# Первичная обработка данных

Создаем учебные данные

numPeople = 10

sex=sample(c("male","female"),numPeople,replace=T)

age = sample(14:102, numPeople, replace=T)

income = sample(20:150, numPeople, replace=T)

minor = age < 18

population = data.frame(sex, age, income, minor)

## Обработка пропущенных и неверных значений

# Создание модельноо объекта

x <- (1:5)^2

x[6] <- 1/0

x[7] <- sqrt(-1)

x[10]<- 0

Результат:

> x

[1] 1 4 9 16 25 Inf NaN NA NA 0

Сделайте указанные проверки. Можно ли с их помощью выделить неверные/пропущенные значения?

x == 4

x == Inf

x == NA

x == NaN

Для проверки используются специальные функции R:

nana <- c(NA, NaN, 0/0, 1/0)

is.na(nana)

is.nan(nana)

is.infinite(nana)

x[is.na(x)]

# Более продвинутый вариант убирания пропущенных значений:

na.omit(x)

Работа с таблицей

population -> pop1

pop1[2,3]<-NA

pop1[4,2]<-NA

pop1

pop1[!is.na(pop1$sex),] -> pop2

pop2[!is.na(pop2$age),] -> pop3

pop3[!is.na(pop3$income),] -> pop4

pop4[!is.na(pop4$minor),] -> pop5

pop5

# то же в цикле

for (name in names(pop1)){

pop1[!is.na(pop1[[name]]),] -> pop1

}

# то же с помощью готовых функций из R

pop1[complete.cases(pop1), ]

na.omit(pop1) # содержит информацию о пропущенных строках

na.action(na.omit(pop1)) -> na.pop

pop1[na.pop,] # "испорченные" строки

Как найти число пропущенных значений? Долю строк с пропущенными значениями?

sum(is.na(pop1))

mean(!complete.cases(pop1))

## Исследование структуры пропущенных значений

# создаем таблицу с большим количеством пропусков

numPeople = 100

sex=sample(c(rep(c("male","female"),4),NA),numPeople,replace=T)

age = sample(18:90, numPeople, replace=T)

age[age > 85] <-NA

income = sample(seq(5:25)\*10, numPeople, replace=T)

income[income>200]<-NA

income[is.na(sex)]<-c(NA,80,NA,100)

population = data.frame(sex, age, income)

summary(population)

#-------------------------

mean(complete.cases(population)) # доля объектов с полной информацией

pop\_bool <- as.data.frame(abs(is.na(population))) # 1 – пропущено, 0 -- нет

table(apply(pop\_bool,1,sum)) # число объектов по числу пропусков

na\_pop <- pop\_bool[apply(pop\_bool,2,sd) > 0] # (частично) неполные столбцы

cor(na\_pop)

## Замещение пропущенных значений

Средними

xx <- x[x != Inf]

xx[is.na(xx)] <- mean(xx[!is.na(xx)])

# другой способ записи того же для mean( )

xx[is.na(xx)] <- mean(xx, na.rm = TRUE)

Случайными (бутстрэп)

y <- x

y[is.na(x)]<-sample(x[!is.na(x)],sum(is.na(x)), rep=TRUE)

# то же через na.omit

y <- x

na.action(na.omit(x)) -> na.x

y[na.x] <- sample(na.omit(x),length(na.x), rep = TRUE)

Больше возможностей дают пакеты (см. Р.Кабаков, "R в действии", глава 15)

## Сортировка и упорядочение

# упорядочиваем возраст

age

sort(age)

order(age)

# упорядочиваем всю таблицу по возрасту

population[order(population$age), ]

# упорядочиваем всю таблицу по полу и возрасту

population[order(population$sex, population$age), ]

# неверная запись

population[order(population[,1:2]), ]

## Визуализация данных с помощью метода главных компонент

Если объект многомерный, его трудно изобразить в двумерной плоскости. Можно использовать попарные диаграммы рассеяния. Но можно также "повернуть" множество точек в Rn так, чтобы сохранить как можно больше информации о взаимном расположении точек.

iris[1:4]->numIris

plot(numIris, col=iris[,5])



prcomp(numIris)$rotation ->rot

numIris%\*%rot # ошибочно

rot%\*%as.matrix(numIris) # ошибочно

as.matrix(numIris)%\*%rot -> newIris # верно

plot(newIris, col=iris[,5])

 Подсчитаем долю дисперсии, описываемую каждой компонентой

iris.pca <- princomp(numIris)

# biplot(iris.pca,cex=0.6)

plot(iris.pca)

text(1:4,iris.pca$sdev^2,labels=format(iris.pca$sdev^2,dig=2))

Если отдельные параметры измерены в разных единицах измерения (или имеют неодинаковый размах/дисперсию), то можно предварительно пронормировать данные с помощью функции scale( ). Оформим вычисления в виде функции.

povorot <- function(data) {

as.matrix(data)%\*%prcomp(data)$rotation -> newD

data.pca <- princomp(data)

# biplot(iris.pca,cex=0.6)

par(mfrow=c(1,2)) ->old.p

plot(newD, col=iris[,5], asp = 1)

plot(data.pca)

text(1:4,data.pca$sdev^2,labels=format(data.pca$sdev^2,dig=2))

par(old.p)

}

povorot(numIris)

povorot(scale(numIris))