# 数据理解与准备

## 数据理解

主要是理解数据，包括分析抽样偏差，数据粒度，数据得精确含义，变量类型，冗余变量，完整性，缺省值，数据链接。这里主要先查看数据，处理冗余变量，进行缺失值的分析等。

## 获取并查看数据：

首先读取`bankloan.csv`文件中的数据为数据框`bankLoan`，并且使用R语言中的`summary(bankLoan)`初步查看数据的位置分布特征。

```
library(dplyr)
library(purrr)
library(xlsx)

setwd("D:/lagua/CODING/R-learn/R-code/Chap2-DataPreparation")

bankLoan = read.csv("bankloan.csv", header=TRUE,
                    fileEncoding ="GBK")

colnames(bankLoan) = c("Age", "Edu", "WorkAge",
                       "Address", "Income",
                       "DebtRatio", "CreditDebt",
                       "OtherDebt", "Default")
# ----------------数据空值检查-----------------
#----------列检查-----------
colnames(bankLoan)
length(colnames(bankLoan))

#----------行检查-----------
summary(bankLoan)
```

得到：

| | 年龄 | 教育 | 工龄 | 地址 | 收入 | 负债率 | 信用卡负债 | 其他负债 | 违约 |
|---|---|---|---|---|---|---|---|---|---|
| count | 700.000000 | 700.000000 | 700.000000 | 700.000000 | 700.000000 | 700.000000 | 700.000000 | 700.000000 | 700.000000 |
| mean | 34.860000 | 1.722857 | 8.388571 | 8.278571 | 45.601429 | 10.260571 | 1.553553 | 3.058209 | 0.261429 |
| std | 7.997342 | 0.928206 | 6.658039 | 6.824877 | 36.814226 | 6.827234 | 2.117197 | 3.287555 | 0.439727 |
| min | 20.000000 | 1.000000 | 0.000000 | 0.000000 | 14.000000 | 0.400000 | 0.011696 | 0.045584 | 0.000000 |
| 25% | 29.000000 | 1.000000 | 3.000000 | 3.000000 | 24.000000 | 5.000000 | 0.369059 | 1.044178 | 0.000000 |
| 50% | 34.000000 | 1.000000 | 7.000000 | 7.000000 | 34.000000 | 8.600000 | 0.854869 | 1.987568 | 0.000000 |
| 75% | 40.000000 | 2.000000 | 12.000000 | 12.000000 | 55.000000 | 14.125000 | 1.901955 | 3.923065 | 1.000000 |
| max | 56.000000 | 5.000000 | 31.000000 | 34.000000 | 446.000000 | 41.300000 | 20.561310 | 27.033600 | 1.000000 |

```
> summary(bankLoan)
      Age              Edu            WorkAge          Address
 Min.   :20.00   Min.   :1.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.:29.00   1st Qu.:1.000   1st Qu.: 3.000   1st Qu.: 3.000
 Median :34.00   Median :1.000   Median : 7.000   Median : 7.000
 Mean   :34.86   Mean   :1.723   Mean   : 8.389   Mean   : 8.279
 3rd Qu.:40.00   3rd Qu.:2.000   3rd Qu.:12.000   3rd Qu.:12.000
 Max.   :56.00   Max.   :5.000   Max.   :31.000   Max.   :34.000
     Income         DebtRatio       CreditDebt       OtherDebt
 Min.   : 14.0   Min.   : 0.40   Min.   : 0.010   Min.   : 0.050
 1st Qu.: 24.0   1st Qu.: 5.00   1st Qu.: 0.370   1st Qu.: 1.048
 Median : 34.0   Median : 8.60   Median : 0.855   Median : 1.985
 Mean   : 45.6   Mean   :10.26   Mean   : 1.553   Mean   : 3.058
 3rd Qu.: 55.0   3rd Qu.:14.12   3rd Qu.: 1.905   3rd Qu.: 3.928
 Max.   :446.0   Max.   :41.30   Max.   :20.560   Max.   :27.030
    Default
 Min.   :0.0000
 1st Qu.:0.0000
 Median :0.0000
 Mean   :0.2614
 3rd Qu.:1.0000
 Max.   :1.0000
```

---

## 冗余变量：

在对原始数据的初步分析之后，发现数据中不存在一些变量可以由其他的变量推导出来，也即没有冗余变量，所以在冗余变量处理部分，不作任何的处理。

---

## 缺失值：

如上图所示的summary(bankLoan)得到的结果所示，变量没有缺省值，即数据是完整的，故在这一步也不对数据进行任何的处理。

# 数据准备

清除变量，处理分类自变量，处理时间变量，异常值，极值，数据分箱，缺失数据，降维，欠采样与过抽样

根据生活的经验，所有的变量，年龄、教育、工龄、地址、收入、负债率、信用卡负债、其它负债等自变量都与是否违约有关。所以初步判定所有变量都是有效的，不用清除。

以下根据统计理论进行进一步地分析。

---

## 数据属性与描述：

### 处理分类型变量

分类自变量的类型，包括分类自变量和定序自变量。

关于分类自变量而言，对于属性变量，最常用的转换是将该自变量转变成哑变量；对于定序自变量，将该变量的序号转换成数值自变量。

经过以上`summary(bankLoan)`以及对原始数据的初步分析知，地址是属性变量，所以使用`as.factor`将变量的类型转变成因子类型，即**分类变量**；教育是定序变量，所以首先将其排序，并且转换成**整数型自变量**；而对于剩下的年龄、工龄、收入、信用卡负债、其他负债自变量，全部都转换成**数值型变量**，而其中的年龄、工龄需要转变成**整数型变量**。

使用R的实现代码如下：

```
# 调整数据框列的类型
bankLoan = bankLoan %>%
  mutate_at(.vars = vars("Address"),
            .fun = as.factor) %>%
  mutate_at(.vars = vars("Age", "WorkAge", "Income",
                         "DebtRatio","CreditDebt","OtherDebt"),
            .fun = as.numeric) %>%
  mutate_at(.vars = vars("Age", "Edu", "WorkAge"),
            .fun = as.integer)
```

### 查看数据描述

主要查看数据的分布，其中包括均值、标准差、最小值、下四分位数、中位数、上四分位数、最大值、缺失值的数量，得到：

```
> loan_nvars_description
           nmiss       mean        std   min      Q1 median      Q3    max
Age            0 34.8600000  7.9973422 20.00 29.0000 34.000 40.0000  56.00
Edu            0  1.7228571  0.9282055  1.00  1.0000  1.000  2.0000   5.00
WorkAge        0  8.3885714  6.6580390  0.00  3.0000  7.000 12.0000  31.00
Address        0  8.2785714  6.8248765  0.00  3.0000  7.000 12.0000  34.00
Income         0 45.6014286 36.8142264 14.00 24.0000 34.000 55.0000 446.00
DebtRatio      0 10.2605714  6.8272336  0.40  5.0000  8.600 14.1250  41.30
CreditDebt     0  1.5534571  2.1172091  0.01  0.3700  0.855  1.9050  20.56
OtherDebt      0  3.0582286  3.2875242  0.05  1.0475  1.985  3.9275  27.03
Default        0  0.2614286  0.4397271  0.00  0.0000  0.000  1.0000   1.00
```

# 异常值：

首先介绍一下**异常值的影响**，自变量中会因为很多的因素导致在收集数据的时候出现异常值，这些异常值的存在会影响我们的分析。尤其是在进行回归分析时，当出现一个距离大部分数据比较远的离群点。我们在进行模型拟合之前，如果不去除这些异常值的话，会将我们的回归模型往离群点处“拉”。所以在数据建模之前，在进行数据准备时，需要分析数据中异常值，以获得预测以及应用分析效果比较好的模型。

其次，对于**异常值的探测**，由于异常值一般是比较少的点，偏离主要的群体比较远的那些观测值。一般对于异常值的探测使用的方法根据维度一般分为以下几类：

1、一维

2、二维及以上

**在一维下**，对于异常值的检验主要有两种方法：**直方图法**和**箱线图法**。其中直方图法，首先将我们的一维变量数据分组，对所有的数据都排序之后分组，计算数据落到每个组的频数，就可以得到直方图。直方图可以显示出原始数据的分布。

直方图法：一般情况下，如果数据没有异常数据的话，数据一般是比较集中的，而异常的数据往往距离中间的数据比较远，从直观上来看，数据在左边或者右边会产生“拖尾”的现象，我们需要处理的是这些”两边“的数据。由于在大数定律下，数据的分布满足

$$p(|X-\mu|<2\sigma)=95.44\%$$

而且假设检验中的$\alpha$的一般取值为0.05，所以控制正常数据落到$2\sigma$的范围之内是合理的。由此原理，我们将$2\sigma$之外的数据看作异常值。

箱线图法：箱线图主要是使用分位数来确定数据中是否有异常值。其中有50%的数据在箱子中，即(Q1,Q3)中，当数据超过上界$Q_3+1.5\Delta Q$和下界$Q_3-1.5\Delta Q$之后，使用圈来表示，这是异常值。

对于**多维数据**，分别有距离法、分类预测模型（聚类）、基于树模型法、基于密度法来识别数据。

距离法：当$d>3$或一个阈值之后，就认为这些数据是异常值。注：距离一般使用的是欧氏距离。

分类预测模型法：k-means，在最小化误差函数的基础上将数据划分为预定的类数$K$. 在聚类完成之后，计算一个类中每个点到类的族中心的距离，如果超过某个阈值，就认为这个点是异常值点。

其次，是对**异常值的处理。**

主要有四种方法：

1. 删除含有异常值的记录
2. 将异常值视为缺失值，最后跟缺失值一起处理。
3. 使用平均值来修正
4. 不处理

根据下面的直方图来看，数据的分布呈现出极值分布，对于年龄，没有异常值；对于教育分布，虽然在右边有一个拖尾的数据，但是这并不是异常的数据，因为在显示中贷款的人群中是存在少部分高学历人群的，这是正常的现象，并不是异常数据，故不对其进行处理；同上，我也认为工龄的分布右边的拖尾是正常的；同理，收入、负债率、信用卡负债、其他负债的右边拖尾数据的存在是非常正常的。又因为在实际中存在有这种看似”异常“的数据，我们可以更加关注他们，对于是否给予他们贷款，需要**具体情况具体分析**，这就是不是仅仅剔除异常值，或者使用上面包含的四种方法的任意一种方法来处理的。在这些异常值中，我们需要**单独拿出这些更加细致地根据具体情况分析，而不是在数据挖掘的初期就剔除这些异常值。**

综上，在异常值处理部分，先不对数据进行预处理。

---

# 极值处理：

**极值化：**实际数据中自变量或因变量的分布会呈现出偏斜有极值的现象，例如下图所示的图4 收入就有明显的极值分布。这回对一些模型产生很大的影响，所以我们需要对这些自变量进行极值化处理。

极值化处理一般有两种方法：

- Box-Cox转化
- 将具有极值化的变量转换为秩，然后再进行分组。

其中Box-Cox变换是指对变量$u$进行：

$$z=\begin{cases}\dfrac{(u+r)^{\lambda-1}}{\lambda} & \lambda\neq 0\\ log(u+r) & \lambda=0\end{cases}$$

$r$是一个常数，对$u$的所有可能取值都满足$u+r>0$，对数转换时Box-Cox转换的一种特殊情形。

首先对本案例数据的所有变量画图，得到：

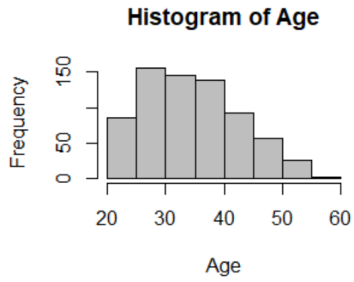


图1 年龄

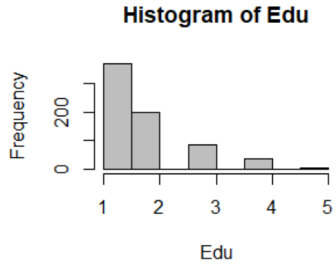


图2 教育

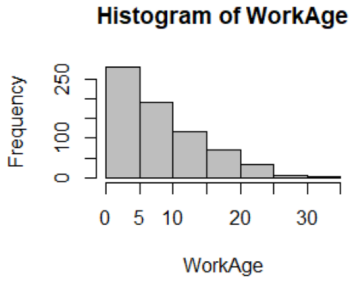


图3 工龄

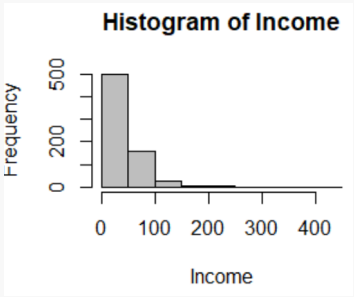


图4 收入

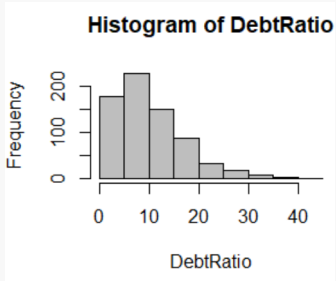


图5 负债率

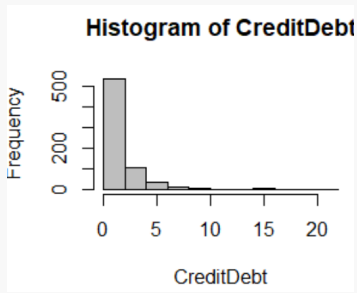


图6 信用卡负债

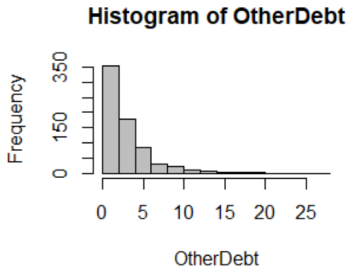


图7 其他负债

如图所示，在经过数据筛选得到整数和数值型的数据中，教育、工龄、收入、信用卡负债、其他负债存在有极值分布，这里仅仅只针对**数值型**的数据收入、信用卡负债、其他负债变换来进行极值处理，但是对**整数型变量**教育、工龄不进行极致化处理。因为如果进行极值化处理，这些整数型数据就不再是整数了，这在数据分析中是不合理的，因为这改变了数据的变量类型。

这里我们首先对数据收入、信用卡负债、其他负债进行简单的数据变换，即$z = log(u+1)$，得到：

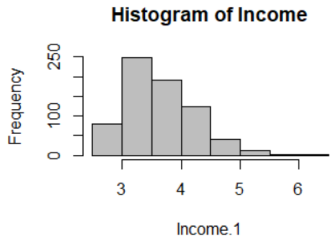


图1 收入.1

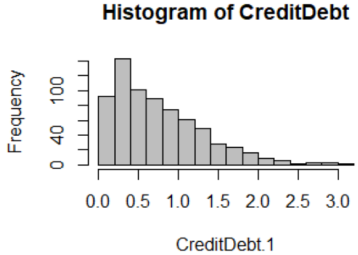


图2 信用卡负债.1

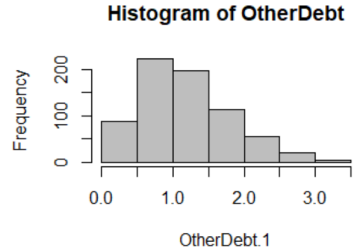


图3 其他负债.1

上图的图1-图3看出，数据的分布不再是极值分布了，至此，极值处理完毕。

## 缺失数据：

根据上一张图片显示的对lona.nvars的描述可知，本案例数据中不存在缺失值。故不对缺失数据进行处理。

---

## 降维：

冗余变量不做处理，因为这里从summary(bankLoan)就已经知道所有的变量的中没有不变化的变量。

## 变量选择：

1. 针对因变量为**二分变量**

（1）对于数值自变量而言，可以使用两样本**t检验**考察因变量取一种值时，与因变量取另外一种值时，该自变量的均值是否相等，然后选择哪些检验**结果显著**（不相等的）自变量。

（2）对于分类型自变量而言，可以使用**卡方检验**考察自变量的取值是否独立于因变量的取值，然后选择那些**结果显著**（不独立）的自变量。

注：以上检验显著，说明不同的自变量对因变量的取值是有显著影响的，这就说明这些自变量会影响因变量的取值，故不进行特征筛选。

2. 因变量为分类变量

   可以将因变量取值两两配对，针对每对取值进行上述**t检验或者卡方检验**，然后选择那些对因变量的**任何一对**取值检验**结果显著**的自变量。

3. 因变量为数值型变量

   可选择将因变量取值离散化之后，再使用上面的方法，或者使用以下：

   （1）计算个数值自变量与因变量的相关系数，**剔除相关系数小或不显著的变量。**

   （2）对每个分类自变量，将其取值两两配对，针对每个取值，用**t检验**考察因变量的均值是否相等，只要对任何一对取值检验**结果显著**，就选择该自变量。

关于变量的选择，还有逐步选择的方法，包括向前选择，向后剔除，向前选择与向后剔除的结合。

**针对本案例数据进行分析：**

本案例数据情况：因变量违约，它只有两个取值0和1，其中1代表违约，0代表正常数据。由此可知，本例因变量是**二分变量**，所以针对以上提及到的第一种情况进行处理。

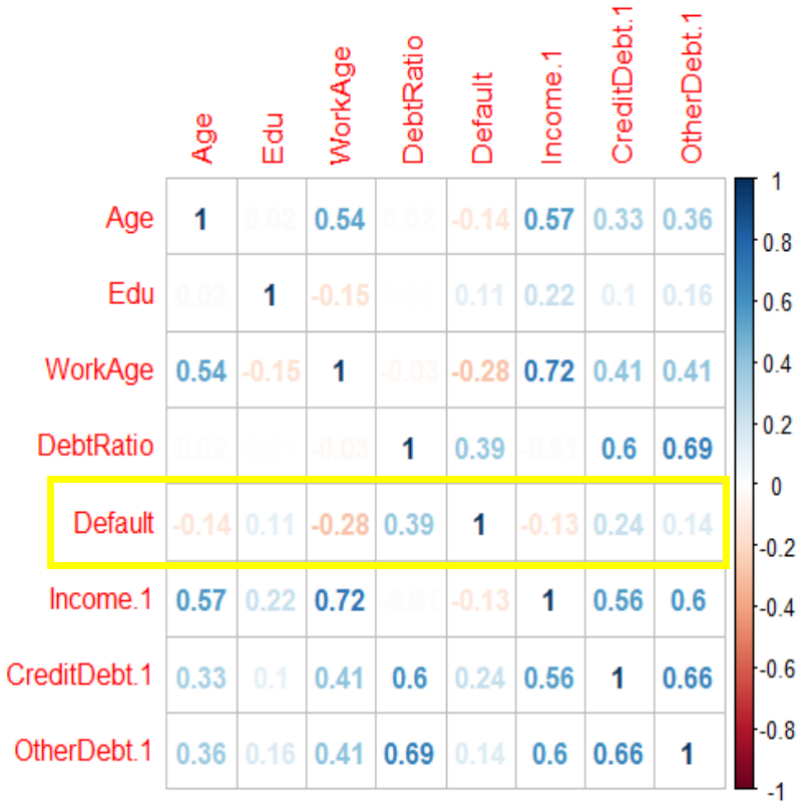
对分类型自变量Address进行卡方检验，得到

```
> chisq.test(table(loanNew$Address, loanNew$Default))

        Pearson's Chi-squared test

data:  table(loanNew$Address, loanNew$Default)
X-squared = 65.001, df = 30, p-value = 0.0002193
```

其中p值是小于0.01的，所以拒绝原假设，认为Address的取值对Default的取值有显著影响。




因为数据过多，我们不可能对所有的自变量都计算对所有因变量Default来进行检验，所以我们首先选出与因变量Default相关性比较小的变量来进行t检验。

如上图所示，相关系数绝对值最小的前四个：年龄、教育、变换后的收入、变换后的其他负债。

分别对其检验，得到结果：

```
> t.test(loanNew$Age ~ loanNew$Default)

        Welch Two Sample t-test

data:  loanNew$Age by loanNew$Default
t = 3.5011, df = 294.08, p-value = 0.0005353
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.096232 3.910923
sample estimates:
mean in group 0 mean in group 1
       35.51451        33.01093
```

如图所示，`t.test(loanNew$Age ~ loanNew$Default)`显示变量关于Default的两个取值是显著的。所以不考虑剔除此变量。

```
> t.test(loanNew$Edu ~ loanNew$Default)

        Welch Two Sample t-test

data:  loanNew$Edu by loanNew$Default
t = -2.9456, df = 300.49, p-value = 0.003475
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.40378330 -0.08034646
sample estimates:
mean in group 0 mean in group 1
       1.659574        1.901639
```

如图所示，`t.test(loanNew$Edu ~ loanNew$Default)`也是显著的，所以不考虑剔除此变量。

```
> t.test(loanNew$Income.1 ~ loanNew$Default)

        Welch Two Sample t-test

data:  loanNew$Income.1 by loanNew$Default
t = 3.5225, df = 310.2, p-value = 0.0004917
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.07700002 0.27188242
sample estimates:
mean in group 0 mean in group 1
       3.698698        3.524257
```

如图所示，`t.test(loanNew$Income.1 ~ loanNew$Default)`也是显著的，所以不考虑剔除此变量。

```
> t.test(loanNew$OtherDebt.1 ~ loanNew$Default)

        Welch Two Sample t-test

data:  loanNew$OtherDebt.1 by loanNew$Default
t = -3.6003, df = 291.97, p-value = 0.0003734
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.31109109 -0.09118377
sample estimates:
mean in group 0 mean in group 1
       1.129340        1.330477
```

如图所示，`t.test(loanNew$OtherDebt.1 ~ loanNew$Default)`也是显著的，所以不考虑剔除此变量。

综上，**不剔除任何的变量**，认为所有的自变量都对因变量Default的取值有影响。

---

## 欠采样与过采样

观察数据：

```
> dim(loanNew[loanNew$Default==1, ])
[1] 183   9
> dim(loanNew[loanNew$Default==0, ])
[1] 517   9
```

其中响应变量占比比较少，因此对其进行过采样处理，以使得**相应变量与非相应变量**之间的比为**1：2**.

# 参考


R语言绘制热图（其实是相关系数图）实践(二) corrplot包

数据预处理之异常值处理

数据清洗中异常值（离群值）的判别和处理方法


# 附录

```
library(dplyr)
library(purrr)
library(xlsx)


# ----------读取数据----------
# Sys.setlocale("LC_ALL","Chinese")
```

```r
setwd("D:/lagua/CODING/R-learn/R-code/Chap2-DataPreparation")

bankLoan = read.csv("bankloan.csv", header=TRUE,
                    fileEncoding ="GBK")


# ----------数据观察：数据空值检查----------
# ----------列检查----------
# 注意：需要检查一下数据是否有列是NA，
# 使用：以下查看
colnames(bankLoan)
length(colnames(bankLoan))
# 如果是，
# 1，手动将看到的最后几列（这里是3列）删除，无论是否看见有数据
# 2. 使用na.omit()直接将所有的具有空的行或者列删除。有风险！！！
# 参考：[R语言 -- 删除 dataFrame/matrix 中含有NA或全为NA的行或列]
# (https://www.jianshu.com/p/26edb1b1e6c7)

# ----------行检查----------
# 行中是否有空的值也需要查看，使用summary(bankLoan)查看
colnames(bankLoan) = c("Age", "Edu", "WorkAge",
                       "Address", "Income",
                       "DebtRatio", "CreditDebt",
                       "OtherDebt", "Default")

# ----------列检查----------
summary(bankLoan)

# 调整数据框列的类型
bankLoan = bankLoan %>%
  mutate_at(.vars = vars("Address"),
            .fun = as.factor) %>%
  mutate_at(.vars = vars("Age", "WorkAge", "Income",
                         "DebtRatio","CreditDebt","OtherDebt"),
            .fun = as.numeric) %>%
  mutate_at(.vars = vars("Age", "Edu", "WorkAge"),
            .fun = as.integer)


# ----------查看数据----------
# 找出列变量的所有整数型或者数值型的数据
loan.nvars <- bankLoan[,lapply(bankLoan,class)=="integer"
                       | lapply(bankLoan,class)=="numeric"]
summary(loan.nvars)


# ----------查看数据描述----------
descrip <- function(nvar)
{
  nmiss <- length(which(is.na(nvar)))
  mean <- mean(nvar,na.rm=TRUE)
  std <- sd(nvar,na.rm=TRUE)
  min <- min(nvar,na.rm=TRUE)
  Q1 <- quantile(nvar,0.25,na.rm=TRUE)
  median <- median(nvar,na.rm=TRUE)
  Q3 <- quantile(nvar,0.75,na.rm=TRUE)
  max <- max(nvar,na.rm=TRUE)
  return(c(nmiss,mean,std,min,Q1,median,Q3,max))
```

```
}

loan_nvars_description <- lapply(loan.nvars,descrip) %>%
  as.data.frame() %>% t()

colnames(loan_nvars_description) <- c("nmiss","mean","std","min","Q1",
                                      "median","Q3","max")
loan_nvars_description


# ----------异常值检测----------
library(vioplot)
# 主要使用直方图来查看异常值的大致分布
for (col in colnames(bankLoan)[-c(4, 9)]){
  # vioplot(bankLoan[[col]], ylab=col)
  hist(bankLoan[[col]], xlab=col,
       main=paste("Histogram of ", col, sep=""))
}

# a = cut(bankLoan$Age, breaks = seq(10, 60, 10)) %>% unique()
# hist(cut(bankLoan$Age, breaks = seq(10, 60, 10)))


# ----------极值处理：（观察）数值变量直方图输出----------
# ----------查看极值----------
library(showtext)
library(sysfonts)
library(showtextdb)
font_add("SIMHEI","SIMHEI.ttf")
font_add("SIMSUN","SIMSUN.TTC")
font_add("kaishu","simkai.ttf")
par(family='STKaiti')
# 设置family='GB1'
pdf("./ch2_case2-2_histogram.pdf",family='GB1')

par(c(3, 3))
for (i in 1:length(loan.nvars)){
  hist(loan.nvars[,i],
       xlab=names(loan.nvars)[i],
       main=paste("Histogram of",names(loan.nvars)[i]),
       col = "grey")
}
dev.off()


# ----------极值处理----------
colnames(bankLoan)
colnames(loan.nvars)

library(MASS)
par(c(1, 3))
# log.vars = c("Income", "CreditRatio", "OtherDebt") # 需要处理的极值
log.vars = c(4, 6, 7)
# new_col = list()
for (i in log.vars){
  var.names = colnames(loan.nvars)
  cur_name = paste(var.names[i],  '.', 1, sep = '')
  new_col = log(loan.nvars[, i] + 1)
```

```
  loan.nvars[[cur_name]] = new_col
  hist(loan.nvars[[cur_name]],
       xlab=paste(var.names[i],  '.', 1, sep = ''),
       main=paste("Histogram of",cur_name),
       col = "grey")
}

loanNew = cbind(bankLoan, loan.nvars[,9:11])
loanNew = loanNew[, -c(5, 7, 8)]
colnames(loanNew)


# ----------数据分箱----------
# TODO


# ----------变量选择----------
##删除冗余变量,生成新数据集loanNew
# > colnames(loanNew)
# [1] "Age"           "Edu"           "WorkAge"      "Address"
# [5] "Income"        "DebtRatio"     "CreditDebt"   "OtherDebt"
# [9] "Default"       "Income.1"      "CreditDebt.1" "OtherDebt.1"
# ----------第一步：数值型变量筛选：卡方检验----------
chisq.test(table(loanNew$Address, loanNew$Default))


# ----------变量筛选第一步：画相关系数图----------
cor(loanNew[,-c(4)])
cor.test(loanNew[,-c(4)])

library(corrplot)
corrplot(corr=cor(loanNew[,-c(4)]))
corrplot(corr=cor(loanNew[,-c(4)]), method="number")

# ----------变量筛选第二步：进行t检验----------
test_vars = c("Age", "Edu", "Income.1", "OtherDebt.1")
t.test(loanNew$Age ~ loanNew$Default)
t.test(loanNew$Edu ~ loanNew$Default)
t.test(loanNew$Income.1 ~ loanNew$Default)
t.test(loanNew$OtherDebt.1 ~ loanNew$Default)


# ----------欠采样与过采样----------
##进行欠抽样使得响应者的比例达到1/3
loan1 <- loanNew[loanNew$Default==1,]
loan0 <- loanNew[loanNew$Default==0,]
n1 <- dim(loan1)[1]
n0 <- 2*n1
# 响应观测数是非响应观测数的1/2
loan0 <- loan0[sample(1:dim(loan0)[1],n0),]
loanNew1 <- rbind(loan1,loan0)


# ----------存储数据----------
write.csv(loan_nvars_description,"static/loan_nvars_description.csv")
write.csv(loanNew,"static/loanNew.csv",row.names=FALSE)
write.csv(loanNew1,"static/loanNew1.csv",row.names=FALSE)
```