SCS2211

Laboratory II

21000298

1)

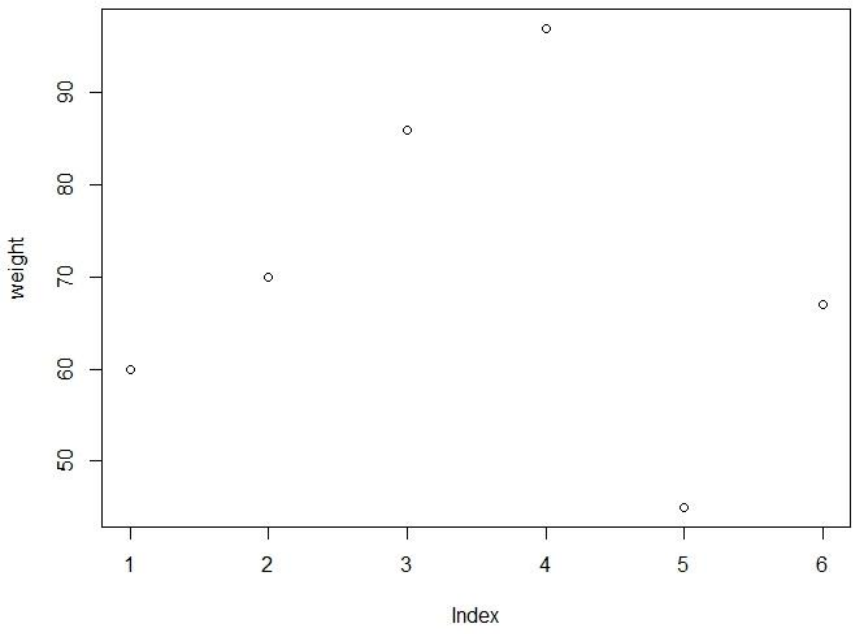
```
> 2+2
[1] 4
> exp(-2)
[1] 0.1353353
> log(100, base=2)
[1] 6.643856
> log(100, base=10)
[1] 2
> runif(10) #generate 10 random numbers between 1 and 10
 [1] 0.9564345 0.2640397 0.6238564 0.8527002 0.5277848 0.9315262 0.4657243 0.9264955 0.8130663
[10] 0.6688136
> runif(10) #2nd time
 [1] 0.4054859 0.4276979 0.1947999 0.3908058 0.1686931 0.9981894 0.1255823 0.2983316 0.2070237
[10] 0.7615949
>
```

2)

```
> x = 2
> x + x
[1] 4
> y = x + 3
> y
[1] 5
> s = "this is a char str"
> s
[1] "this is a char str"
```

3)

```
> weight = c(60, 70, 86, 97, 45, 67)
> weight
[1] 60 70 86 97 45 67
> plot(weight)
> seq(0,1, length = 11)
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> seq(4, 10, 0.5)
 [1]  4.0  4.5  5.0  5.5  6.0  6.5  7.0  7.5  8.0
[10]  8.5  9.0  9.5 10.0
> seq(length=10)
 [1]  1  2  3  4  5  6  7  8  9 10
> seq(length=10)
 [1]  1  2  3  4  5  6  7  8  9 10
> help(seq)
> seq(4,10)
[1]  4  5  6  7  8  9 10
> seq(4,10, length=20)
 [1]  4.000000  4.315789  4.631579  4.947368
 [5]  5.263158  5.578947  5.894737  6.210526
 [9]  6.526316  6.842105  7.157895  7.473684
[13]  7.789474  8.105263  8.421053  8.736842
[17]  9.052632  9.368421  9.684211 10.000000
> x = seq(10)
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> #c() function is used to combine vectors as well
as scalars
> c(x, 1:10, 100)
 [1]   1   2   3   4   5   6   7   8   9  10   1
[12]   2   3   4   5   6   7   8   9  10 100
>
> a = c(1,2,3,4,5,6,7)
> a^2
[1]  1  4  9 16 25 36 49
>
```



seq {base} R Documentation

Sequence Generation

Description

Generate regular sequences. `seq` is a standard generic with a default method. `seq.int` is a primitive which can be much faster but has a few restrictions. `seq_along` and `seq_len` are very fast primitives for two common cases.

Usage

```
seq(...)

## Default S3 method:
seq(from = 1, to = 1, by = ((to - from)/(length.out - 1)),
    length.out = NULL, along.with = NULL, ...)

seq.int(from, to, by, length.out, along.with, ...)

seq_along(along.with)
seq_len(length.out)
```

Arguments

`...`	arguments passed to or from methods.
`from, to`	the starting and (maximal) end values of the sequence. Of length `1` unless just `from` is supplied as an unnamed argument.
`by`	number: increment of the sequence.
`length.out`	desired length of the sequence. A non-negative number, which for `seq` and `seq.int` will be rounded up if fractional.
`along.with`	take the length from the length of this argument.

4)

```
> x = rnorm(100)
> x
  [1]  0.49477148  0.29453923  0.08646578
  [4] -0.25867894 -0.62044815 -0.28849412
  [7]  0.59843421 -2.23841649 -0.39161191
 [10] -2.06303404  1.73308339 -0.19376409
 [13]  0.24590574  0.76148706  0.55696117
 [16]  1.23838018 -0.01870060 -1.84095392
 [19] -0.71823224 -1.23378298 -0.91987014
 [22] -0.45052762 -1.60948604 -0.17766269
 [25] -2.50074673  0.07698542 -1.42468435
 [28]  0.23852217  1.69987024 -1.12218756
 [31] -0.93113166 -0.19399932 -1.22756853
 [34]  1.75073437 -0.47248586  1.73147013
 [37] -0.16374439  0.25103133  0.95333138
 [40]  0.31649789  0.13350309  0.27438596
 [43]  1.03709307  0.46154285 -0.68216155
 [46]  0.96923308  1.52893850  0.70773462
 [49] -0.11437242  0.37154121 -1.02872826
 [52]  0.10842713 -0.22327698 -1.05717215
 [55] -2.72472887  0.86180224  0.50213631
 [58] -0.79884958 -1.04657937  0.55960644
 [61]  0.09343408  0.20187268  0.70382207
 [64] -2.04733028  0.03442386 -0.30932682
 [67] -0.44973241 -0.43674171  0.63113200
 [70]  0.13431748 -0.52852759  0.26962133
 [73] -0.46033398 -0.41707346  0.13359825
 [76]  0.33019866  1.97601021  0.82587435
 [79]  1.26766475 -0.21451741 -1.91902483
 [82]  0.84757398 -2.29082715  0.39533046
 [85]  0.74151676  0.31397169  0.89089199
 [88] -0.79941923 -0.34423529  1.28475826
 [91]  0.47275665 -0.74812339 -1.25122805
 [94] -1.46636744 -1.64214177 -0.20598955
 [97] -0.18564341 -0.87794823 -0.16472474
[100] -1.79703105
```

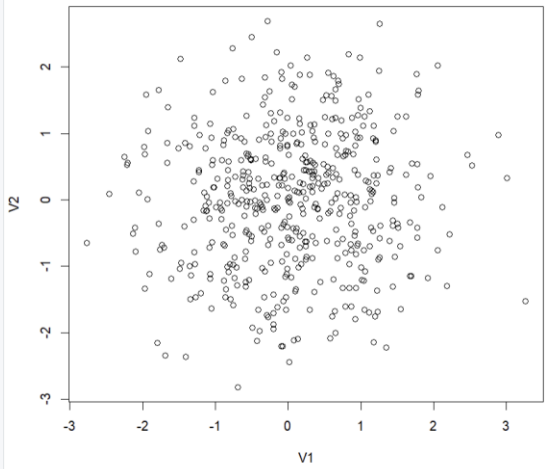
```
> mean(x)
[1] -0.1519918
> sd(x)
[1] 1.012246
> var(x)
[1] 1.024641
> median(x)
[1] -0.1390584
> quantile(x)
        0%        25%        50%        75%
-2.7247289 -0.7608049 -0.1390584  0.4966127
      100%
 1.9760102
> IQR(x)
[1] 1.257418
> fivenum(x)
[1] -2.7247289 -0.7734865 -0.1390584  0.4984539
[5]  1.9760102
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.7247 -0.7608 -0.1391 -0.1520  0.4966  1.9760
```

5)

```
> getwd()
[1] "E:/22Second Sem/Laboratory 2/R/Practical - 01"
> setwd("E:/22Second Sem/Laboratory 2/R/Practical - 01")
>
> getwd()
[1] "E:/22Second Sem/Laboratory 2/R/Practical - 01"
> setwd("E:/22Second Sem/Laboratory 2/R/Practical - 01/check_setwd")
> getwd()
[1] "E:/22Second Sem/Laboratory 2/R/Practical - 01/check_setwd"
> setwd("E:/22Second Sem/Laboratory 2/R/Practical - 01")
> getwd()
[1] "E:/22Second Sem/Laboratory 2/R/Practical - 01"
> |
```

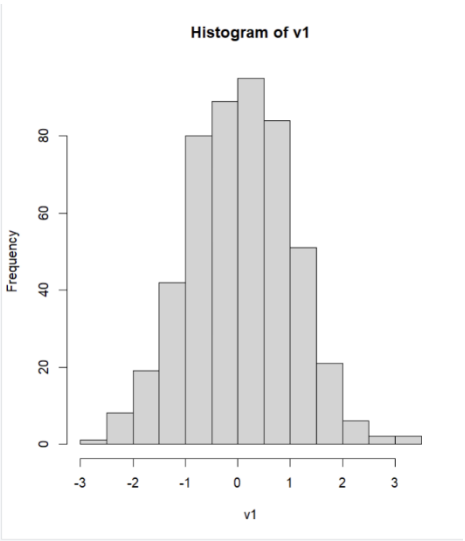
```
> d1 = read.table("d1.txt")
> d1
         V1      V2
1    0.8706 -0.7560
2    0.3308  0.1212
3   -1.3479 -1.4958
4    1.5479 -1.6462
5   -0.6166  0.1929
6   -0.6986 -1.2898
7   -1.4236 -0.3998
8   -0.1096  0.7754
9   -0.1027  0.9221
10  -0.3227  1.5445
11  -0.8396 -1.0096
12  -1.6606  0.8567
13   3.0125  0.3253
14   1.5008  1.2524
15  -0.3590  0.2202
16  -0.1149  0.8152
17   0.3589  1.1394
18  -0.0121 -0.4117
19  -0.5704 -0.4611
20   0.9340 -1.3290
21  -1.1383 -0.0875
22   1.9572  0.3554
23  -1.7632 -0.7431
24  -0.3233  0.0720
25   0.8726 -0.3564
26   0.1179 -0.4317
27  -1.5031  0.7793
28   0.9218  0.8163
29   0.1632  1.1483
30  -0.8147 -1.0850
31   0.4037 -0.1361
32   0.7562  0.6551
33   2.1844 -1.2935
34  -0.0938  1.9267
35  -0.2942  0.2333
36  -0.5048 -0.5450
37  -1.9270  1.0328
```

```
> summary(d1)
       V1                  V2
 Min.   :-2.77120   Min.   :-2.819800
 1st Qu.:-0.58403   1st Qu.:-0.688750
 Median : 0.03610   Median : 0.038650
 Mean   : 0.05106   Mean   : 0.003601
 3rd Qu.: 0.73395   3rd Qu.: 0.685400
 Max.   : 3.26620   Max.   : 2.689000
> plot(d1)
> col1 = d1[1]
> col1Vector = c(col1)
> col1Vector
$V1
  [1]  0.8706  0.3308 -1.3479  1.5479 -0.6166 -0.6986 -1.4236 -0.1096 -0.1027 -0.3227 -0.8396
 [12] -1.6606  3.0125  1.5008 -0.3590 -0.1149  0.3589 -0.0121 -0.5704  0.9340 -1.1383  1.9572
 [23] -1.7632 -0.3233  0.8726  0.1179 -1.5031  0.9218  0.1632 -0.8147  0.4037  0.7562  2.1844
 [34] -0.0938 -0.2942 -0.5048 -1.9270  0.5214 -2.0544 -0.2698 -0.2750  0.8451 -2.7712  0.9181
 [45] -0.5193 -1.2930 -0.0087  0.2334 -0.4242 -1.4694  1.2064  0.1973 -0.7069  1.2164  0.3672
 [56] -1.2675  0.6211 -1.7955 -1.0578  0.1515  0.4113  0.0705 -1.9330  0.8187  1.2618  1.1673
 [67] -0.5703 -0.3939 -1.0174  0.2502  0.2521 -0.7715  0.9195  0.8522  0.6571 -0.7535 -1.3385
 [78]  0.6522  1.4472 -1.2901 -2.2082  1.4361 -0.0617  1.1778  0.9855 -1.2186 -0.4319 -0.8349
 [89]  0.1797  1.1663  0.0560 -2.1000  1.2353  0.0027 -0.4595 -2.1032  0.3739  0.2452  0.3386
[100] -1.0781 -0.7302 -0.9163  1.7876 -0.8204 -0.1967 -0.8901  0.9107 -0.0123  0.0728  0.9394
[111]  0.6752  0.7860 -2.1327  0.2199  0.9663  0.3500  0.9018  2.1212  1.2486 -1.1232 -0.8301
[122] -0.1218 -0.6426 -0.0789  1.2281 -0.5314 -0.2861 -0.2276  0.6745  1.0368 -0.1495 -0.3171
[133]  0.9336  0.7233  0.4882  3.2662  0.0317  0.3413 -0.0835  0.6164 -0.5250  1.0077  1.8291
[144]  0.0853 -0.0683 -0.5635 -0.4736 -1.7035 -0.0537 -0.8813  1.2559  0.1558  0.0549  1.3986
[155] -1.7756  0.3297 -1.0579 -0.6434  0.2584  0.8917 -0.8366  0.5531  1.4584 -0.8551 -0.9921
[166] -0.0117  0.6269  0.0015 -0.8163  0.1151  0.2071 -0.4446 -1.1205  0.4354  0.0171 -0.3630
[177] -0.6312 -0.5003 -0.8672 -1.0401  1.2654 -0.2415 -1.7290 -0.4882  1.0604 -0.5382  1.7773
[188] -0.7795 -0.7530 -1.0331  1.1638 -0.5801  0.4173 -1.6481 -0.5727  0.6592 -1.2804  0.0501
[199]  0.5484  1.7784  0.6411  0.9467 -0.3543 -0.5788  0.0565 -0.8933  1.1221  0.7758  0.9625
[210]  1.2608 -0.1079 -0.5772  0.5256  1.1520  0.8584  0.9456  1.5061  0.2953  1.2041 -0.0118
[221]  0.7895 -1.4036  1.5421 -1.7749  0.2057 -0.3462 -1.1590  0.3358  0.3322  1.4112  0.1369
[232]  0.5776  0.2431  1.4677  1.1306  0.0107  0.1243  1.7794 -0.2395  0.8580 -0.8784 -0.7619
[243]  0.8628  0.6483  1.0581 -0.6332  1.0180  0.1719 -0.0475  1.6891  1.4370 -2.2511  0.3565
[254] -0.8502 -0.2996 -0.6342  1.6245  1.2411  0.5553  0.7034  0.4582  0.6840  0.2513 -0.1785
[265]  0.5077 -0.3099 -0.3944 -0.2697 -0.0881  0.0080  2.5318 -1.2232 -0.0718  0.2461 -0.0506
[276] -0.7302  0.3270  0.7530 -1.1537 -0.4079 -1.2879  0.0836  0.1638  0.6826 -1.0864  0.2975
[287] -0.1433  1.3920  0.3067 -0.5372 -0.2289 -0.5362  1.4390 -0.5111 -1.6068 -0.2012  1.1435
```

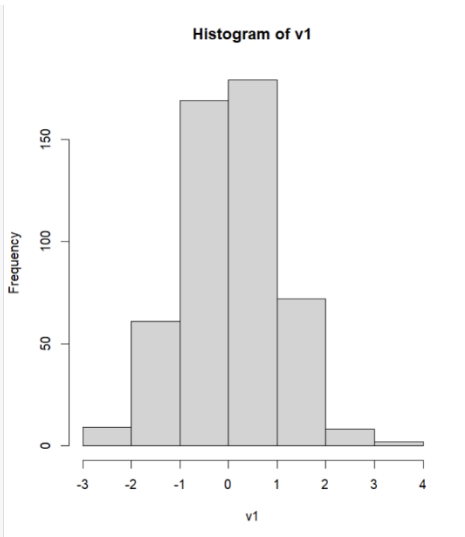



```
> v1 = as.numeric(unlist(col1))
> v1
  [1]  0.8706  0.3308 -1.3479  1.5479 -0.6166 -0.6986 -1.4236 -0.1096 -0.1027 -0.3227 -0.8396
 [12] -1.6606  3.0125  1.5008 -0.3590 -0.1149  0.3589 -0.0121 -0.5704  0.9340 -1.1383  1.9572
 [23] -1.7632 -0.3233  0.8726  0.1179 -1.5031  0.9218  0.1632 -0.8147  0.4037  0.7562  2.1844
 [34] -0.0938 -0.2942 -0.5048 -1.9270  0.5214 -2.0544 -0.2698 -0.2750  0.8451 -2.7712  0.9181
 [45] -0.5193 -1.2930 -0.0087  0.2334 -0.4242 -1.4694  1.2064  0.1973 -0.7069  1.2164  0.3672
 [56] -1.2675  0.6211 -1.7955 -1.0578  0.1515  0.4113  0.0705 -1.9330  0.8187  1.2618  1.1673
 [67] -0.5703 -0.3939 -1.0174  0.2502  0.2521 -0.7715  0.9195  0.8522  0.6571 -0.7535 -1.3385
 [78]  0.6522  1.4472 -1.2901 -2.2082  1.4361 -0.0617  1.1778  0.9855 -1.2186 -0.4319 -0.8349
 [89]  0.1797  1.1663  0.0560 -2.1000  1.2353  0.0027 -0.4595 -2.1032  0.3739  0.2452  0.3386
[100] -1.0781 -0.7302 -0.9163  1.7876 -0.8204 -0.1967 -0.8901  0.9107 -0.0123  0.0728  0.9394
[111]  0.6752  0.7860 -2.1327  0.2199  0.9663  0.3500  0.9018  2.1212  1.2486 -1.1232 -0.8301
[122] -0.1218 -0.6426 -0.0789  1.2281 -0.5314 -0.2861 -0.2276  0.6745  1.0368 -0.1495 -0.3171
[133]  0.9336  0.7233  0.4882  3.2662  0.0317  0.3413 -0.0835  0.6164 -0.5250  1.0077  1.8291
[144]  0.0853 -0.0683 -0.5635 -0.4736 -1.7035 -0.0537 -0.8813  1.2559  0.1558  0.0549  1.3986
[155] -1.7756  0.3297 -1.0579 -0.6434  0.2584  0.8917 -0.8366  0.5531  1.4584 -0.8551 -0.9921
[166] -0.0117  0.6269  0.0015 -0.8163  0.1151  0.2071 -0.4446 -1.1205  0.4354  0.0171 -0.3630
[177] -0.6312 -0.5003 -0.8672 -1.0401  1.2654 -0.2415 -1.7290 -0.4882  1.0604 -0.5382  1.7773
[188] -0.7795 -0.7530 -1.0331  1.1638 -0.5801  0.4173 -1.6481 -0.5727  0.6592 -1.2804  0.0501
[199]  0.5484  1.7784  0.6411  0.9467 -0.3543 -0.5788  0.0565 -0.8933  1.1221  0.7758  0.9625
[210]  1.2608 -0.1079 -0.5772  0.5256  1.1520  0.8584  0.9456  1.5061  0.2953  1.2041 -0.0118
[221]  0.7895 -1.4036  1.5421 -1.7749  0.2057 -0.3462 -1.1590  0.3358  0.3322  1.4112  0.1369
[232]  0.5776  0.2431  1.4677  1.1306  0.0107  0.1243  1.7794 -0.2395  0.8580 -0.8784 -0.7619
[243]  0.8628  0.6483  1.0581 -0.6332  1.0180  0.1719 -0.0475  1.6891  1.4370 -2.2511  0.3565
[254] -0.8502 -0.2996 -0.6342  1.6245  1.2411  0.5553  0.7034  0.4582  0.6840  0.2513 -0.1785
[265]  0.5077 -0.3099 -0.3944 -0.2697 -0.0881  0.0080  2.5318 -1.2232 -0.0718  0.2461 -0.0506
[276] -0.7302  0.3270  0.7530 -1.1537 -0.4079 -1.2879  0.0836  0.1638  0.6826 -1.0864  0.2975
[287] -0.1433  1.3920  0.3067 -0.5372 -0.2289 -0.5362  1.4390 -0.5111 -1.6068 -0.2012  1.1435
[298]  0.6633  0.1641  1.7854 -0.5877  0.2590 -0.8718 -0.7879 -0.3443  0.6476  2.0541  0.7989
[309] -1.0711 -0.2052 -0.5544 -0.2929  1.1802  0.3774  0.9916  0.6035 -0.7929  0.9308 -1.3504
[320]  0.7998  0.5996  1.1004 -1.4197 -0.2036  0.0179  0.1004  0.8764  0.7007  0.6520  0.1785
[331]  1.6760 -0.3251  0.1011 -0.5767 -0.0615  0.7256 -1.2273  0.1909 -0.7348  0.7884 -1.9654
[342] -1.9555  1.6243 -0.1494 -1.9683 -1.6861 -0.0869  0.3074  0.3375  1.0670  0.3329 -0.0735
[353]  1.0478  2.0589  0.0267 -0.2405  1.0385 -0.5072 -1.4831 -0.1065  0.2453  0.1458 -1.1181
[364] -0.5969 -0.6795  0.9551 -0.1308  0.6494  0.5985 -0.1925 -1.9808  0.3629  0.7039 -0.7526
[375] -0.0169 -0.4417  1.7013 -0.0306  0.2039 -1.0478 -0.7354 -1.3566  0.7286  0.6072 -0.8283
[386]  2.8876  0.9191  0.4058  1.9162 -2.4557 -0.0911 -0.0247  0.1999 -1.0383  0.5294 -2.2171
[397]  0.1074  0.4389 -0.0410  0.3643 -0.7637  0.5495 -0.3746 -1.6592 -0.3102 -0.6199 -1.0246
[408]  0.5050  0.0405 -0.2547 -0.5477  1.6766  1.1730 -0.1891  0.0866  1.5126  1.7933  1.1026
```

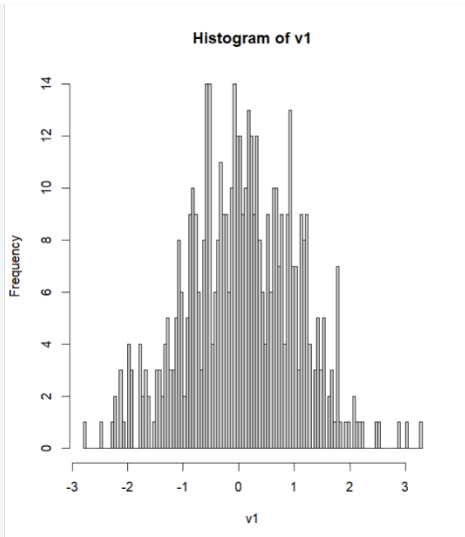
```
> hist(v1)
```




```
> hist(v1, 5)
```

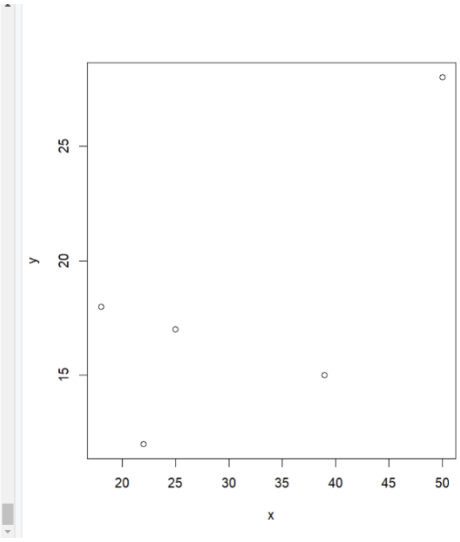



```
> hist(v1, 100)
```



6)

```
> source("myscript.txt")
```



7)

```
> 10
[1] 10
> -2
[1] -2
> 0.02
[1] 0.02
> 1.5e2
[1] 150
> 1e-7
[1] 1e-07
> 0XF
[1] 15
> 0XFA
[1] 250
> 0x1.1p1
[1] 2.125
>
```

```
> 123L
[1] 123
> 1L
[1] 1
> 0x10L
[1] 16
> 1000000L
[1] 1000000
> 1e6L
[1] 1000000
> 1.1L
[1] 1.1
Warning message:
integer literal 1.1L contains decimal; using numeric value
> 1.1L
[1] 1.1
Warning message:
integer literal 1.1L contains decimal; using numeric value
> 1e-3L
[1] 0.001
Warning message:
non-integer value 1e-3L qualified with L; using numeric value
> 0x1.1p-2
[1] 0.265625
> typeof(2)
[1] "double"
> typeof(2L)
[1] "integer"
>
```

```
> TRUE
[1] TRUE
> FALSE
[1] FALSE
>
```

```
> 2i
[1] 0+2i
> 2+4.1i
[1] 2+4.1i
> 1i
[1] 0+1i
> 1e-2i
[1] 0+0.01i
>
```

```
> name = "Anne"
> name
[1] "Anne"
> name = "Anne Mary"
> name
[1] "Anne Mary"
>
```

```
> NULL
NULL
> NA
[1] NA
> inf
Error: object 'inf' not found
> inf
Error: object 'inf' not found
> Inf
[1] Inf
> NaN
[1] NaN
> 1/0
[1] Inf
> 0//1
Error: unexpected '/' in "0//"
> 0/1
[1] 0
> 0/0
[1] NaN
> -2/0
[1] -Inf
>
```

8)

```
> 2>1
[1] TRUE
> 4<=3
[1] FALSE
> "Mary" == "Mary"
[1] TRUE
> "Monday" == "monday"
[1] FALSE
> 2 == 2L
[1] TRUE
> identical(2, 2L)
[1] FALSE
> !(TRUE)
[1] FALSE
> !(2>1)
[1] FALSE
> (2>1)&(5<10)
[1] TRUE
> ("A" == "A") | ("B" == "b")
[1] TRUE
>
```

10)

```
>
> sumproduct = function(a = 1, b = 2, c)
+ return(list(sum = a+b+c, product = a*b*c))
> sumproduct(1,2,3)
$sum
[1] 6

$product
[1] 6

> sumproduct
function(a = 1, b = 2, c)
return(list(sum = a+b+c, product = a*b*c))
> args(sumproduct)
function (a = 1, b = 2, c)
NULL
>
```

11)

```
> installed.packages()
             Package
base         "base"
boot         "boot"
class        "class"
cluster      "cluster"
codetools    "codetools"
compiler     "compiler"
datasets     "datasets"
foreign      "foreign"
graphics     "graphics"
grDevices    "grDevices"
grid         "grid"
KernSmooth   "KernSmooth"
lattice      "lattice"
MASS         "MASS"
Matrix       "Matrix"
methods      "methods"
mgcv         "mgcv"
nlme         "nlme"
nnet         "nnet"
parallel     "parallel"
rpart        "rpart"
spatial      "spatial"
splines      "splines"
stats        "stats"
stats4       "stats4"
survival     "survival"
tcltk        "tcltk"
tools        "tools"
translations "translations"
utils        "utils"
             LibPath
base         "C:/Program Files/R/R-4.3.2/library"
boot         "C:/Program Files/R/R-4.3.2/library"
class        "C:/Program Files/R/R-4.3.2/library"
cluster      "C:/Program Files/R/R-4.3.2/library"
codetools    "C:/Program Files/R/R-4.3.2/library"
compiler     "C:/Program Files/R/R-4.3.2/library"
datasets     "C:/Program Files/R/R-4.3.2/library"
foreign      "C:/Program Files/R/R-4.3.2/library"
```