



北京邮电大学

BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS



Data Analytics with R
语言数据分析



数据对象

艾新波 / 2018 • 北京



课程体系



R语言数据分析



上部：论道



- 第1章 气象万千、数以等观
- 第2章 所谓学习、归类而已
- 第3章 格言联璧话学习
- 第4章 源于数学、归于工程



中部：执具



- 第5章 工欲善其事必先利其器
- 第6章 基础编程
- 第7章 数据对象



- 第8章 人人都爱tidyverse
- 第9章 最美不过数据框



下部：博术



- 第10章 观数以形
- 第11章 相随相伴、谓之关联
- 第12章 既是世间法、自当有分别
- 第13章 方以类聚、物以群分
- 第14章 庐山烟雨浙江潮

矩阵运算

求解以下线性方程组：

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 1 \\ 2x_1 + 2x_2 + 5x_3 = 2 \\ 3x_1 + 5x_2 + x_3 = 3 \end{cases}$$

函数 **solve (A, b)** 用于求解线性方程组 $AX = b$

若 **b** 缺省，则默认为单位矩阵

矩阵运算

#定义系数矩阵

```
A <- matrix(  
  c(1, 2, 3,  
    2, 2, 5,  
    3, 5, 1),  
  ncol = 3,  
  byrow = TRUE)  
b <- 1:3
```

```
solve(A,b)
```

```
#> [1] 1 0 0
```

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 1 \\ 2x_1 + 2x_2 + 5x_3 = 2 \\ 3x_1 + 5x_2 + x_3 = 3 \end{cases}$$

矩阵运算

#可以利用solve函数求逆矩阵

diag(3) #生成单位矩阵

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

```
#> [1,]      1     0     0
```

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

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

solve(A, **diag**(3))

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

```
#> [1,] -1.53  0.87  0.27
```

```
#> [2,]  0.87 -0.53  0.07
```

```
#> [3,]  0.27  0.07 -0.13
```

$$\text{solve}(A, b)$$

$$AX = b$$

$$\Downarrow$$

$$AX = E$$

$$X = X'$$

矩阵运算

```
solve(A) %*% A
```

```
#>                [,1]                [,2]                [,3]  
#> [1,]  1.00000000000000044408921  0.0000000000000008881784  0.000000000000010547119  
#> [2,] -0.00000000000000044408921  0.9999999999999993338662 -0.000000000000007077672  
#> [3,] -0.0000000000000005551115 -0.0000000000000001110223  1.000000000000000000000
```

```
sqrt(2) ^ 2 == 2
```

```
#> [1] FALSE
```

```
dplyr::near(sqrt(2) ^ 2, 2)
```

```
#> [1] TRUE
```

```
all(near(solve(A) %*% A, diag(3)))
```

```
#> [1] TRUE
```

R与Matlab的比较

功能	Matlab	R
矩阵相乘	<code>A*B</code>	<code>A%*%B</code>
矩阵元素相乘	<code>A.*B</code>	<code>A*B</code>
矩阵转置	<code>A'</code>	<code>t(A)</code>
矩阵行列式	<code>det(A)</code>	<code>det(A)</code>
矩阵求逆	<code>inv(A)</code>	<code>solve(A)</code>
矩阵的迹	<code>trace(A)</code>	<code>sum(diag(A))</code>
矩阵的秩	<code>rank(A)</code>	<code>qr(A)\$rank</code>

更多R与Matlab的对照：

<http://cran.r-project.org/doc/contrib/Hiebeler-matlabR.pdf>

数据对象



当前位置

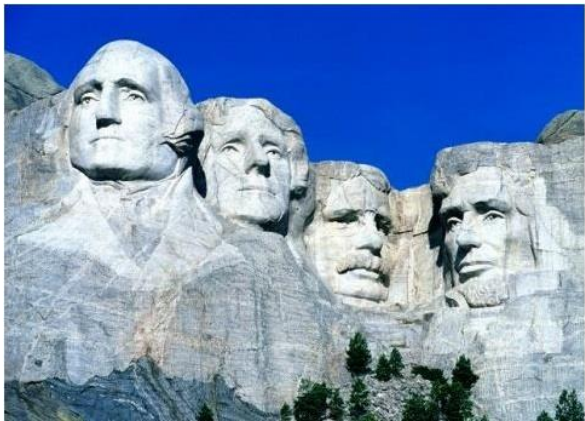
设备或人工采集到的数据
(对现实系统进行观测、记录)



数组

数组是矩阵的扩展，矩阵是二维数组
下面以图像处理为例，简述三维数组的操作

345



482



数组

#读入一个彩色JPEG文件，在R里边就是一个数组

```
jpg_url <-
```

```
"https://raw.githubusercontent.com/byaxb/RDataAnalytics/master  
/data/presidents.jpg"
```

```
download.file(jpg_url,  
              "presidents.jpg", mode="wb")
```

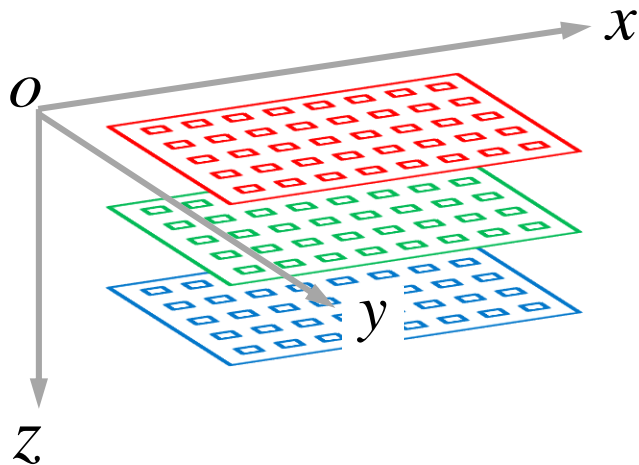
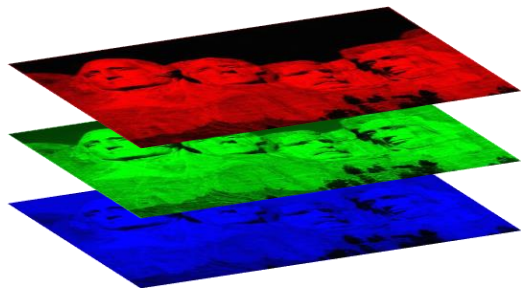
```
library(imager)
```

```
presidents <- load.image("presidents.jpg")
```

```
str(presidents)
```

```
#> 'cimg' num [1:482, 1:345, 1, 1:3] 1 0.976 0.929 0.914 ...
```

图像与数组



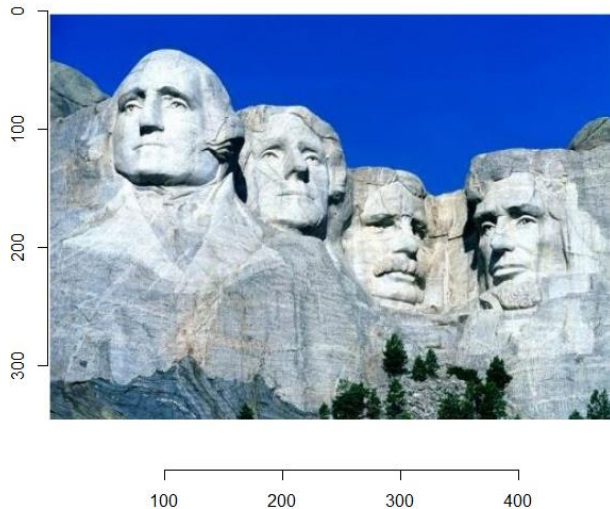
图像



数组

图像与数组

```
presidents <- load.image("presidents.jpg")  
plot(presidents)
```



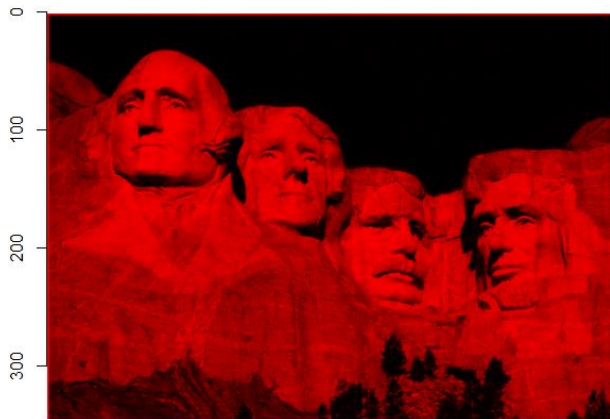
图像与数组

```
presidents <- load.image("presidents.jpg")
```

```
presidents[, , 2] <- 0
```

```
presidents[, , 3] <- 0
```

```
plot(presidents)
```



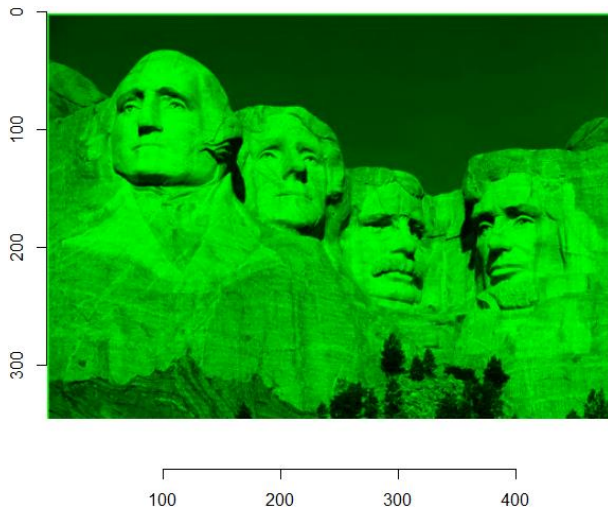
图像与数组

```
presidents <- load.image("presidents.jpg")
```

```
presidents[, , 1] <- 0
```

```
presidents[, , 3] <- 0
```

```
plot(presidents)
```



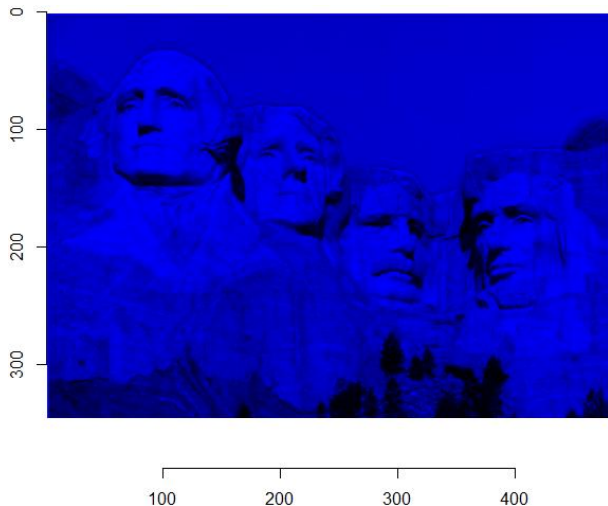
图像与数组

```
presidents <- load.image("presidents.jpg")
```

```
presidents[, , 1] <- 0
```

```
presidents[, , 2] <- 0
```

```
plot(presidents)
```



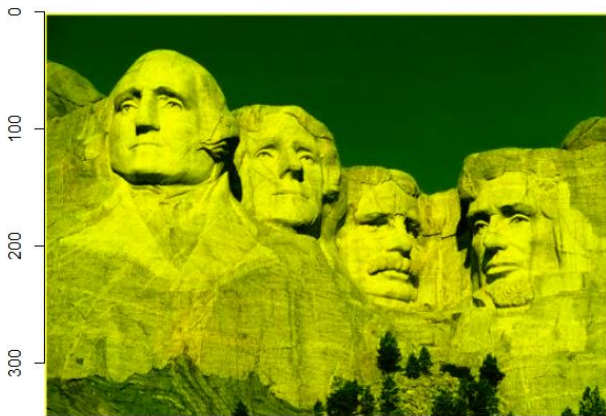
图像与数组

```
presidents <- load.image("presidents.jpg")
```

```
#黄色<==#FFFF00
```

```
presidents[, , 3] <- 0
```

```
plot(presidents)
```



100 200 300 400

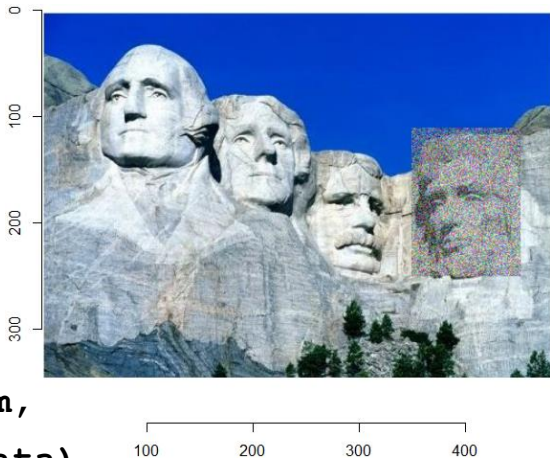
图像与数组

```
area_coor_x <- 350:449 #100
area_coor_y <- 110:259 #150
array_dim <- c(length(area_coor_x),
               length(area_coor_y),
               3)

array_data <- runif(prod(array_dim))
random_noise <- array(dim = array_dim,
                     data = array_data)

presidents[area_coor_x, area_coor_y,] <-
  (1 - 0.6) * presidents[area_coor_x, area_coor_y,] +
  0.6 * random_noise

plot(presidents)
```



A decorative blue border with rounded corners frames the entire slide. Two thin blue crosshair-like lines are positioned diagonally, one in the upper right and one in the lower left, intersecting at the center of the text.

谢谢聆听
Thank you

教师个人联系方式

艾新波

手机: 13641159546

QQ: 23127789

微信: 13641159546

E-mail: 13641159546@126.com

axb@bupt.edu.cn

地址: 北京邮电大学科研楼917室

课程网址: <https://github.com/byaxb/RDataAnalytics>

