R Notebook

1. Assign a set of 100 numbers to a variable x. Make sure it includes some positive and some negative numbers. Display the contents of the vetor afterward. Is it really a vector? Enter is.vector(x) to confirm.

```
x<-seq(-9,91, length=100)
x
```

```
##   [1] -9.00000000 -7.98989899 -6.97979798 -5.96969697 -4.95959596 -3.94949495
##   [7] -2.93939394 -1.92929293 -0.91919192  0.09090909  1.10101010  2.11111111
##  [13]  3.12121212  4.13131313  5.14141414  6.15151515  7.16161616  8.17171717
##  [19]  9.18181818 10.19191919 11.20202020 12.21212121 13.22222222 14.23232323
##  [25] 15.24242424 16.25252525 17.26262626 18.27272727 19.28282828 20.29292929
##  [31] 21.30303030 22.31313131 23.32323232 24.33333333 25.34343434 26.35353535
##  [37] 27.36363636 28.37373737 29.38383838 30.39393939 31.40404040 32.41414141
##  [43] 33.42424242 34.43434343 35.44444444 36.45454545 37.46464646 38.47474747
##  [49] 39.48484848 40.49494949 41.50505051 42.51515152 43.52525253 44.53535354
##  [55] 45.54545455 46.55555556 47.56565657 48.57575758 49.58585859 50.59595960
##  [61] 51.60606061 52.61616162 53.62626263 54.63636364 55.64646465 56.65656566
##  [67] 57.66666667 58.67676768 59.68686869 60.69696970 61.70707071 62.71717172
##  [73] 63.72727273 64.73737374 65.74747475 66.75757576 67.76767677 68.77777778
##  [79] 69.78787879 70.79797980 71.80808081 72.81818182 73.82828283 74.83838384
##  [85] 75.84848485 76.85858586 77.86868687 78.87878788 79.88888889 80.89898990
##  [91] 81.90909091 82.91919192 83.92929293 84.93939394 85.94949495 86.95959596
##  [97] 87.96969697 88.97979798 89.98989899 91.00000000
```

```
is.vector(x)
```

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

2. Print the 3rd and 6th elements of x with a single command.

```
x[c(3,6)]
```

```
## [1] -6.979798 -3.949495
```

3. Print (i.e. display) all elements of x that are non-negative.

```
x[x>0]
```

```
##  [1]  0.09090909  1.10101010  2.11111111  3.12121212  4.13131313  5.14141414
##  [7]  6.15151515  7.16161616  8.17171717  9.18181818 10.19191919 11.20202020
## [13] 12.21212121 13.22222222 14.23232323 15.24242424 16.25252525 17.26262626
## [19] 18.27272727 19.28282828 20.29292929 21.30303030 22.31313131 23.32323232
## [25] 24.33333333 25.34343434 26.35353535 27.36363636 28.37373737 29.38383838
## [31] 30.39393939 31.40404040 32.41414141 33.42424242 34.43434343 35.44444444
## [37] 36.45454545 37.46464646 38.47474747 39.48484848 40.49494949 41.50505051
## [43] 42.51515152 43.52525253 44.53535354 45.54545455 46.55555556 47.56565657
## [49] 48.57575758 49.58585859 50.59595960 51.60606061 52.61616162 53.62626263
## [55] 54.63636364 55.64646465 56.65656566 57.66666667 58.67676768 59.68686869
## [61] 60.69696970 61.70707071 62.71717172 63.72727273 64.73737374 65.74747475
## [67] 66.75757576 67.76767677 68.77777778 69.78787879 70.79797980 71.80808081
## [73] 72.81818182 73.82828283 74.83838384 75.84848485 76.85858586 77.86868687
## [79] 78.87878788 79.88888889 80.89898990 81.90909091 82.91919192 83.92929293
## [85] 84.93939394 85.94949495 86.95959596 87.96969697 88.97979798 89.98989899
## [91] 91.00000000
```

4.Change the last value of your x vector to a different number. Change the 2nd, 6th, and 10th values of x to 1, 2, 3 with a single command.

```
x[100] <- 789

x
```

```
##   [1]  -9.00000000  -7.98989899  -6.97979798  -5.96969697  -4.95959596
##   [6]  -3.94949495  -2.93939394  -1.92929293  -0.91919192   0.09090909
##  [11]   1.10101010   2.11111111   3.12121212   4.13131313   5.14141414
##  [16]   6.15151515   7.16161616   8.17171717   9.18181818  10.19191919
##  [21]  11.20202020  12.21212121  13.22222222  14.23232323  15.24242424
##  [26]  16.25252525  17.26262626  18.27272727  19.28282828  20.29292929
##  [31]  21.30303030  22.31313131  23.32323232  24.33333333  25.34343434
##  [36]  26.35353535  27.36363636  28.37373737  29.38383838  30.39393939
##  [41]  31.40404040  32.41414141  33.42424242  34.43434343  35.44444444
##  [46]  36.45454545  37.46464646  38.47474747  39.48484848  40.49494949
##  [51]  41.50505051  42.51515152  43.52525253  44.53535354  45.54545455
##  [56]  46.55555556  47.56565657  48.57575758  49.58585859  50.59595960
##  [61]  51.60606061  52.61616162  53.62626263  54.63636364  55.64646465
##  [66]  56.65656566  57.66666667  58.67676768  59.68686869  60.69696970
##  [71]  61.70707071  62.71717172  63.72727273  64.73737374  65.74747475
##  [76]  66.75757576  67.76767677  68.77777778  69.78787879  70.79797980
##  [81]  71.80808081  72.81818182  73.82828283  74.83838384  75.84848485
##  [86]  76.85858586  77.86868687  78.87878788  79.88888889  80.89898990
##  [91]  81.90909091  82.91919192  83.92929293  84.93939394  85.94949495
##  [96]  86.95959596  87.96969697  88.97979798  89.98989899 789.00000000
```

```
x[c(2,6,10)] <- c(1,2,3)
x
```

```
##   [1]  -9.0000000   1.0000000  -6.9797980  -5.9696970  -4.9595960   2.0000000
##   [7]  -2.9393939  -1.9292929  -0.9191919   3.0000000   1.1010101   2.1111111
##  [13]   3.1212121   4.1313131   5.1414141   6.1515152   7.1616162   8.1717172
##  [19]   9.1818182  10.1919192  11.2020202  12.2121212  13.2222222  14.2323232
##  [25]  15.2424242  16.2525253  17.2626263  18.2727273  19.2828283  20.2929293
##  [31]  21.3030303  22.3131313  23.3232323  24.3333333  25.3434343  26.3535354
##  [37]  27.3636364  28.3737374  29.3838384  30.3939394  31.4040404  32.4141414
##  [43]  33.4242424  34.4343434  35.4444444  36.4545455  37.4646465  38.4747475
##  [49]  39.4848485  40.4949495  41.5050505  42.5151515  43.5252525  44.5353535
##  [55]  45.5454545  46.5555556  47.5656566  48.5757576  49.5858586  50.5959596
##  [61]  51.6060606  52.6161616  53.6262626  54.6363636  55.6464646  56.6565657
##  [67]  57.6666667  58.6767677  59.6868687  60.6969697  61.7070707  62.7171717
##  [73]  63.7272727  64.7373737  65.7474747  66.7575758  67.7676768  68.7777778
##  [79]  69.7878788  70.7979798  71.8080808  72.8181818  73.8282828  74.8383838
##  [85]  75.8484848  76.8585859  77.8686869  78.8787879  79.8888889  80.8989899
##  [91]  81.9090909  82.9191919  83.9292929  84.9393939  85.9494949  86.9595960
##  [97]  87.9696970  88.9797980  89.9898990 789.0000000
```

5. For each of the following examples determine why the code does not work and correct it. Use R's built in help functions if you have trouble on the second set.

```
#my_variable <- 10
x <- 10
#my_variable
x
```

```
## [1] 10
```

```
x <- c(2,34,61,21,NA ,32)
y<-c(5,56,789,23,3,90)
?mean
```

```
## starting httpd help server ... done
```

```
z <- mean(x*y)
z <- mean(x*y,trim = 0,na.rm = TRUE)
z
```

```
## [1] 10681.2
```