

Homework on Vectors

Samriddh Gupta

1. Describe the difference between `is.finite(x)` and `!is.infinite(x)`.

```
x <- c(0,1,2,3,4,5,10, NA, Inf, -Inf)
is.finite(x)
```

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

```
!is.infinite(x)
```

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

So `is.finite()` consider the NA to be not a number ,but the `!is.infinite` consider NA to be a number. I think probably '0'

2. Read the source code for `dplyr::near()` (Hint: to see the source code, drop the `()`). How does it work?

```
dplyr::near
```

```
## function (x, y, tol = .Machine$double.eps^0.5)
## {
##   abs(x - y) < tol
## }
## <bytecode: 0x00000000147963d8>
## <environment: namespace:dplyr>
```

it takes the number and check for the 0.5 range. if it is less than 0.5 then it goes to left side of the number line to the integer and if it more then it will go to the right side of the number line.

3. Create functions that take a vector as input and returns:

- a. The last value. Should you use `[` or `[[`?
- b. The elements at even numbered positions.
- c. Every element except the last value.
- d. Only even numbers (and no missing values).

```
f1<- function(x){
  return(x[length(x)])
}
```

```
x<-c(1,2,3,4)
y<-c(1,2,3,4,Inf)
f1(x)
```

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

```
f1(y)
```

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

```
f2<- function(x){  
  for (i in 1:length(x)){  
    if (i%%2==0){  
      print(x[i])  
    }  
  }  
}  
  
x<-c(1,2,3,4,6,4,3,2,1,3,5,6,5,12)  
y<-c(1,2,3,4,Inf,"a")  
f2(x)
```

```
## [1] 2  
## [1] 4  
## [1] 4  
## [1] 2  
## [1] 3  
## [1] 6  
## [1] 12
```

```
f2(y)
```

```
## [1] "2"  
## [1] "4"  
## [1] "a"
```

```
f3<- function(x){  
  return(x[1:length(x)-1])  
}  
  
x<-c(1,2,3,4)  
y<-c(1,2,3,4,Inf)  
f3(x)
```

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

```
f3(y)
```

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

```
f4<- function(x){  
  x[which(x%%2==0)]  
}  
  
x<-c(1,2,3,4,6,4,3,2,1,3,5,6,5,12,NA)  
y<-c(1,2,3,4,Inf,NA,13,45,77,888)  
f4(x)
```

```
## [1] 2 4 6 4 2 6 12
```

```
f4(y)
```

```
## [1] 2 4 888
```

5. Why is `x[-which(x > 0)]` not the same as `x[x <= 0]`?

```
x<-c(1,2,-3,-4,-6,4,-3,2,-1,0,3,5,6,5,12,NA,-Inf,NaN)
x[-which(x > 0)]
```

```
## [1] -3 -4 -6 -3 -1 0 NA -Inf NaN
```

```
x[x <= 0]
```

```
## [1] -3 -4 -6 -3 -1 0 NA -Inf NA
```

I think that both of them are pretty same except for the fact that the NaN is viewed by them differently. and I think that is the main difference here.

6. What happens when you subset with a positive integer that's bigger than the length of the vector?
What happens when you subset with a name that doesn't exist?

```
x <- c(1:20)
x[30]
```

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

```
x <- c(a = 1, b = 2,x=3)
x["v"]
```

```
## <NA>
## NA
```

If the range of the value is bigger than what you have in the vector than it gives you a missing value. same happens with the subset of the name that goes not exist . it gives you a missing value or NA

6. `z <- list(a=c(2,4,6), b = "a rather long string", c = pi, d = list(1,2,3))` Write commands to do each of the following: Return the number pi Return the number 3 Return the vector 2,4,6

```
z <- list(a=c(2,4,6), b = "a rather long string", c = pi, d = list(1,2,3))
z$c
```

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

```
z$d[3]
```

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

```
z$a
```

```
## [1] 2 4 6
```