Assignment on Exercise 4-6

Alex

14/09/2019

Chapter 4 Exercise 4.1

Exercise 4.2

```
# Question a
foo <- c(7,1,7,10,5,9,10,3,10,8)
foo
```

```
##  [1]  7  1  7 10  5  9 10  3 10  8
```

```
A <- foo[foo > 5|foo == 2]
A
```

```
## [1]  7  7 10  9 10 10  8
```

```
# Question b
bar <- c(8,8,4,4,5,1,5,6,6,8)
bar
```

```
##  [1] 8 8 4 4 5 1 5 6 6 8
```

```
E <- bar[bar<6|bar==6 & bar!=4]
E
```

```
## [1] 4 4 5 1 5 6 6
```

```
# Question c
C <- A & E
C
```

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

```
# Question d
baz <- foo + bar
baz
```

```
##  [1] 15  9 11 14 10 10 15  9 16 16
```

```
# i
bazE <- baz[baz >= 14 & baz!=15]
bazE
```

```
## [1] 14 16 16
```

```r
#ii
V <- baz[baz/foo]
V
```

```
##  [1]  9 16 15 15  9 15 15 11 15  9
```

```r
VE <- V[V > 4| V <= 2]
VE
```

```
##  [1]  9 16 15 15  9 15 15 11 15  9
```

```r
# Question e
A2 <- foo[foo > 5||foo == 2]
E2 <- bar[bar<6|bar==6 && bar!=4]
C2 <- A && E
bazE2 <- baz[baz >= 14 && baz!=15]
VE2 <- V[V > 4|| V <= 2]

A2
```

```
##  [1]  7  1  7 10  5  9 10  3 10  8
```

```r
E2
```

```
## [1] 4 4 5 1 5
```

```r
C2
```

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

```r
bazE2
```

```
## numeric(0)
```

```r
VE2
```

```
##  [1]  9 16 15 15  9 15 15 11 15  9
```

#Exercise 4.3

```r
# Question a
foo <- c(7,5,6,1,2,10,8,3,8,2)
#i elements greater or equal  to 5
bar <- foo[(foo > 5)| (foo == 5)]
bar
```

```
## [1]  7  5  6 10  8  8
```

```r
#ii left over
Res <- foo[foo<=5]
Res
```

```
## [1] 5 1 2 3 2
```

```r
# Question b
baz <- matrix(bar,nrow = 2,ncol = 3)
baz
```

```
##      [,1] [,2] [,3]
## [1,]    7    6    8
## [2,]    5   10    8
```

```r
#i.
baz[c(1,2),3] <- baz[1,2] ^2
baz
```

```
##      [,1] [,2] [,3]
## [1,]    7    6   36
## [2,]    5   10   36
```

```r
#ii
baz[c(1,2),3] <- 25
baz
```

```
##      [,1] [,2] [,3]
## [1,]    7    6   25
## [2,]    5   10   25
```

```r
# Question c
qux <- array(c(10,5,1,4,7,4,3,3,1,3,4,3,1,7,8,3,7,3), dim = c(3,2,3))
qux
```

```
## , , 1
##
##      [,1] [,2]
## [1,]   10    4
## [2,]    5    7
## [3,]    1    4
##
## , , 2
##
##      [,1] [,2]
## [1,]    3    3
## [2,]    3    4
## [3,]    1    3
##
## , , 3
##
##      [,1] [,2]
## [1,]    1    3
## [2,]    7    7
## [3,]    8    3
```

```r
#i identify dim-specific index positions
di <- which(qux== 3|qux ==4)
di
```

```
## [1]  4  6  7  8 10 11 12 16 18
```

```r
#ii Replace all elements in qux
qux[qux< 3 | qux >=7]<- 100
qux
```

```
## , , 1
##
##      [,1] [,2]
## [1,]  100    4
## [2,]    5  100
## [3,]  100    4
##
## , , 2
##
##      [,1] [,2]
## [1,]    3    3
## [2,]    3    4
## [3,]  100    3
##
## , , 3
##
##      [,1] [,2]
## [1,]  100    3
## [2,]  100  100
## [3,]  100    3
```

```r
# Question d
#i extract second value
ex <- foo[c(T,F)]
ex
```

```
## [1] 7 6 2 8 8
```

```r
ex2 <- foo[c(0,1)]
ex2
```

```
## [1] 7
```

```r
Explanation<-(" R returns 7 the first second element only The extraction is limited to only one, this is
Explanation
```

```
## [1] " R returns 7 the first second element only The extraction is limited to only one, this is becaus
```

#Exercise 4.4

```r
# Question a
cat("\"The quick brown fox\n\    jumped over\n\tthe lazy dogs\"")
```

```
## "The quick brown fox
##     jumped over
##  the lazy dogs"
```

```r
# Question b
num1 <- 4
num2 <- 0.75
paste("The result of multiplying",num1,"by",num2, "is", num1 * num2)
```

```
## [1] "The result of multiplying 4 by 0.75 is 3"
```

```r
# Question c
R <- "/Users/tdavies/Documents/RBook/"
sub(pattern = "tdavies", replacement = "aogbole", x=R)
```

```
## [1] "/Users/aogbole/Documents/RBook/"
```

```r
# Question d
bar <- "How much wood could a woodchuck chuck"
#i store a new string
sub(pattern = "How much wood could a woodchuck chuck", replacement = "if a woodchuck could chuck wood",
```

```
## [1] "if a woodchuck could chuck wood"
```

```r
bar
```

```
## [1] "How much wood could a woodchuck chuck"
```

```r
#ii replace all wood instances with metal
gsub(pattern = "wood", replacement = "metal",x=bar)
```

```
## [1] "How much metal could a metalchuck chuck"
```

```r
# Question e
two <- "Two 6-packs for $12.99"
#i .
substr(x=two,start = 5,stop = 10)
```

```
## [1] "6-pack"
```

```r
#ii
substr(x=two,start = 16,stop = 21)<- "$10.99"
two
```

```
## [1] "Two 6-packs for$10.999"
```

#Exercise 4.5

```r
# Question a
sex.char <- c("M","M","M","M","M","M","M","M","M","M","M","F","F","F","F","F","F","F","F")
party.char <- c("National",
"Labour", "Greens", "Maori", "Other")

# Question b
sex.char.fac<- factor(x=sex.char,ordered = TRUE)
sex.char.fac
```

```
##  [1] M M M M M M M M M M M F F F F F F F F
## Levels: F < M
```

```r
party.char.fac <- factor(x=party.char ,ordered = TRUE)
party.char.fac
```

```
## [1] National Labour   Greens   Maori    Other
## Levels: Greens < Labour < Maori < National < Other
```

```r
Exp<-(" It does not make sense as the ordering is based on alphabetical arrangement rather than the unic
Exp
```

```
## [1] " It does not make sense as the ordering is based on alphabetical arrangement rather than the uni
```

```r
# Question c
sex.char.fac<- factor(x=sex.char,ordered = FALSE)
sex.char.fac
```

```
##  [1] M M M M M M M M M M M F F F F F F F F
## Levels: F M
```

```r
party.char.fac <- factor(x=party.char ,ordered = FALSE)
#i.factor vector for only male participants
party.char.fac[sex.char.fac=="M"]
```

```
##  [1] National Labour   Greens   Maori    Other    <NA>     <NA>
##  [8] <NA>     <NA>     <NA>     <NA>
## Levels: Greens Labour Maori National Other
```

```r
#ii Returns genders who choose National
sex.char.fac[party.char.fac=="National"]
```

```
## [1] M M M F
## Levels: F M
```

```r
# Question d
newparty<- c("National","Maori","Maori","Labour","Greens","Labour")
newsex<- c("M","M","F","F","F","M")
sex2<- factor(x=c(newsex),levels=levels(sex.char.fac), ordered=TRUE)
sex2
```

```
## [1] M M F F F M
## Levels: F < M
```

```
party2<- factor(x=c(newparty),levels=levels(party.char.fac), ordered=TRUE)
party2
```

```
## [1] National Maori    Maori    Labour   Greens   Labour
## Levels: Greens < Labour < Maori < National < Other
```

```
c(sex.char.fac,sex2)
```

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

```
c(party.char.fac,party2)
```

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

```
# To order combined gener
levels(sex.char.fac)[c(sex.char.fac,sex2)]
```

```
##  [1] "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "F" "F" "F" "F" "F" "F"
## [18] "F" "F" "M" "M" "F" "F" "F" "M"
```

```
sexcomb <- levels(sex.char.fac)[c(sex.char.fac,sex2)]
sexcomb.fac <- factor(x=sexcomb,levels=levels(sex.char.fac),ordered=TRUE)
sexcomb.fac
```

```
##  [1] M M M M M M M M M M M F F F F F F F M M F F F M
## Levels: F < M
```

```
# To order combined party
partycomb <- levels(party.char.fac)[c(party.char.fac,party2)]
partycomb.fac <- factor(x=partycomb,levels=levels(party.char.fac),ordered=TRUE)
partycomb.fac
```

```
##  [1] National Labour   Greens   Maori    Other    National Maori
##  [8] Maori    Labour   Greens   Labour
## Levels: Greens < Labour < Maori < National < Other
```

```
# Question e
Conf<- c(93, 55, 29, 100,
52, 84, 56, 0, 33, 52, 35, 53, 55, 46, 40, 40, 56, 45, 64, 31, 10, 29, 40,
95, 18, 61)
Per<-c(0,30,70,100)
lab<- c("Low","Moderate","High")
cut(x=Conf,breaks = Per,right = F,include.lowest = T, labels = lab)
```

```
##  [1] High     Moderate Low      High     Moderate High     Moderate
##  [8] Low      Moderate Moderate Moderate Moderate Moderate Moderate
## [15] Moderate Moderate Moderate Moderate Moderate Moderate Low
## [22] Low      Moderate High     Low      Moderate
## Levels: Low Moderate High
```

```
# Question f
Conf.fac<- factor(x=Conf,ordered = TRUE)
Conf.fac
```

```
##  [1] 93  55  29  100 52  84  56  0   33  52  35  53  55  46  40  40  56
## [18] 45  64  31  10  29  40  95  18  61
## 20 Levels: 0 < 10 < 18 < 29 < 31 < 33 < 35 < 40 < 45 < 46 < 52 < ... < 100
```

```
#i.Levels of individuals originally identified with Labour
Conf.fac[party.char.fac=="Labour"]
```

```
## [1] 55 56 53 56 29
## 20 Levels: 0 < 10 < 18 < 29 < 31 < 33 < 35 < 40 < 45 < 46 < 52 < ... < 100
```

```
#ii levels of individuals identified with National
Conf.fac[party.char.fac=="National"]
```

```
## [1] 93 84 35 40 10 61
## 20 Levels: 0 < 10 < 18 < 29 < 31 < 33 < 35 < 40 < 45 < 46 < 52 < ... < 100
```

```
Notice <- ("I noticed that there are more records for National than labour which is 6:5 respectively ")
```

#Chapter 5 Exercise 5.1

```
#Question a
num<- seq(from=-4,to=4,length= 20)
num
```

```
##  [1] -4.0000000 -3.5789474 -3.1578947 -2.7368421 -2.3157895 -1.8947368
##  [7] -1.4736842 -1.0526316 -0.6315789 -0.2105263  0.2105263  0.6315789
## [13]  1.0526316  1.4736842  1.8947368  2.3157895  2.7368421  3.1578947
## [19]  3.5789474  4.0000000
```

```
LogVec<-matrix(c(F,T,T,T,F,T,T,F,F),nrow=3, ncol=3)
LogVec
```

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

```
CharVec<-c("don", "quixote")
CharVec
```

```
## [1] "don"     "quixote"
```

```
FacVec<- factor(x=c("LOW","MED","LOW","MED","MED","HIGH"))
FacVec
```

```
## [1] LOW  MED  LOW  MED  MED  HIGH
## Levels: HIGH LOW MED
```

```r
L <- list(num,LogVec,CharVec,FacVec)
L
```

```
## [[1]]
##  [1] -4.0000000 -3.5789474 -3.1578947 -2.7368421 -2.3157895 -1.8947368
##  [7] -1.4736842 -1.0526316 -0.6315789 -0.2105263  0.2105263  0.6315789
## [13]  1.0526316  1.4736842  1.8947368  2.3157895  2.7368421  3.1578947
## [19]  3.5789474  4.0000000
##
## [[2]]
##       [,1]  [,2]  [,3]
## [1,] FALSE  TRUE  TRUE
## [2,]  TRUE FALSE FALSE
## [3,]  TRUE  TRUE FALSE
##
## [[3]]
## [1] "don"     "quixote"
##
## [[4]]
## [1] LOW  MED  LOW  MED  MED  HIGH
## Levels: HIGH LOW MED
```

```r
# i Extract elements nrow(2:1), ncol(2:3)
L[[2]][2:1,2:3]
```

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

```r
#ii Sub to overwrite quixote and don

L[[3]][1]<-sub(pattern="don", replacement="Don",x=L[[3]][1])
L[[3]][2]<-sub(pattern="quixote", replacement="Quixote",x=L[[3]][2])
L[[3]]
```

```
## [1] "Don"     "Quixote"
```

```r
cat("\"Windmills! ATTACK!\"\n"," ","", "-\\" ,L[[3]],"/-")
```

```
## "Windmills! ATTACK!"
##     -\ Don Quixote /-
```

```r
#iii values greater than 1
Gval <- L[[1]][L[[1]]>1]
Gval
```

```
## [1] 1.052632 1.473684 1.894737 2.315789 2.736842 3.157895 3.578947 4.000000
```

```r
#iv using which factor vector assigned to MED
which(x=L[[4]]=="MED")
```

```
## [1] 2 4 5
```

```
#Question b

newlist <- list(facs=L[[4]],nums=c(3,2.1,3.3,4,1.5,4.9),oldlist=L[1:3])
newlist
```

```
## $facs
## [1] LOW  MED  LOW  MED  MED  HIGH
## Levels: HIGH LOW MED
##
## $nums
## [1] 3.0 2.1 3.3 4.0 1.5 4.9
##
## $oldlist
## $oldlist[[1]]
##  [1] -4.0000000 -3.5789474 -3.1578947 -2.7368421 -2.3157895 -1.8947368
##  [7] -1.4736842 -1.0526316 -0.6315789 -0.2105263  0.2105263  0.6315789
## [13]  1.0526316  1.4736842  1.8947368  2.3157895  2.7368421  3.1578947
## [19]  3.5789474  4.0000000
##
## $oldlist[[2]]
##       [,1]  [,2]  [,3]
## [1,] FALSE  TRUE  TRUE
## [2,]  TRUE FALSE FALSE
## [3,]  TRUE  TRUE FALSE
##
## $oldlist[[3]]
## [1] "Don"     "Quixote"
```

```
#i Extract elements of facs greater = 3
Egreater <- newlist$facs[][newlist$nums>= 3]
Egreater
```

```
## [1] LOW  LOW  MED  HIGH
## Levels: HIGH LOW MED
```

```
#ii add new member
newlist$flags <- rep(x=newlist$oldlist[[2]][,3], times=2)
newlist$flags
```

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

```
#iii extract the entries of nums =false
entries<- newlist$nums[][!newlist$flags]
entries
```

```
## [1] 2.1 3.3 1.5 4.9
```

```
#iv Overwrite oldlist
newlist$oldlist[[3]]<- "Don Quixote"
```

Exercise 5.2

```
#Question a
dframe <- data.frame(person=c("Stan","Francine","Steve","Roger","Hayley","Klaus"),
         sex=factor(x=c("M","F","M","M","F","M")),
         funny=factor(x=c("High","Med","Low","High","Med","Med"),
                                    levels=c("Low","Med","High")),stringsAsFactors=FALSE)
dframe
```

```
##     person sex funny
## 1     Stan   M  High
## 2 Francine   F   Med
## 3    Steve   M   Low
## 4    Roger   M  High
## 5   Hayley   F   Med
## 6    Klaus   M   Med
```

```
# Question b
dframe$age <- c(41,41,15,1600,21,60)
dframe
```

```
##     person sex funny  age
## 1     Stan   M  High   41
## 2 Francine   F   Med   41
## 3    Steve   M   Low   15
## 4    Roger   M  High 1600
## 5   Hayley   F   Med   21
## 6    Klaus   M   Med   60
```

```
# Question c
dframe <- dframe[,c(1,4,2,3)]
dframe
```

```
##     person  age sex funny
## 1     Stan   41   M  High
## 2 Francine   41   F   Med
## 3    Steve   15   M   Low
## 4    Roger 1600   M  High
## 5   Hayley   21   F   Med
## 6    Klaus   60   M   Med
```

```
# Question d
mydata <- data.frame(person=c("Peter","Lois","Meg","Chris","Stewie","Brian"),
age=c(42,40,17,14,1,7),sex=factor(x=c("M","F","F","M","M", "M")),funny=factor(x=c("High","High","Low","
mydata
```

```
##   person age sex funny age.mon
## 1  Peter  42   M  High     504
## 2   Lois  40   F  High     480
## 3    Meg  17   F   Low     204
## 4  Chris  14   M   Med     168
## 5 Stewie   1   M  High      12
## 6  Brian   7   M   Med      84
```

```
mydata2 <- mydata[,-5]
mydata2
```

```
##   person age sex funny
## 1  Peter  42   M  High
## 2   Lois  40   F  High
## 3    Meg  17   F   Low
## 4  Chris  14   M   Med
## 5 Stewie   1   M  High
## 6  Brian   7   M   Med
```

```
# Question e
mydataframe <- rbind(mydata2,dframe)
mydataframe
```

```
##      person  age sex funny
## 1     Peter   42   M  High
## 2      Lois   40   F  High
## 3       Meg   17   F   Low
## 4     Chris   14   M   Med
## 5    Stewie    1   M  High
## 6     Brian    7   M   Med
## 7      Stan   41   M  High
## 8  Francine   41   F   Med
## 9     Steve   15   M   Low
## 10    Roger 1600   M  High
## 11   Hayley   21   F   Med
## 12    Klaus   60   M   Med
```

```
# Question f
mydataframe[mydataframe$sex=="F"&(mydataframe$funny=="Med"|mydataframe$funny=="High"),c("person","age")]
```

```
##      person age
## 2      Lois  40
## 8  Francine  41
## 11   Hayley  21
```

```
# Question g
mydataframe[substr(x=mydataframe$person,start=1,stop=1)=="S",]
```

```
##   person age sex funny
## 5 Stewie   1   M  High
## 7   Stan  41   M  High
## 9  Steve  15   M   Low
```

#Chapter 6 Exercise 6

```
#Question 6.1
# Question a
foo <- c(13563,-14156,-14319,16981,12921,11979,9568,8833,-12968,8133)
#(i)output not finite
foo[is.finite(foo^75)]
```

```
## [1] 11979  9568  8833  8133
```

```
#(ii)ex lude negative infinity
foo[-which(foo^75==-Inf)]
```

```
## [1] 13563 16981 12921 11979  9568  8833  8133
```

```
# Question b
bar <- matrix(c(77875.4,-35466.25,-39803.81,27551.45,-73333.85,55976.34,23764.3,36599.69,76694.82,-3647
#(i)is NaN
which(is.nan(bar^65/Inf),arr.ind=T)
```

```
##      row col
## [1,]   1   1
## [2,]   2   2
## [3,]   3   2
## [4,]   3   3
## [5,]   2   4
```

```
#(ii)Not NaN
bar[!is.nan(bar^67+Inf)]
```

```
##  [1]  77875.40 -35466.25 -39803.81  27551.45  55976.34  23764.30  36599.69
##  [8]  76694.82 -36478.88  47032.00
```

```
bar[bar^67!=-Inf]
```

```
##  [1]  77875.40 -35466.25 -39803.81  27551.45  55976.34  23764.30  36599.69
##  [8]  76694.82 -36478.88  47032.00
```

```
#(iii)
bar[bar^67==-Inf|is.finite(bar^67)]
```

```
## [1] -35466.25 -39803.81  27551.45 -73333.85  23764.30  36599.69 -36478.88
## [8] -70585.69
```

#Exercise 6.2

```
# Question a
foo <- c(4.3,2.2,NULL,2.4,NaN,3.3,3.1,NULL,3.4,NA)
#(i) the length of foo
length(x=foo)
```

```
## [1] 8
```

```
#(ii) not result in 4 and 8
which(x=is.na(x=foo))
```

```
## [1] 4 8
```

```r
#(iii)locations of two Null values
is.null(foo)
```

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

```r
#(iv)
is.na(x=foo[8])+4/NULL
```

```
## numeric(0)
```

```r
#Question b
bar <- list(c(7,7,NA,3,NA,1,1,5,NA))
#(i)
names(bar) <- "alpha"
#(ii)
is.null(x=bar$beta)
```

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

```r
#(iii)
bar$beta <- which(x=is.na(x=bar$alpha))
bar
```

```
## $alpha
## [1]  7  7 NA  3 NA  1  1  5 NA
##
## $beta
## [1] 3 5 9
```

#Exercise 6.3

```r
# Question a
#(i)
foo <- array(data=1:36,dim=c(3,3,4))
foo
```

```
## , , 1
##
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
##
## , , 2
##
##      [,1] [,2] [,3]
## [1,]   10   13   16
## [2,]   11   14   17
## [3,]   12   15   18
##
## , , 3
```

```
##
##      [,1] [,2] [,3]
## [1,]   19   22   25
## [2,]   20   23   26
## [3,]   21   24   27
##
## , , 4
##
##      [,1] [,2] [,3]
## [1,]   28   31   34
## [2,]   29   32   35
## [3,]   30   33   36
```

```
class(foo)
```

```
## [1] "array"
```

```
attributes(foo)
```

```
## $dim
## [1] 3 3 4
```

```
comment <- ("It is implicit")
#(ii)
bar <- as.vector(foo)
bar
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## [24] 24 25 26 27 28 29 30 31 32 33 34 35 36
```

```
class(bar)
```

```
## [1] "integer"
```

```
attributes(bar)
```

```
## NULL
```

```
comment2 <- ("It is implicit")
#(iii)
baz <- as.character(bar)
baz
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14"
## [15] "15" "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28"
## [29] "29" "30" "31" "32" "33" "34" "35" "36"
```

```
class(baz)
```

```
## [1] "character"
```

```
attributes(baz)
```

```
## NULL
```

```
comment <- ("It is implicit")
#(iv)
qux <- as.factor(baz)
qux
```

```
##  [1] 1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23
## [24] 24 25 26 27 28 29 30 31 32 33 34 35 36
## 36 Levels: 1 10 11 12 13 14 15 16 17 18 19 2 20 21 22 23 24 25 26 ... 9
```

```
class(qux)
```

```
## [1] "factor"
```

```
attributes(qux)
```

```
## $levels
##  [1] "1"  "10" "11" "12" "13" "14" "15" "16" "17" "18" "19" "2"  "20" "21"
## [15] "22" "23" "24" "25" "26" "27" "28" "29" "3"  "30" "31" "32" "33" "34"
## [29] "35" "36" "4"  "5"  "6"  "7"  "8"  "9"
##
## $class
## [1] "factor"
```

```
comment3 <- ("It is explicit")
#(v)
quux <- bar+c(-0.1,0.1)
quux
```

```
##  [1]  0.9  2.1  2.9  4.1  4.9  6.1  6.9  8.1  8.9 10.1 10.9 12.1 12.9 14.1
## [15] 14.9 16.1 16.9 18.1 18.9 20.1 20.9 22.1 22.9 24.1 24.9 26.1 26.9 28.1
## [29] 28.9 30.1 30.9 32.1 32.9 34.1 34.9 36.1
```

```
class(quux)
```

```
## [1] "numeric"
```

```
attributes(quux)
```

```
## NULL
```

```
comment <- ("It is implicit")
#(b)
foo.sum <- is.numeric(foo)+is.integer(foo)
bar.sum <- is.numeric(bar)+is.integer(bar)
baz.sum <- is.numeric(baz)+is.integer(baz)
qux.sum <- is.numeric(qux)+is.integer(qux)
quux.sum <- is.numeric(quux)+is.integer(quux)
Fac <- factor(x=c(foo.sum,bar.sum,baz.sum,qux.sum,quux.sum),levels=c(0,1,2))
Fac
```

```
## [1] 2 2 0 0 1
## Levels: 0 1 2
```

```
as.numeric(Fac)
```

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

```
#(c)
matx <- matrix(data=2:13,nrow=3,ncol=4)
matx
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    5    8   11
## [2,]    3    6    9   12
## [3,]    4    7   10   13
```

```
as.character(as.vector(t(matx)))
```

```
##  [1] "2"  "5"  "8"  "11" "3"  "6"  "9"  "12" "4"  "7"  "10" "13"
```

```
#(d)
matd <- cbind(c(34,23,33,42,41),c(0,1,1,0,0),c(1,2,1,1,2))
matd
```

```
##      [,1] [,2] [,3]
## [1,]   34    0    1
## [2,]   23    1    2
## [3,]   33    1    1
## [4,]   42    0    1
## [5,]   41    0    2
```

```
#(i)
matd <- as.data.frame(matd)
matd
```

```
##   V1 V2 V3
## 1 34  0  1
## 2 23  1  2
## 3 33  1  1
## 4 42  0  1
## 5 41  0  2
```

```r
#(ii)
matd[,2] <- as.logical(matd[,2])
matd
```

```
##   V1    V2 V3
## 1 34 FALSE  1
## 2 23  TRUE  2
## 3 33  TRUE  1
## 4 42 FALSE  1
## 5 41 FALSE  2
```

```r
#(iii)
matd[,3] <- as.factor(matd[,3])
matd
```

```
##   V1    V2 V3
## 1 34 FALSE  1
## 2 23  TRUE  2
## 3 33  TRUE  1
## 4 42 FALSE  1
## 5 41 FALSE  2
```

```r
matd$V3
```

```
## [1] 1 2 1 1 2
## Levels: 1 2
```