

Chap1      introduction 1

x <- rnorm(10)	Tạo 10 số ngẫu nhiên từ phân phối chuẩn và lưu trữ chúng trong biến x	
x		
mean(x)		
sd(x)		
plot(x)		
install.packages("ggplot2") library(ggplot2) x <- c(1.1,2,3.5,3.9,4.2) y <- c(2,2.2,-1.3,0,0.2) qplot(x,y) update.packages()		
2. vector		
> 2+3 [1] 5 > 14/6 [1] 2.333333 > 14/6+5 [1] 7.333333 > 14/(6+5) [1] 1.272727 > 3^2 [1] 9	> sqrt(9) [1] 3 > sqrt(x=5.3) [1] 2.302173 > log(x=8, base=2) [1] 3 > exp(x=3) # e3 [1] 20.08554	
x <- -3 x <- x + 5 z <- x * 3.5		
ls()	Lệnh hiển thị tất cả các biến hiện được khai báo.	
A <- c(1,3,5)		Gán vector vào biến A
B <- matrix(c(1,2,3,4),2,2)		Gán matrix vào biến B Tự động điền dọc rồi ngang

Vector Lưu trữ dữ liệu có cùng định dạng (số nguyên, số thực, chuỗi, giá trị logic, v.v.) theo thứ tự.	
<pre>vec &lt;- c(1,4,-2,3) vec2 &lt;- c(0,2) vec3 &lt;- c(vec, vec2)    #cùng format</pre>	
<p>Dãy số</p> <pre>3:8 x &lt;- 2.5:7 3.5:-3.5</pre>	
seq( ) và rep( )	
<pre>seq(from=3,to=21,by=3) seq(from=3,to=21,length.out=5) #5 số cách đều nhau</pre>	<pre>[1] 3 6 9 12 15 18 21 3.0 7.5 12.0 16.5 21.0</pre>
<pre>rep(x=1,times=3) vec &lt;- c(1,0,2,4) rep(x=vec,times=2) rep(x=vec,each=2) rep(x=vec,each=2, time=2)</pre>	<pre>[1] 1 0 2 4 1 0 2 4 [1] 1 1 0 0 2 2 4 4</pre>
Sắp xếp, độ dài và lấy giá trị phần tử của vector	
<pre>vec &lt;- c(-1, 3, 2, -5, 6) sort(x=vec) sort(x=vec,decreasing=TRUE) length(x=vec) vec[1] vec[4] vec[length(vec)]    # end ele</pre>	#tăng dần
<pre>vec[-1]    #in ra 1 vector xóa số đầu, vector gốc giữ nguyên vec[-(length(vec))]</pre> <pre>vec[1:3]</pre>	

<pre>vec[c(1,3,5)] vec[2] &lt;- 0    #thay đổi ele trong vector vec[1:3] &lt;- 1</pre>		
Phép toán với vector		
<pre>vec= c(-1, 3, 2, -5) vec2 = rep(3,4) vec + vec2 vec3+3 vec * c(1,-1) # (-1,3,2,-5) * (1,-1,1,-1) sum(vec) prod(vec)</pre>		
<b>Matrix</b>	Ma trận là một cấu trúc dữ liệu kết hợp nhiều vector.	
<pre>A &lt;- matrix(data=c(1,2,3,4,5,6),nrow=2,ncol=3) A &lt;- matrix(data=c(1,2,3,4,5,6),nrow=2,ncol=3, byrow=TRUE)</pre>	<pre>dim(A) nrow(A), nrow(B)</pre>	
<pre>A &lt;- rbind(1:4,c(0,2,-1,1)) B &lt;- cbind(1:4,c(0,2,-1,1))</pre>	<pre>A[2,3]</pre>	
<pre>A[,2]    #2nd col A[1,]    #1st row A[,2:3]  # Trích xuất ma trận con từ cột thứ 2 và thứ 3 của A A[,c(1,3)] # Trích xuất ma trận con từ cột thứ 1 và thứ 3 của A diag(A) # trích xuất các phần tử đường chéo của A</pre>	<pre>&gt; A       [,1] [,2] [,3] [,4] [1,]    1    2    3    4 [2,]    0    2   -1    1</pre>	
<pre>A[, -2] A[-1, -c(1,3)] A[,3] &lt;- 1:2 A[1,4] &lt;- 0 diag(A) &lt;- 3</pre>	<pre>Del 2 nd col Del 1st row, 1,3 col 3rd col val</pre>	
<pre>A &lt;- rbind(c(1,2), c(0,-1)) t(A) B &lt;- rbind(c(1,-1), c(3,0))</pre>		

A+B   2*A  A%*%B   solve(A)   det(A)	Định thức	
Non numeric value		
<pre>&gt; sort(x=vec,decreasing=FALSE) &gt; A &lt;- matrix(data=c(1,2,3,4,5,6),nrow=2,ncol=3, byrow=TRUE)</pre>		
<pre>y &lt;- T x &lt;- c(T,F,T,T,F,F) matrix(data=x,nrow=2)</pre>		
<pre>&gt; 1 == 2 [1] FALSE &gt; 1 &gt; 2 [1] FALSE &gt; (2-1) &lt;= 2 [1] TRUE &gt; 1 != (2+3) [1] TRUE</pre>		
<pre>&gt; x &lt;- c(1,0,2,3) &gt; y &lt;- c(0,2,-1,3) &gt; x == y A &lt;- cbind(1:2, 3:4) B &lt;- cbind(c(1,1), c(2,4)) A == B</pre>		
<pre>&gt; (3&gt;1) &amp;&amp; (2!=1) [1] TRUE &gt; T &amp;&amp; (T    F) [1] TRUE  &gt; x &lt;- c(T,F,T,F) &gt; y &lt;- c(F,F,T,T) &gt; x&amp;y      # 4의 원소 각각에 AND 연산자 적용 [1] FALSE FALSE TRUE FALSE &gt; x y      # 4의 원소 각각에 OR 연산자 적용 [1] TRUE FALSE TRUE TRUE</pre>		
Factor		
<pre>name &lt;- c("Kim", "Lee", "Park", "Han", "Choi", "Kang", "Song", "Yoon") &gt; btype &lt;- c("A", "B", "A", "AB", "O", "A", "O", "O") &gt; grade &lt;- c("A", "C", "B", "A", "A", "B", "B", "F")</pre>		

btype.fac <- factor(btype) btype.fac grade.fac <- factor(grade) grade.fac levels(grade.fac)	btype vector → factor
grade.fac <- factor(grade, order=TRUE, levels=c("A","B","C","D","F"))	
grade.fac str(grade.fac) name[grade.fac == "A"] as.character(grade.fac) as.integer(grade.fac)	