Exercise 0: Basic arithmetic

MA615/415

September 11, 2016

Exercises 0. Basics

1. Assignment, variables

Try some simple commands

```
    a <- 2
    b <- 50
    c <- 3

    a+b
```

```
## [1] 52
```

```
    b/a
```

```
## [1] 25
```

```
    50%%3
```

```
## [1] 2
```

```
    sin(0)
```

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

```
    cos(0)
```

```
## [1] 1
```

```
    sin45 <- sin(pi/4)
    4*asin(sin45)
```

```
## [1] 3.141593
```

```
    e3 <- exp(3)
    log(e3)
```

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

You can find a list of operators here: R Language Definition, section 3.1.4.

2. Try some basic Vector operations

```r
# create
a_vec <- c(3, 50, 5, 6, 10, 40, 20, 2, 100, 30, 25, 80 )
# length
length(a_vec)
```

```
## [1] 12
```

```r
# sum
sum(a_vec)
```

```
## [1] 371
```

```r
# cumulative sum
cumsum(a_vec)
```

```
##  [1]   3  53  58  64  74 114 134 136 236 266 291 371
```

```r
# summary statistics
summary(a_vec)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    5.75   22.50   30.92   42.50  100.00
```

```r
# variance
var(a_vec)
```

```
## [1] 1011.72
```

```r
# is it reatlly a vector?
str(a_vec)
```

```
##  num [1:12] 3 50 5 6 10 40 20 2 100 30 ...
```

```r
# was that not clear?
is.vector(a_vec)
```

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

```r
# assign the 5th element of a_vec to d
d <- a_vec[5]
a_vec
```

```
##  [1]   3  50   5   6  10  40  20   2 100  30  25  80
```

```r
# what does this mean?
length(d)
```

```
## [1] 1
```

```
    # Really?  Well, ...
    is.vector(d)
```

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

```
    # how about random numbers?
    random_vec <- runif(20)
    length(random_vec)
```

```
## [1] 20
```

```
    random_vec
```

```
##  [1] 0.09476575 0.07373983 0.32019838 0.22392460 0.70281761 0.95796753
##  [7] 0.59538778 0.69474385 0.72035028 0.04131636 0.38034970 0.77432046
## [13] 0.91653545 0.31234795 0.89253305 0.67020925 0.12219199 0.05021232
## [19] 0.88804299 0.56952657
```

```
    # what if random_vec was really big?
    random_vec_big <-  runif(10000)
    length(random_vec_big)
```

```
## [1] 10000
```

```
    head(random_vec_big)
```

```
## [1] 0.91763625 0.94532263 0.06001194 0.35077978 0.91305539 0.58709097
```

```
    # need more than 6?

    head(random_vec_big, 20)
```

```
##  [1] 0.917636247 0.945322625 0.060011945 0.350779785 0.913055389
##  [6] 0.587090966 0.025925874 0.998129319 0.148178514 0.168042855
## [11] 0.922037861 0.345945783 0.677181285 0.899191394 0.004142052
## [16] 0.099861941 0.621276136 0.079525674 0.817814041 0.127502363
```

```
    # too many decimal places?

    options(digits = 2)
    tail(random_vec_big)
```

```
## [1] 0.838 0.014 0.675 0.156 0.576 0.146
```

```
    # What if I want my random series to be repeatable?

    set.seed(2)
    r <- runif(30)
    head(r,10)
```

```
##  [1] 0.18 0.70 0.57 0.17 0.94 0.94 0.13 0.83 0.47 0.55
```

```
    r1 <- runif(20)
    head(r1,10)
```

```
##  [1] 0.01 0.16 0.81 0.87 0.51 0.63 0.84 0.28 0.67 0.15
```

```
    set.seed(2)
    s <- runif(10)
    s
```

```
##  [1] 0.18 0.70 0.57 0.17 0.94 0.94 0.13 0.83 0.47 0.55
```

```
    # not enough digits?

    options(digits = 4)
    head(r)
```

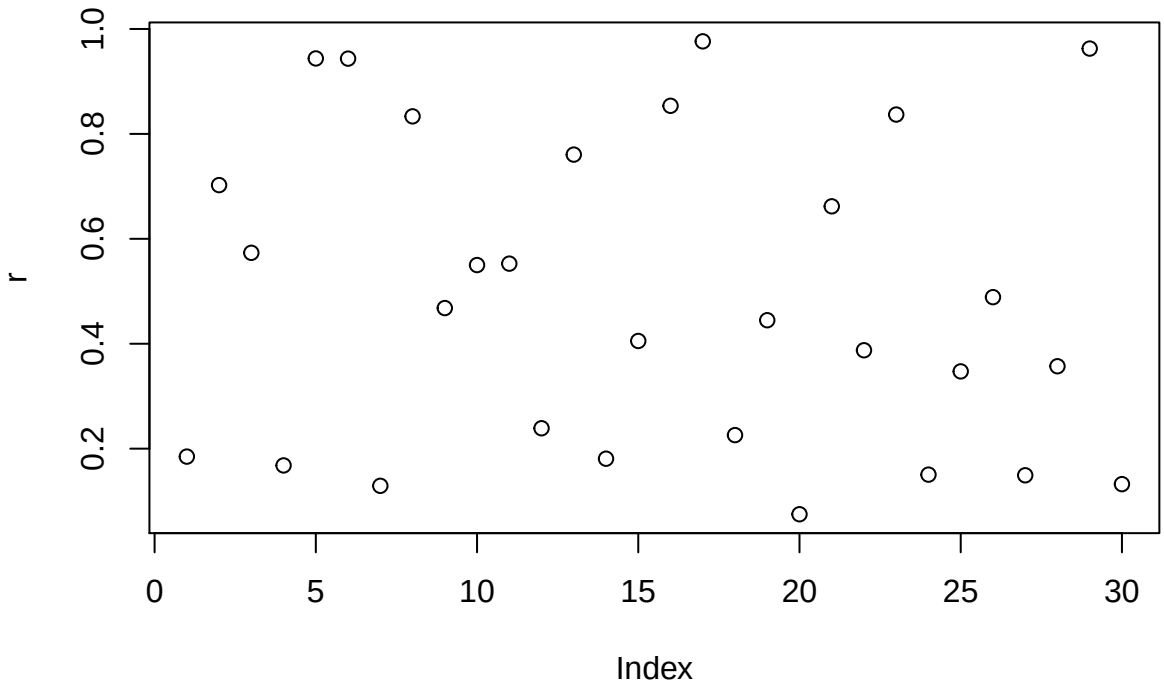
```
## [1] 0.1849 0.7024 0.5733 0.1681 0.9438 0.9435
```

```
    head(s)
```

```
## [1] 0.1849 0.7024 0.5733 0.1681 0.9438 0.9435
```

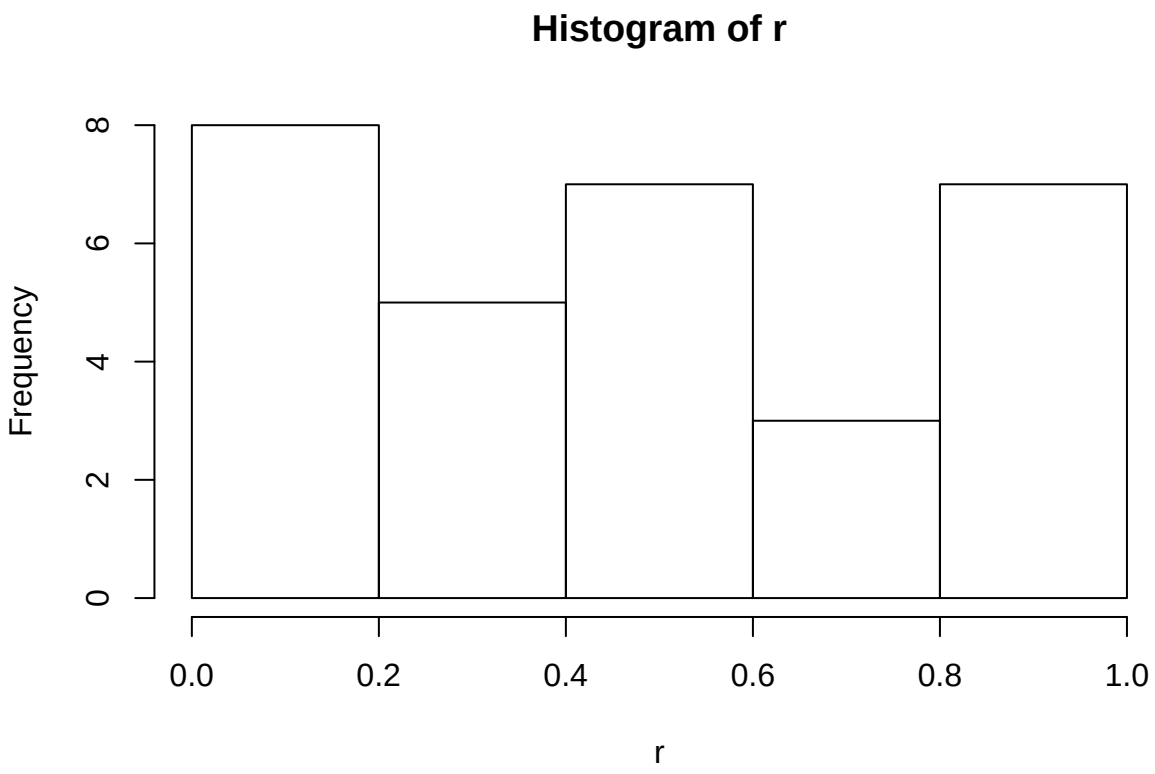
```
    # can I get a picture of that?

    plot(r)
```



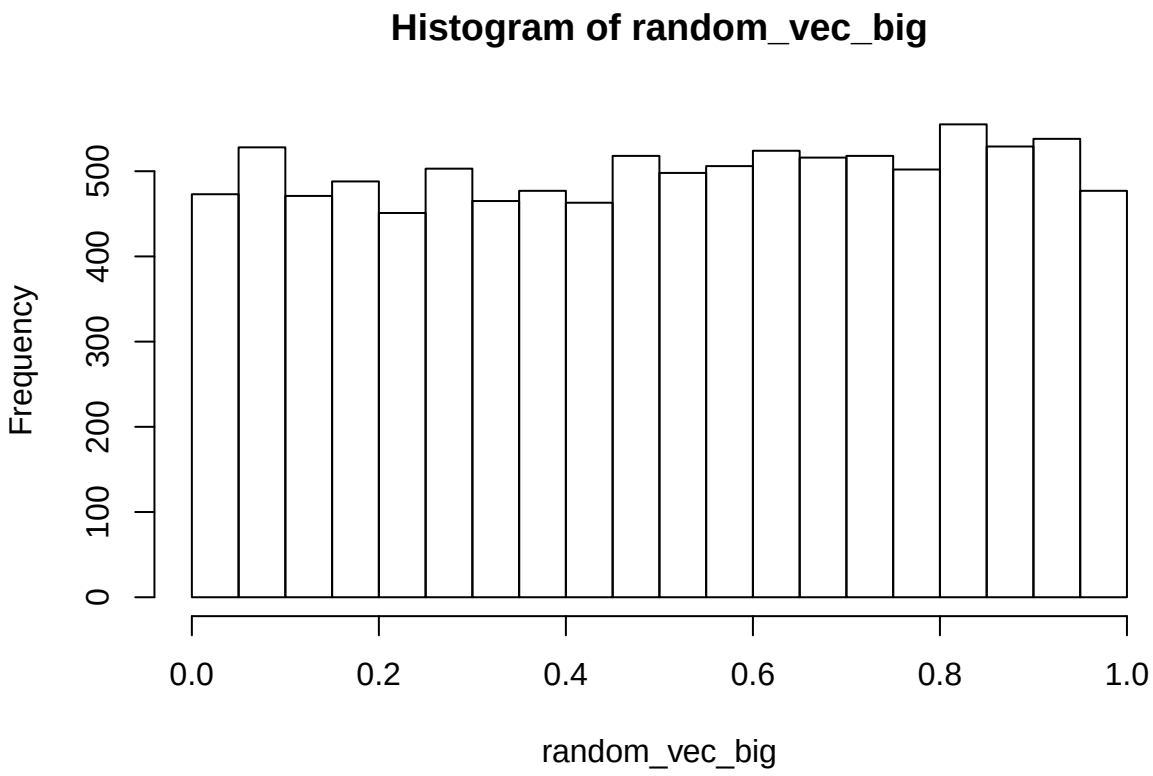
```r
# that's not very helpful.  try this --

hist(r)
```




```r
# doesn't look uniform?  increase the sample size

hist(random_vec_big)
```

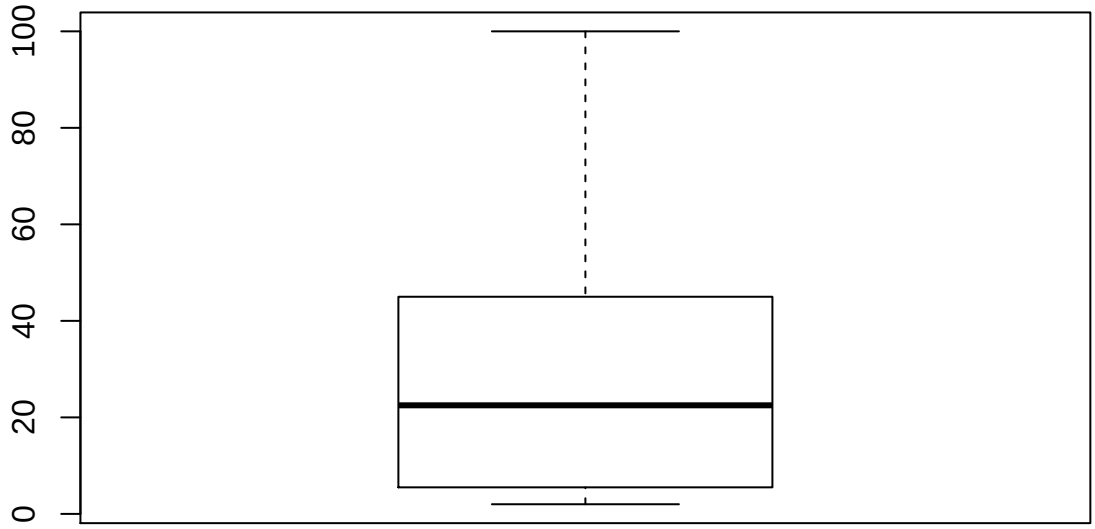


All these blocks of R may not be what you need.

Remember a_vec? Sure, avec is (3, 50, 5, 6, 10, 40, 20, 2, 100, 30, 25, 80). Here's a_vec's mean 30.9167.

Wow – a_vec is a strange set of data, what does it look like?

```
boxplot(a_vec)
```



3. Finally, let's look at matrice, lists, and data.frames

```
# make a random matrix
A <- matrix(data = floor(30*runif(16)), nrow = 4)
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   16   12   13   25
## [2,]    7   25    2    4
## [3,]   22   29   19   10
## [4,]    5    6   11   14
```

```
# how big is the matrix?
dim(A)
```

```
## [1] 4 4
```

```
# Let's make a vector to multiply by A

b <- floor(20*runif(dim(A)[2]))
b
```

```
## [1]  2  7 19  2
```

```r
 # solve Ax = b

 x <- solve(A,b)

 x
```

```
## [1]  0.2828  0.2370  0.5835 -0.5182
```

```r
b1 <- A%*%x

b1
```

```
##      [,1]
## [1,]    2
## [2,]    7
## [3,]   19
## [4,]    2
```

```r
b2 <- as.vector(A%*%x)

b2
```

```
## [1]  2  7 19  2
```

```r
# inner product  dot(e,f) = e'f

e <- floor(20*runif(4))
f <- floor(20*runif(4))

e
```

```
## [1]  0  3 16 17
```

```r
f
```

```
## [1] 10 12 16  5
```

```r
crossprod(e,f)
```

```
##      [,1]
## [1,]  377
```

```r
sum(e*f)
```

```
## [1] 377
```

```r
t(e)%*%f
```

```
##      [,1]
## [1,]  377
```

```r
# outer product

outer(e,f,"*")
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    0    0    0    0
## [2,]   30   36   48   15
## [3,]  160  192  256   80
## [4,]  170  204  272   85
```

```r
outer(e,f,">")
```

```
##       [,1]  [,2]  [,3]  [,4]
## [1,] FALSE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE
## [3,]  TRUE  TRUE FALSE  TRUE
## [4,]  TRUE  TRUE  TRUE  TRUE
```

```r
#using automatic coersion

outer(e,f,">")*1
```

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