CDT Module 1 - R Review

Valerie Bradley

6/10/2018

1. Sequences.

Generate the following sequences using rep(), seq() and arithmetic:

(a) 1,3,5,7,. . . ,21.

```
seq(1,21,2)
```

```
##  [1]  1  3  5  7  9 11 13 15 17 19 21
```

(b) 1,10,100,. . . ,10^9.

```
10^seq(0,9,1)
```

```
##  [1]          1         10        100       1000      10000     100000
##  [7]    1000000   10000000  100000000 1000000000
```

(c) 0,1,2,3,0,. . . ,3,0,1,2,3 [with each entry appearing 6 times]

```
rep(seq(0,3,1),6)
```

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

(d) 0,0,0,1,1,1,2,. . . ,4,4,4.

```
sort(rep(seq(0,4,1),6))
```

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

(e) 50,47,44,. . . ,14,11.

```
seq(50,11,-3)
```

```
##  [1] 50 47 44 41 38 35 32 29 26 23 20 17 14 11
```

(f) 1,2,5,10,20,50,100,. . . ,5×10^4.

2. Arithmetic

(a) $cos\left(\frac{\pi n}{3}\right)$, for $n = 0, ..., 10$

```
n = 0:10
cos(pi * n / 3)
```

```
##  [1]  1.0  0.5 -0.5 -1.0 -0.5  0.5  1.0  0.5 -0.5 -1.0 -0.5
```

(b) $1, 9, 98, 997, ..., 999994$.

```
x = seq(0,6,1)
10^x - x
```

```
## [1]      1      9     98    997   9996  99995 999994
```

(c) $e^n - 3n$,for $n = 0, ..., 10$.

```r
n = 0:10
exp(1)^n - 3*n
```

```
##  [1]     1.0000000    -0.2817182     1.3890561    11.0855369    42.5981500
##  [6]   133.4131591   385.4287935  1075.6331584  2956.9579870  8076.0839276
## [11] 21996.4657948
```

(d) $3n$ mod 7, for $n = 0, ..., 10$.

```r
n = 0:10
(3 * n) %% 7
```

```
##  [1] 0 3 6 2 5 1 4 0 3 6 2
```

(e) Let

$$S_n = \sum_{i=1}^{n} \frac{(-1)^{i+1}}{2i-1} \lim_{n \to \text{inf}} Sn = \frac{\pi}{4}$$

evaluate $4S_{10}$, $4S_{100}$ and $4S_{1000}$

```r
myseq = function(n){
  i = seq(1,n,1)
  4*sum((-1)^(i+1)/(2*i - 1))
}

data.frame(s_10 = myseq(10), s_100 = myseq(100), s_1000 = myseq(1000))
```

```
##      s_10    s_100   s_1000
## 1 3.04184 3.131593 3.140593
```

3. Subsetting

```r
set.seed(123)
x = rnorm(100)
```

(a) the 25th, 50th and 75th elements;

```r
x[c(25,50,75)]
```

```
## [1] -0.62503927 -0.08336907 -0.68800862
```

(b) the first 25 elements;

```r
x[1:25]
```

```
##  [1] -0.56047565 -0.23017749  1.55870831  0.07050839  0.12928774
##  [6]  1.71506499  0.46091621 -1.26506123 -0.68685285 -0.44566197
## [11]  1.22408180  0.35981383  0.40077145  0.11068272 -0.55584113
## [16]  1.78691314  0.49785048 -1.96661716  0.70135590 -0.47279141
## [21] -1.06782371 -0.21797491 -1.02600445 -0.72889123 -0.62503927
```

(c) all elements except those from the 31st to the 40th.

```r
x[-(31:40)]
```

```
##  [1] -0.560475647 -0.230177489  1.558708314  0.070508391  0.129287735
##  [6]  1.715064987  0.460916206 -1.265061235 -0.686852852 -0.445661970
## [11]  1.224081797  0.359813827  0.400771451  0.110682716 -0.555841135
## [16]  1.786913137  0.497850478 -1.966617157  0.701355902 -0.472791408
```

```
## [21] -1.067823706 -0.217974915 -1.026004448 -0.728891229 -0.625039268
## [26] -1.686693311  0.837787044  0.153373118 -1.138136937  1.253814921
## [31] -0.694706979 -0.207917278 -1.265396352  2.168955965  1.207961998
## [36] -1.123108583 -0.402884835 -0.466655354  0.779965118 -0.083369066
## [41]  0.253318514 -0.028546755 -0.042870457  1.368602284 -0.225770986
## [46]  1.516470604 -1.548752804  0.584613750  0.123854244  0.215941569
## [51]  0.379639483 -0.502323453 -0.333207384 -1.018575383 -1.071791226
## [56]  0.303528641  0.448209779  0.053004227  0.922267468  2.050084686
## [61] -0.491031166 -2.309168876  1.005738524 -0.709200763 -0.688008616
## [66]  1.025571370 -0.284773007 -1.220717712  0.181303480 -0.138891362
## [71]  0.005764186  0.385280401 -0.370660032  0.644376549 -0.220486562
## [76]  0.331781964  1.096839013  0.435181491 -0.325931586  1.148807618
## [81]  0.993503856  0.548396960  0.238731735 -0.627906076  1.360652449
## [86] -0.600259587  2.187332993  1.532610626 -0.235700359 -1.026420900
```

(d) all values larger than 1.5 (how many are there?);

```
length(x[x>1.5])/100
```

```
## [1] 0.08
```

```
1-pnorm(1.5)
```

```
## [1] 0.0668072
```

(e) what about the entries that are either > 1.5 or < -1?

```
x[x>1.5 | x < -1]
```

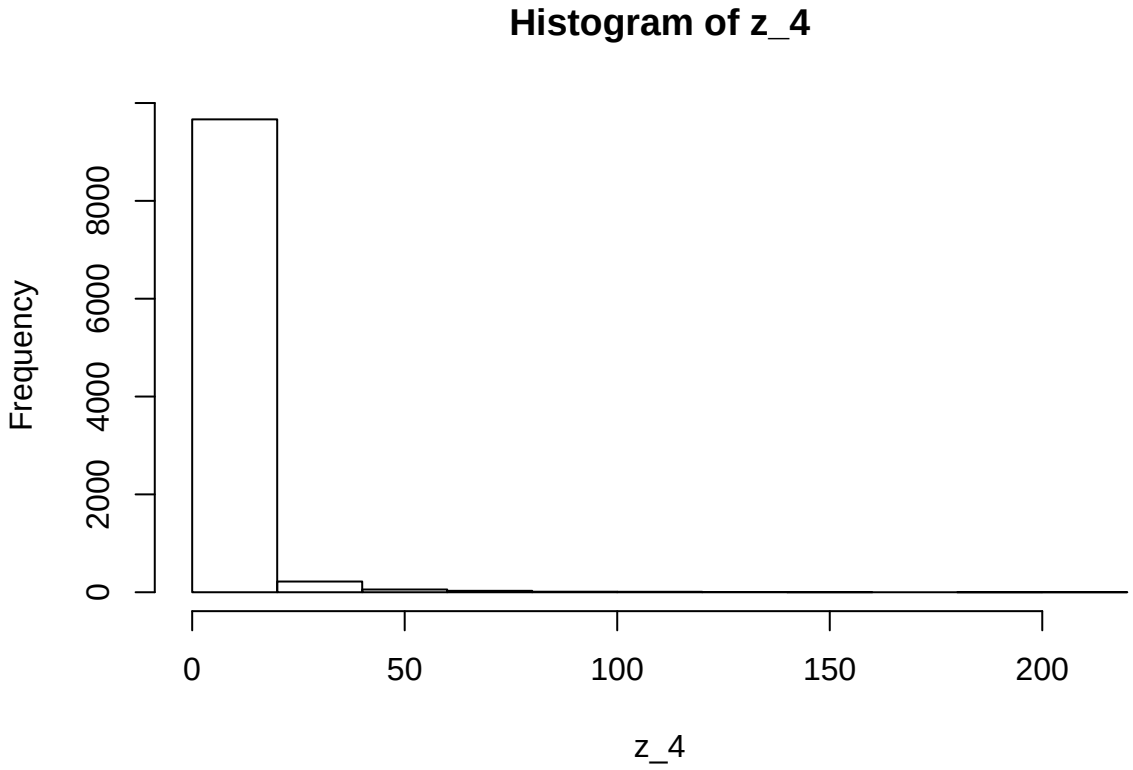
```
##  [1]  1.558708  1.715065 -1.265061  1.786913 -1.966617 -1.067824 -1.026004
##  [8] -1.686693 -1.138137 -1.265396  2.168956 -1.123109  1.516471 -1.548753
## [15] -1.018575 -1.071791  2.050085 -2.309169 -1.220718  2.187333  1.532611
## [22] -1.026421
```

4. Monte Carlo Integration

```
z = rnorm(10000)
z_4 = z^4
mean(z_4)
```

```
## [1] 3.003656
```

```
hist(z_4)
```




```
mean(z_4)
```

```
## [1] 3.003656
```

```
S_sq = 1 / (length(z_4) - 1) * sum((z_4 - mean(z_4))^2)
S_sq
```

```
## [1] 94.64962
```

```
# var(z_4) #check that it's the same

#get conf int
mean(z_4) + qnorm(c(0.025, 0.975)) * sqrt(S_sq)/sqrt(length(z_4))
```

```
## [1] 2.812975 3.194337
```