 <- fortify(china_map)
china_map3 <- join(china_map2, china_map1, type="full",by="id")
china_map4 <- join(china_map3, data1, type="full",by="NAME")
china_data_REmap$地区生产总值<-as.numeric(as.character(china_data_REmap$地区生产总值))
china_data_REmap$年末人口数<-as.numeric(as.character(china_data_REmap$年末人口数))
china_map4$年末人口数<-as.numeric(as.character(china_map4$年末人口数))


theme_opts <- list(theme(panel.grid.minor = element_blank(),#设置网格线为空
                         panel.grid.major = element_blank(),#你可以去掉
                         panel.background = element_rect(fill="slategray1"),#设置图版背景色
                         plot.background = element_rect(fill="papayawhip"),#设置绘图区背景色
                         panel.border = element_blank(),
```

```
                          legend.background = element_rect(fill=rgb(red = 242, green = 242, blue = 242, max = 255)),
                          axis.line = element_blank(),
                          axis.text.x = element_blank(),
                          axis.text.y = element_blank(),
                          axis.ticks = element_blank(),
                          axis.title.x = element_blank(),
                          axis.title.y = element_blank(),#以上全是设置 xy 轴
                          plot.title = element_text(size=10)))
ggplot ()+
  geom_polygon(data=china_map4,aes(x=long,y=lat,group=group,fill=年末人口数),colour="gray40")+
  scale_fill_gradient(name="各省市人口分布",low="white",high="red")+
  geom_errorbar(data=china_data_REmap,aes(x=lon, ymin=lat, ymax=lat + 地区生产总值/7000 ),
                colour="gold",size=5, width=0,alpha=0.9)+
  geom_text(data =china_data_REmap,aes(x=lon,y=lat,label=NAME),colour="black",size=5,
            vjust=0,nudge_y=0.5)+
  labs(title ="各地区生产总值")+
  ylim (18, 54)+theme_opts
```

## 五、空间统计

### 1、 空间自相关原理及应用领域（10 分）

**（1） 空间自相关技术概念：**

空间自相关是指一个区域单位上的某种属性（如人口、发病率）与临近区域单位上的同一属性值之间的相关程度，其基本度量指标是空间自相关系数，用空间自相关系数来检验区域单位的某一属性值是否高高想邻、低低相邻或者高低间错分布，即有无聚集性。

**（2）空间自相关技术的原理：**

- **空间自相关的三种情况：**

一种是正相关，指邻近区域有相同或相似的属性值，如果某变量属性值在空间分布上呈现出高的地方周围也高，低的地方周围也低，称为空间正相关，表面变量属性值具有空间扩散性；另一种是负相关，指邻近区域有不同的属性值，如果在空间分布上呈现出高的地方周围低，低的地方周围高，则成为空间负相关，表明此变量属性具有空间极化特征；第三种是无相关，指变量属性值在空间分布上呈现出随机性，辨明空间自相关不明显，是一种随机分布现象。

- **空间自相关的实例类型：**

全域性空间自相关、局域性空间自相关；前者从整个研究范围分析某种属性值空间分布是否有聚集性，但不能确切指出聚集的地方；后者指在特定区域分析某种属性值空间分布有无聚集性，其分析结果可解释和探测存在空间聚集型的“热点”或“冷点”区域。

- **常用分析方法：**

Moran’s I 系数、 Geary’s C 系数、Getis 系数、Moran 散点图三个系数均分为全域性和局域性，适用点各不相同。全域性 Moran’s I 系数与全域性 Geary’s C 系数可用于检验某属性值书否出现空间聚集性；全域性 Getis 系数可用于检验当出现聚集性时，聚集性类型是什么，比如是高值聚集还是低值聚集；局域性的 Moran’s I、Geary’s C、Getis 系数可以用于确定具体的聚集区域在什么位置，热点与冷点

分布在哪里；Moran 散点图常用来研究局部的空间不稳定性，对区域单元的空间关联模式进行二维可视化展示。

**（3）空间自相关技术的应用**

- **在计量经济学中的应用：**

根据空间自相关定义,集聚经济是一种空间自相关现象,因而可以采用空间自相关方法度量集聚经济程度。而对地区区域经济发展水平及其差异的研究一直是各相关科学学者关注的热点,其研究范围和尺度也有较大的差异,从研究国家之间、地带之间、省际之间的差异变化到省区内部的区域经济差异变化,研究的尺度也从以国家、省为区域单元到以乡镇为区域单元。集聚经济的类型决定着区域经济的发展方向,传统的度量指标如基尼系数、Ellison-Glaeser 指数等,但是这类指标都存在着空间邻近概念的缺失,却没有考虑相邻地区间经济的相互影响,反映不够全面。因而该空间自相关技术不仅可以用图形示意区域集聚经济的类型,而且还可以用一些量化指标,结合区域经济增长极理论和中心-周围理论,解释区域的集聚经济的空间格局。

- **生态学研究中的应用**

早在五六十年代,就开始有人尝试用于生态学和进化研究,然而,在生物学领域广泛开始应用也才是近十几年的事情。Sokal&Oden 在将空间自相关分析引入生物学领域时就强调了它在生态学研究中的应用,众多研究者开展了对植物群落,鸟类迁徙,以及土壤土性等方面的研究。如比较成功的有加利福尼亚大学教授 WalterD.Koenig 对加利福尼亚州陆鸟间隔距离不等居住环境、大小、饮食习性,迁徙特点等的空间自相关分析。

- **居群遗传结构分析**

空间自相关分析技术可用来描述基因频率的地理变异,概况度量某一变量(如基因频率、基因型)在某一地理区域的数值依赖邻近区域同一变量数值的强弱,它是研究遗传变异空间结构的一种有效的方法,目前广泛用于研究居群内、居群间的基因频率和基因型的空间相关分析等。

- **在公共卫生流行病和传染病中的应用**

公共卫生对于人群疾病的研究和探讨,主要目的在于探讨人群疾病的时间、空间和人群分布,也就是流行病学中通常所说的“三间分布”。而空间数据的统计分析方法着重考虑了聚居、生存环境等地理空间信息的差异,相对于传统的经典的统计学方法优势更加明显,且应用更加广泛。地理要素在空间上表现出不同的区域差异,这些差异会引起环境状态的不同,而在医学领域,疾病的地理信息,逐渐受到重视,并进一步研究疾病分布的地区差异,而疾病在空间上表现出的各种空间差异外,疾病还具有发生、扩散、流行、人类干预和消除的时间延续过程。另外,疾病在时间、空间上的分布差异,而易受疾病感染的人群即高危人群的分布在空间上也具有差异性。

**2、 时空扫描统计原理及应用（10 分）**

**（1）时空扫描统计的概念：**

在无任何先验假设的情况下，对聚集性的位置和空间范围进行准确定位。时空扫描统计目的在于探测某属性值异常升高的时空区域，并检验这种升高是不是由随机变异造成的。

**（2）时空扫描的基本原理：**

其原假设$H_0$为：某属性值空间分布是完全随机的；备择假设$H_1$为：与扫描窗口外相比，属性值在窗口内的增加。空间扫描统计采用移动窗口法，在研究区域内建立活动圆形窗口对某属性值进行扫描统计。以疾病发生率为例，窗口的大小和位置处于动态改变之中，以避免认为选定研究区域和圆心位置造成选择偏倚。窗口的圆心在地图中延网格或地理单位中心变动，扫描半径按人后辖区范围划分，具体在 0 至总人口一定比例定值之间变动（一般为总人口 5%）。对于每次变动，将计算窗口内与计算窗口外区域之间的疾病发生率差异，采用对数似然比（LLR）进行检验。寻找所有位置、所以大小窗口中的最大对数似然比值，此处为最有可能存在聚集性的区域，也就是所有位置、所以大小窗口中的最不可能由随机变动造成的。最终选取 LLR 值最大窗口为高发病聚集窗口，确定此窗口所包括的地区，并计算该地区的相对危险度(RR)及检验有无统计学意义。

**（3） 时空扫描法的应用：**

在探讨疾病空间聚集性方法有简单的描述统计与聚集性检验。聚集性检验可以分为一般聚集性检验和焦点聚集性检验。全局聚集性检验的代表性方法可以用于检验整个研究区域内是否存在聚集性，但不对聚集性的具体位置进行定位；焦点聚集性检验用于检验某确定点周围是的有聚集性，因此无论是全局性还是焦点聚集检验，都只能单方面说明某研究区域存在聚焦点或聚集区的位置，而不能完整地对某聚集区内的位置和空间范围做一个精确定位。

因此，空间描述统计目前主要应用在检验疾病在某研究区域是否存在聚集性以及对聚集区的大小和位置进行定位，为疾病防治提供科学依据。