Краткое введение в мир матричной алгебры

Анализ и визуализация многомерных данных с использованием R

Вадим Хайтов, Марина Варфоломеева

Вы сможете

- Объяснить что такое матрицы и какие бывают их основные разновидности
- Выполнить базовые операции с матрицами с использованием функций R
- Применить в среде R методы матричной алгебры для решения простейших задач

Зачем нужны матрицы?

Матричные объекты

- Есть много типов объектов, для которых такое выражение оказывается наиболее етсественным (изображения, описания многомерных объектов и т.д.)
- В матрицах, как и в обычных числах, скрыта информация, которую можно извлекать и преобразовывать по определнным правилам

Структура матриц

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1c} \\ a_{21} & a_{22} & \cdots & a_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1} & a_{r2} & \cdots & a_{rc} \end{pmatrix}$$

Размер (порядок) матрицы $r \times c$

Разновидности матриц

Вектор-строка (row matrix)

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$$

Вектор-столбец (column matrix)

$$\mathbf{B} = \begin{pmatrix} 1 \\ 4 \\ 7 \\ 10 \end{pmatrix}$$

Разновидности матриц

Прямоугольные матрицы (rectangular matrices)

$$\mathbf{C} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$

$$\mathbf{D} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

В таком виде обычно представляются исходные данные

Квадратные матрицы (square matrices)

Это наиболее "операбельные" матрицы

$$\mathbf{E} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Диагональные матрицы (diagonal matrix)

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Квадратные матрицы (square matrices)

Треугольные матрицы (triangular matrices)

$$\mathbf{H} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 5 & 6 & 7 \\ 0 & 0 & 9 & 10 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

или

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 5 & 0 & 0 \\ 4 & 7 & 9 & 0 \\ 5 & 8 & 10 & 11 \end{pmatrix}$$

Квадратные матрицы (square matrices)

Единичная матрица (identity matrix)

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Единичная матрица (обозначение $\mathbf{I}$) занимают особое место в матричной алгебре.
Она выполняет ту же роль, которую выполняет единица в обычной алгебре.

Матрицы ассоциации

Изначально результаты исследования имеют вид исходной матрицы (обычно прямоугольной)

$$\mathbf{Y} = [n_{objects} \times p_{descriptors}]$$

Информация из этой матрицы конденсируется в двух других матрицах

Q анализ

$$\mathbf{A}_{nn} = [n_{objects} \times n_{objects}]$$

R анализ

$$\mathbf{A}_{pp} = [p_{descriptors} \times p_{descriptors}]$$

Матрицы ассоциации

Это симметричные квадратные матрицы

$$\mathbf{A}_{pp} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{pmatrix}$$

В этой матрице $a_{ij} = a_{ji}$

Большинство многомерных методов имеет дело именно с такими матрицами

Особенность квадратных матриц

Для квадратных матриц могут быть найдены (но не обязательно существуют) некоторые важные для матричной алгебры показатели: *определитель*, *инверсия*, *собственные значения* и *собственные вектора*

Задание

Создайте с помощью R следующие матрицы

```
##      [,1] [,2] [,3]
## [1,]    1    5    9
## [2,]    2    6   10
## [3,]    3    7   11
## [4,]    4    8   12
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
```

Операции с матрицами

Транспонирование матриц

```
A <- matrix(1:12, ncol = 3)
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    9
## [2,]    2    6   10
## [3,]    3    7   11
## [4,]    4    8   12
```

Транспонированная матрица $\mathbf{A}$

```
B <- t(A)
B
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    6    7    8
## [3,]    9   10   11   12
```

Сложение матриц

```
A + 4
```

```
##      [,1] [,2] [,3]
## [1,]    5    9   13
## [2,]    6   10   14
## [3,]    7   11   15
## [4,]    8   12   16
```

```
A + A
```

```
##      [,1] [,2] [,3]
## [1,]    2   10   18
## [2,]    4   12   20
## [3,]    6   14   22
## [4,]    8   16   24
```

Но! Нельзя складывать матрицы разных размеров

```
A + B
```

Биологическое приложение

Предположим, что мы подсчитывали двумя разными методами крупных и мелких животных трех видов в одних и тех же пробах

```
##         Sp1 Sp2 Sp3
## Sample1   8  10   9
## Sample2   9   7  12
## Sample3  10   9  12
## Sample4  10  10  11
## Sample5  11  11   8
```

```
##         Sp1 Sp2 Sp3
## Sample1  55  49  58
## Sample2  49  47  49
## Sample3  44  49  54
## Sample4  43  49  56
## Sample5  57  55  57
```

Общее обилие

```
Large + Small
```

```
##         Sp1 Sp2 Sp3
## Sample1  63  59  67
## Sample2  58  54  61
## Sample3  54  58  66
## Sample4  53  59  67
## Sample5  68  66  65
```

Простое умножение

Умножение на число

```
A * 4
```

```
##      [,1] [,2] [,3]
## [1,]    4   20   36
## [2,]    8   24   40
## [3,]   12   28   44
## [4,]   16   32   48
```

Простое умножение матрицы на вектор возможно только если число элементов в векторе равно числу строк в матрице

```
A * c(10, 11, 12, 13)
```

```
##      [,1] [,2] [,3]
## [1,]   10   50   90
## [2,]   22   66  110
## [3,]   36   84  132
## [4,]   52  104  156
```

Все элементы первой строки матрицы умножаются на первый элемент вектора, все элементы второй строки на второй элемент вектора и т.д.

Биологическое применение

Допустим, учет организмов в части описаний проходил не на всей выборке, а лишь в ее части.

```
Rpocessed_portion <- c(1, 1, 1/2, 1/3, 1/4)
Processed_Factor <- 1/Rpocessed_portion
```

```
Small * Processed_Factor
```

```
##         Sp1 Sp2 Sp3
## Sample1  55  49  58
## Sample2  49  47  49
## Sample3  88  98 108
## Sample4 129 147 168
## Sample5 228 220 228
```

Длина вектора

Длина вектора, или норма вектора

$$\|\mathbf{b}\| = \sqrt{b_1^2 + b_2^2 + \cdots + b_n^2}$$

Нормализованный вектор

$$\mathbf{c} = \mathbf{b} \cdot 1/\|\mathbf{b}\|$$

Задание

Найдите нормализованный вектор для следующего вектора

```
Vec <- 1:5
Vec
```

```
## [1] 1 2 3 4 5
```

Решение

```
Vec/sqrt(sum(Vec^2))
```

```
## [1] 0.135 0.270 0.405 0.539 0.674
```

```
Vec/norm(t(Vec), type = "F")
```

```
## [1] 0.135 0.270 0.405 0.539 0.674
```

Скалярное произведение векторов

Допустимо только для векторов одинаковой размерности

$$\mathbf{a} \bullet \mathbf{b} = \begin{pmatrix} a_1 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} \times \begin{pmatrix} b_1 & b_3 & b_4 & b_5 & b_6 & b_7 \end{pmatrix} = x$$

Результат этой операции - число (скаляр)

Биологическое применение

Сколько особей родится в популяции, если мы знаем репродуктивные характеристики всех возрастных групп?

$$\begin{pmatrix} N_1 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \\ N_7 \end{pmatrix} \times \begin{pmatrix} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & F_7 \end{pmatrix}$$

```
N <- c(20, 40, 32, 45, 80, 50, 10)
Fert <- c( 0,  0,   1,   2,   2,   0,   0)

t(N) %*% (Fert)
```

```
##      [,1]
## [1,]  282
```

Умножение матриц

Умножать можно только в том случае, если число строк одной матрицы равно числу столбцов другой матрицы

```
A %*% B
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  107  122  137  152
## [2,]  122  140  158  176
## [3,]  137  158  179  200
## [4,]  152  176  200  224
```

```
A %*% t(A)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  107  122  137  152
## [2,]  122  140  158  176
## [3,]  137  158  179  200
## [4,]  152  176  200  224
```

НО! Нельзя произвести такое умножение

```
A %*% A
```

Биологическое применение

Простейший пример использования умножения матрц - построение модели динамики демографической структуры популяции Для вычислений необходим начальный *демографический вектор* и *матрица Лесли*

$$\begin{pmatrix} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & F_7 \\ P_{1-2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & P_{2-3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & P_{3-4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & P_{4-5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P_{5-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & P_{6-7} & 0 \end{pmatrix} \times \begin{pmatrix} N1_t \\ N3_t \\ N4_t \\ N5_t \\ N6_t \\ N7_t \end{pmatrix} = \begin{pmatrix} N1_{t+1} \\ N3_{t+1} \\ N4_{t+1} \\ N5_{t+1} \\ N6_{t+1} \\ N7_{t+1} \end{pmatrix}$$

Простейшая демографическая модель

Демографический вектор в момент времени t

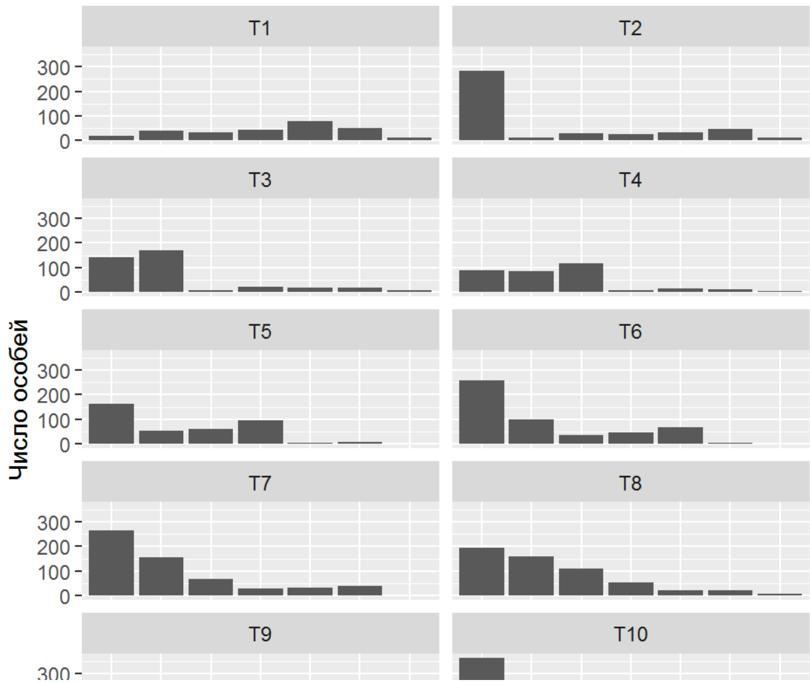
```
##     Age T1
## 1     0 20
## 2  1-10 40
## 3 11-20 32
## 4 21-35 45
## 5 36-45 80
## 6 46-55 50
## 7 56-65 10
```

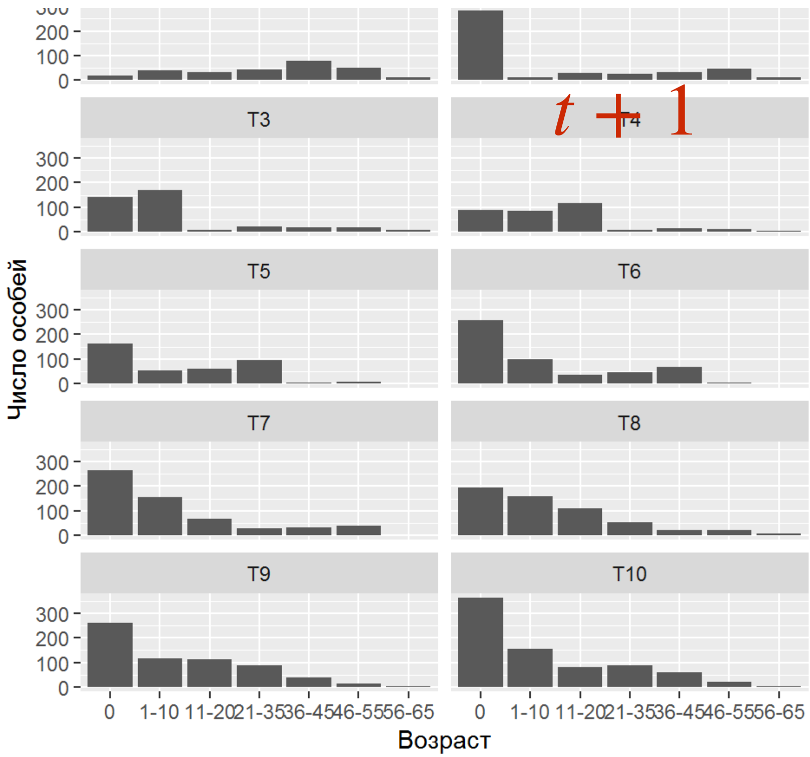
Матрица Лесли

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,]  0.0  0.0  1.0  2.0  2.0  0.0    0
## [2,]  0.6  0.0  0.0  0.0  0.0  0.0    0
## [3,]  0.0  0.7  0.0  0.0  0.0  0.0    0
## [4,]  0.0  0.0  0.8  0.0  0.0  0.0    0
## [5,]  0.0  0.0  0.0  0.7  0.0  0.0    0
## [6,]  0.0  0.0  0.0  0.0  0.6  0.0    0
## [7,]  0.0  0.0  0.0  0.0  0.0  0.2    0
```

Демографическая струкутра

```
Pop$T2 <- as.vector( Lesl %*% (Pop$T1 ))
Pop$T3 <- as.vector( Lesl %*% (Pop$T2 ))
Pop$T4 <- as.vector( Lesl %*% (Pop$T3 ))
Pop$T5 <- as.vector( Lesl %*% (Pop$T4 ))
Pop$T6 <- as.vector( Lesl %*% (Pop$T5 ))
Pop$T7 <- as.vector( Lesl %*% (Pop$T6 ))
Pop$T8 <- as.vector( Lesl %*% (Pop$T7 ))
Pop$T9 <- as.vector( Lesl %*% (Pop$T8 ))
Pop$T10 <- as.vector( Lesl %*% (Pop$T9 ))
```





Некоторые свойства произведения матриц

1. Если существует произведение матриц $\mathbf{BC}$, то не обязательно существует $\mathbf{CB}$

```
B <- matrix(1:24, ncol = 4)
C <- matrix(1:12, ncol = 3)
```

```
B %*% C
```

```
##      [,1] [,2] [,3]
## [1,]  130  290  450
## [2,]  140  316  492
## [3,]  150  342  534
## [4,]  160  368  576
## [5,]  170  394  618
## [6,]  180  420  660
```

НО!

```
C %*% B
```

Такое произведение невозможно

Некоторые свойства произведения матриц

1. Всегда существует такое произведение матриц $\mathbf{CC'}$ и $\mathbf{C'C}$

```
C %*% t(C)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  107  122  137  152
## [2,]  122  140  158  176
## [3,]  137  158  179  200
## [4,]  152  176  200  224
```

```
t(C) %*% C
```

```
##      [,1] [,2] [,3]
## [1,]   30   70  110
## [2,]   70  174  278
## [3,]  110  278  446
```

Некоторые свойства произведения матриц

1. Произведение матриц $\mathbf{BC}$ как правило не равно $\mathbf{CB}$

```
B <- matrix(1:9, ncol = 3)
C <- matrix(11:19, ncol = 3)

B %*%  C
```

```
##      [,1] [,2] [,3]
## [1,]  150  186  222
## [2,]  186  231  276
## [3,]  222  276  330
```

```
C %*% B
```

```
##      [,1] [,2] [,3]
## [1,]   90  216  342
## [2,]   96  231  366
## [3,]  102  246  390
```

Некоторые свойства произведения матриц

1. $[\mathbf{BC}]' = \mathbf{C}'\mathbf{B}'$

```
t(B %*% C)
```

```
##      [,1] [,2] [,3]
## [1,]  150  186  222
## [2,]  186  231  276
## [3,]  222  276  330
```

```
t(C) %*% t(B)
```

```
##      [,1] [,2] [,3]
## [1,]  150  186  222
## [2,]  186  231  276
## [3,]  222  276  330
```

Некоторые свойства произведения матриц

1. Произведение $\mathbf{BB'}$ и $\mathbf{B'B}$ всегда дает симметричную матрицу

```
B %*% t(B)
```

```
##      [,1] [,2] [,3]
## [1,]   66   78   90
## [2,]   78   93  108
## [3,]   90  108  126
```

```
t(B) %*% B
```

```
##      [,1] [,2] [,3]
## [1,]   14   32   50
## [2,]   32   77  122
## [3,]   50  122  194
```

Обращение (инверсия) матриц

В матричной алгебре нет процедуры деления. Вместо нее используют обращение матриц.

$$\mathbf{X}^{-1}\mathbf{X} = \mathbf{I}$$

Обратить можно только такую матрицу, у которой определитель не равен нулю

$$|\mathbf{X}| \neq 0$$

Матрицы, у которых определитель $|\mathbf{X}| = 0$ называются *сингулярными* матрицами они не могут быть инвертированы.

Важное свойство: Только квадратные матрицы имеют одну единственную обратную матрицу.

Поэтому для квадратных матриц справедливо $\mathbf{X}\mathbf{X}^{-1} = \mathbf{X}^{-1}\mathbf{X}$

Решение в среде R

Создадим матрицу

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8   10
```

Ее определитель

```
det(X)
```

```
## [1] -3
```

Решение в среде R

Обратная матрица

```
solve(X)
```

```
##        [,1]  [,2] [,3]
## [1,] -0.667 -1.33    1
## [2,] -0.667  3.67   -2
## [3,]  1.000 -2.00    1
```

По определению $\mathbf{X}^{-1}\mathbf{X} = \mathbf{I}$

```
round(solve(X) %*% X )
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

Примнение обратных матриц

Простейший случай использования обратных матриц - решение систем линейных уравнений

$$\begin{cases} 1x + 2y + 3z = 2 \\ 4x + 5y + 6z = 4 \\ 7x + 8y + 10z = 10 \end{cases}$$

Эту стстему можно представить в матричном виде

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 10 \end{pmatrix}$$

Тогда

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix}^{-1} \begin{pmatrix} 2 \\ 4 \\ 10 \end{pmatrix}$$

Задание

Решите приведенную систему уравнений с использованием матричной алгебры

Решение

```
Coef <- matrix(c(1 , 2 , 3 ,
         4 , 5 , 6 ,
         7 , 8 , 10), byrow = T, ncol = 3)
Val <- c(2,4,10)

solve(Coef) %*% Val
```

```
##       [,1]
## [1,]  3.33
## [2,] -6.67
## [3,]  4.00
```

Подбор параметров линейной регрессии

При подборе коэффициентов методом нименьших квадратов нам надо решить следующее матричное уравнение:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}$$

Здесь

$\mathbf{y}$ - вектор предсказанных значений

$\mathbf{X}$ - модельная матрица $\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$

$\boldsymbol{\beta}$ - вектор коэффициентов модели

Решение этого уравнения

Умножим обе части уравнения на транспонированную матрицу $\mathbf{X}'$

$$\mathbf{X}'\mathbf{y} = \mathbf{X}'\mathbf{X}\boldsymbol{\beta}$$

Матрица $\mathbf{X}'\mathbf{X}$ - это всегда квадратная матрица. Ее можно обратить.

Тогда

$$\boldsymbol{\beta} = [\mathbf{X}'\mathbf{X}]^{-1}[\mathbf{X}'\mathbf{y}]$$

Подбор коэффициентов линейной регрессии вручную

Подбираем коэффициенты с помощью фунции `lm()`

```
data(cars)
Mod <- lm(dist ~ speed, data = cars)
coefficients(Mod)
```

```
## (Intercept)       speed
##      -17.58        3.93
```

Подбираем коэффициенты вручную

```
X <- data.frame(Int = 1, x = cars$speed)
X <- as.matrix(X)

y <- cars$dist
solve(t(X) %*% X) %*% (t(X) %*%y)
```

```
##       [,1]
## Int -17.58
## x     3.93
```

```
d <- solve(t(X) %*% X)
```

Сингулярное разложение матриц (Singular value decomposition)

Теорема Экарта-Янга

Любую прямоугольную матрицу $\mathbf{Y}$ можно представить в виде произведения трех матриц:

$$\mathbf{Y}_{n\times p} = \mathbf{U}_{n\times p}\mathbf{D}_{p\times p}\mathbf{V}'_{p\times p}$$

То есть можно найти три "вспомогательных" матрицы, через котрые можно выразить любую другую матрицу.

Здесь

$\mathbf{Y}_{n\times p}$ - любая прямоугольная матрица $\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1c} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1} & a_{n2} & \cdots & a_{np} \end{pmatrix}$

$\mathbf{D}_{p\times p}$ - диагональная матрица $\begin{pmatrix} d_{11} & 0 & \cdots & 0 \\ 0 & d_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{pp} \end{pmatrix}$

По главной диагонали располагаются "особые" числа, называющиеся **сингулярными числами**. Сингулярные числа ранжируются от большего к меньшему.

$\mathbf{U}$ и $\mathbf{V}$ - левая и правая матрицы сингулярных векторов

Сингулярное разложение матрицы средствами R

```
set.seed(12345)
B <- matrix(round(runif(50, 1, 5))  , byrow = T, ncol=5) #Некоторая матрица
SVD <- svd(B) #Сингулярное Разложение матрицы B с помощью функции svd()
V <- SVD$v #"Вспомогательная" матрица - левые сингулярные вектора
D <- SVD$d #Вектор сингулярных чисел
U <- SVD$u #"Вспомогательная" матрица - правые сингулярные вектора
```

Вычислим $\mathbf{VDU}'$

```
U %*% diag(D) %*% t(V)
```

```
##       [,1] [,2] [,3] [,4] [,5]
##  [1,]    4    5    4    5    3
##  [2,]    2    2    3    4    5
##  [3,]    1    2    4    1    3
##  [4,]    3    3    3    2    5
##  [5,]    3    2    5    4    4
##  [6,]    3    4    3    2    3
##  [7,]    4    1    2    4    2
##  [8,]    2    4    5    3    2
##  [9,]    4    3    5    4    2
## [10,]    2    1    1    1    4
```

Задание

Вычислите матрицу, которая получится при использовании только 1 и 2 сингулярного числа для матрицы $\mathbf{B}$, использованной на предыдущем слайде.

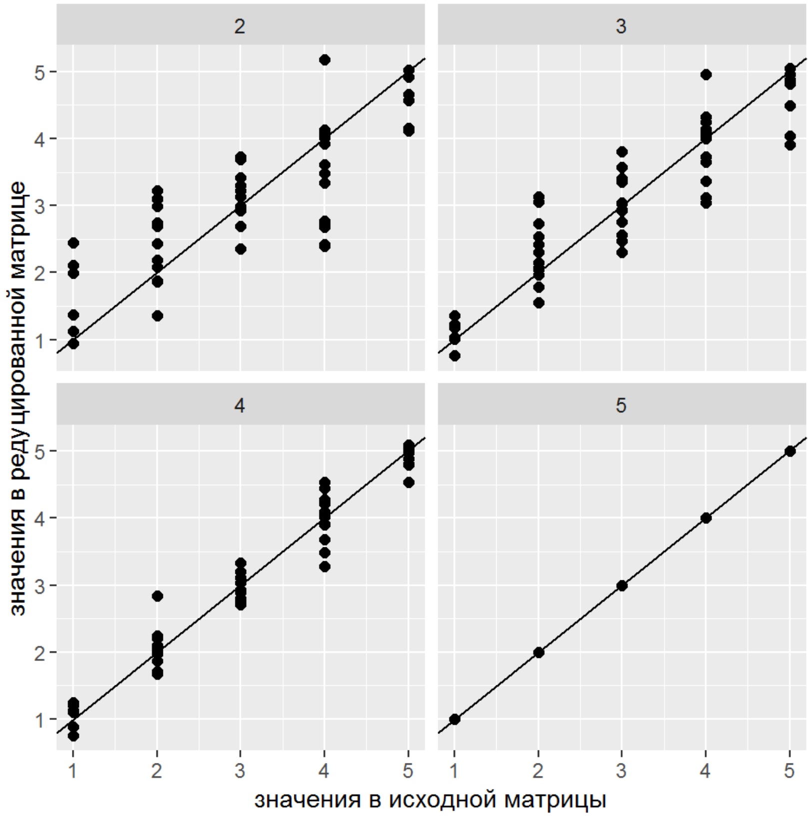
Решение

```
U[,1:2] %*% diag(D[1:2]) %*% t(V[,1:2])
```

```
##       [,1]  [,2] [,3] [,4] [,5]
##  [1,] 3.92 4.116 5.17 4.56 2.96
##  [2,] 2.75 2.430 3.22 2.77 4.92
##  [3,] 1.99 1.864 2.43 2.10 2.92
##  [4,] 2.69 2.351 3.13 2.69 5.02
##  [5,] 3.30 3.220 4.14 3.61 4.01
##  [6,] 2.70 2.674 3.42 2.99 3.00
##  [7,] 2.39 2.444 3.09 2.72 2.19
##  [8,] 3.11 3.333 4.16 3.68 1.88
##  [9,] 3.48 3.732 4.66 4.12 2.09
## [10,] 1.36 0.948 1.37 1.13 4.06
```

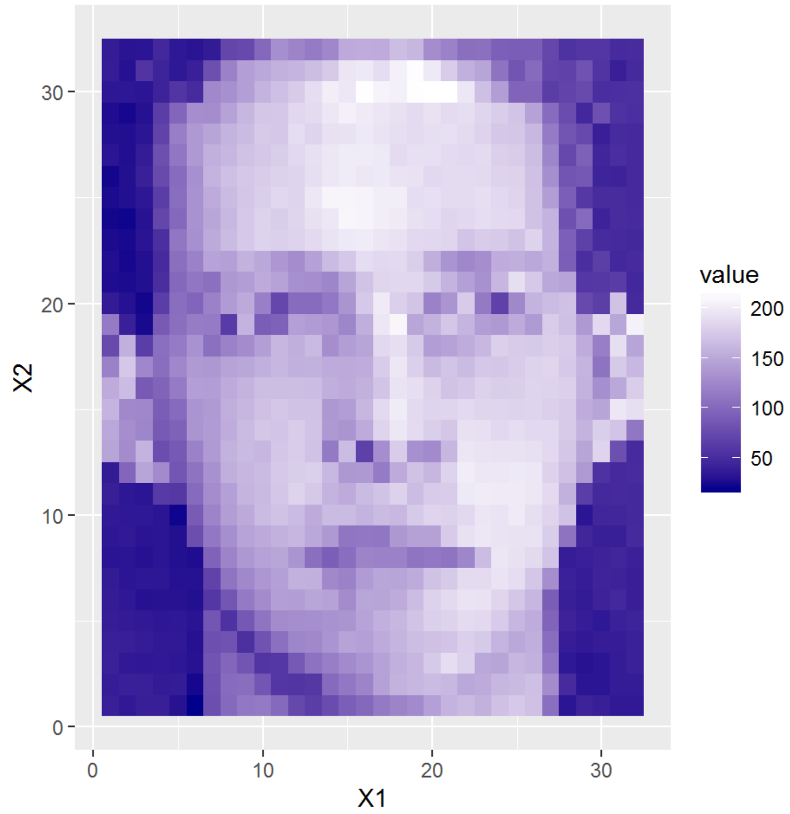
Важное свойство сингулярных чисел

Если вычислить матрицу на основе не всех, а части сингулярных чисел, то новая матрица будет подобна исходной матрице.



Применение свойства сингулярных чисел в сжатии изображений

Пример взят из курса лекций "Data Analysis" by Jeffrey Leek (https://github.com/jtleek/dataanalysis/tree/master/week3)



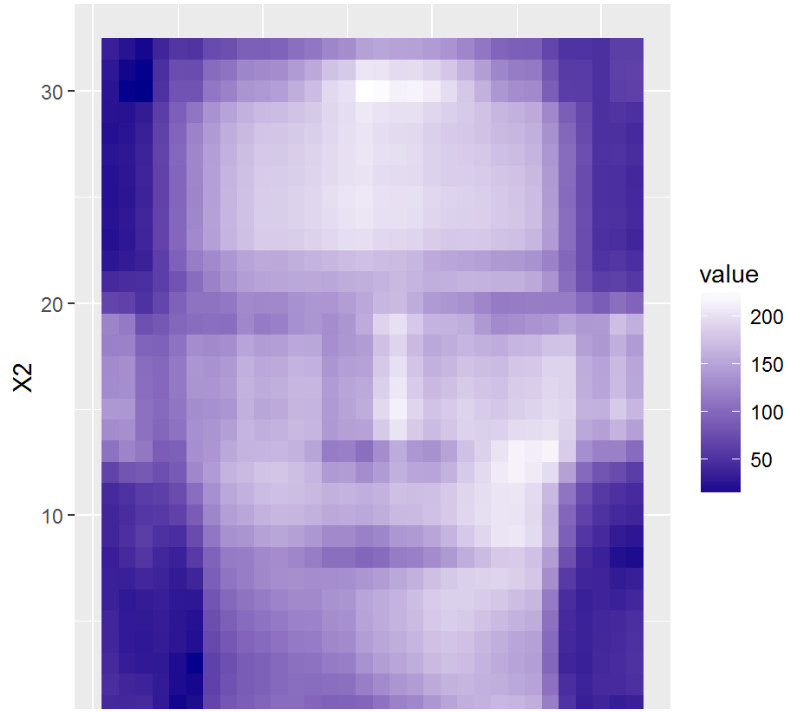
Произведем сингулярное разложение матрицы `faceData`

```
SVD_face <- svd(faceData)

U <- SVD_face$u
D <- SVD_face$d
V <- SVD_face$v

reduction <- function(x) U[,1:x] %*% diag(D[1:x]) %*% t(V[, 1:x])

gg_face(reduction(4))
```



Применение SVD в биологических исследованиях

SVD - это метод, на котором основаны разные типы анализа, связанные со снижением размерности: PCA, CCA.

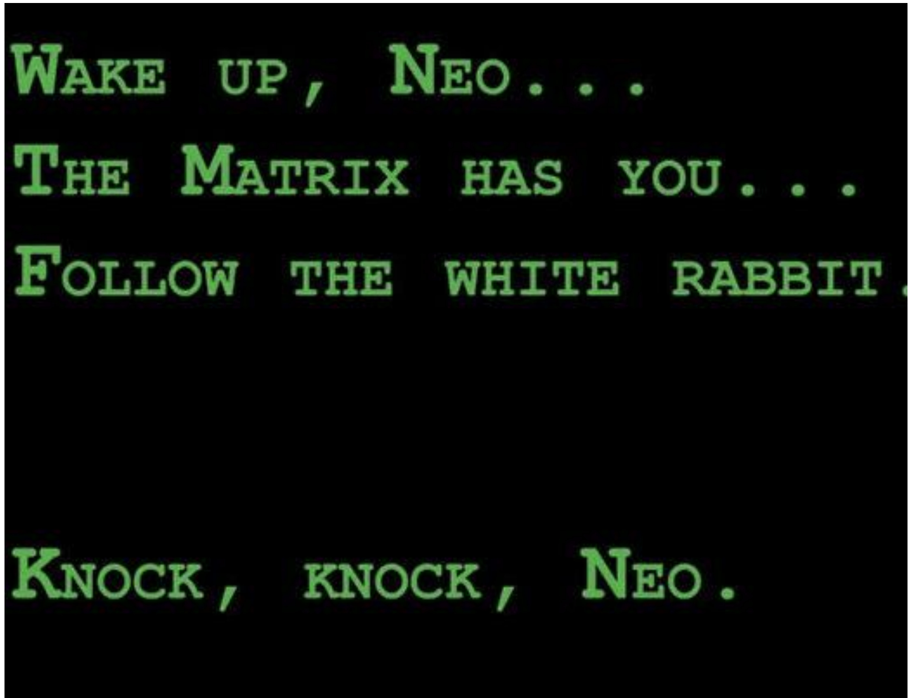
О них в следующей лекции

Summary

- Матричная алгебра позволяет решать самые разные типы задач.
- Матричные методы лежат в основе очень многих типов анализа.
- В основе многих методов снижения размерности лежит SVD.

Что почитать

- Legendre P., Legendre L. (2012) Numerical ecology. Second english edition. Elsevier, Amsterdam. Глава 2. Matrix algebra: a summary.


Wake up, Neo...
The Matrix has you...
Follow the white rabbit.
Knock, knock, Neo.