Exercises, Chapter 8

- TODO: Finish Exercises

Exercise 8.1

```
reverse = function (x) x[length(x):1]

reverse(1:15)
```

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

```
x<-sample(100,10,replace=FALSE)
x
```

```
##  [1] 73 99  7 38 60 98 85 83 36 19
```

```
reverse(x)
```

```
##  [1] 19 36 83 85 98 60 38  7 99 73
```

Exercise 8.2

```
my.cos = function(angle=NaN, degrees=FALSE) {
    ifelse(is.na(angle),
            NaN,
            ifelse(degrees,
                    cos(angle),
                    cos(180/pi*angle)
            )
    )
}

my.cos(90,TRUE)
```

```
## [1] -0.4480736
```

```
my.cos(90,degrees<-TRUE)
```

```
## [1] -0.4480736
```

```
my.cos(pi/2)
```

```
## [1] -0.4480736
```

```
my.cos()
```

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

Exercise 8.3

```
-8^(1/3)
```

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

```
# but
cube.root <- function(x) { (x)^(1/3) }
cube.root(c(-8,8,729,1000000))
```

```
## [1] NaN   2   9 100
```

```
# and
cube.root <- function(x) { as.numeric(x)^(1/3) }
cube.root(c(-8,8,729,1000000))
```

```
## [1] NaN   2   9 100
```

```
# so, ...
cube.root <- function(x) { y<-abs(x)^(1/3) ; ifelse(x>=0,y,-y) ; }
cube.root(c(-8,8,729,1000000))
```

```
## [1]  -2   2   9 100
```

Exercise 8.4

```
tmean = function (x,k) {
    sorted_x<-sort(x)
    n_from<-k+1
    n_to<-length(x)-k
    mean(sorted_x[n_from:n_to])
}

# case 1
tmean(c(9.4,9.6,9.1,9.5,9.3),1)
```

```
## [1] 9.4
```

```
# check against :
mean(c(9.4,9.5,9.3))
```

```
## [1] 9.4
```

```
# case 2
tmean(1:18,4)
```

```
## [1] 9.5
```

```
# check against :
mean(5:14)
```

```
## [1] 9.5
```

```
# setup for alternative forms
x<-sample(100,30,replace=FALSE)
x
```

```
##  [1]  71  97  49  28  89  22  94  37  74   1  63  65  70  24  62  99  31
## [18]  21  38  87  47 100  26  78  23   5  17  20   4  27
```

```
tmean(x,3)
```

```
## [1] 48.45833
```

```
tmean(x,7)
```

```
## [1] 45.9375
```

```
# alternative form #1
# GOTCHA: parenthesis are REQUIRED on the indexes
tmean = function (x,k) {
    y<-sort(x)
    mean(y[(k+1):(length(y)-k)])
}

tmean(c(9.4,9.6,9.1,9.5,9.3),1)
```

```
## [1] 9.4
```

```
tmean(x,3)
```

```
## [1] 48.45833
```

```
tmean(x,7)
```

```
## [1] 45.9375
```

```
# alternative form #2
# sort, then subset, then take the mean
tmean = function (x,k) { mean(sort(x)[(k+1):(length(x)-k)]); }
tmean(c(9.4,9.6,9.1,9.5,9.3),1)
```

```
## [1] 9.4
```

```
tmean(x,3)
```

```
## [1] 48.45833
```

```
tmean(x,7)
```

```
## [1] 45.9375
```

Exercise 8.5

```
moveave = function (x) { (x[1:length(x)-1] + x[2:length(x)])/2 }
moveave(c(1,2,3))
```

```
## [1] 1.5 2.5
```

```
moveave(c(10,20,30,40,50))
```

```
## [1] 15 25 35 45
```

Exercise 8.6

```
L2 = function (x) { sqrt(sum(x^2)) }
L2(c(3,4))
```

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

```
L2(c(1,1,1))
```

```
## [1] 1.732051
```

```
sqrt(3)
```

```
## [1] 1.732051
```

Exercise 8.7

```
Lp = function (x, p) { (sum(abs(x)^p))^(1/p)}
Lp(c(3,4),2)
```

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

```
Lp(c(1,1,1),3)
```

```
## [1] 1.44225
```

```
3^(1/3)
```

```
## [1] 1.44225
```

```
Lp(c(1,1,1),2)
```

```
## [1] 1.732051
```

Exercise 8.8

```
hmean_orig = function (x) { 1/((1/length(x))*(sum(1/x))) }
hmean_simp = function (x) { length(x) / (sum(1/x)) }
gmean      = function (x) { prod(x)^(1/(length(x))) }
qmean_orig = function (x) { sqrt( (1/length(x)) * (sum(x^2)) ) }
qmean_simp = function (x) { sqrt( (sum(x^2))/length(x) ) }

rv <- sample(100,30,replace = FALSE)
rvmin <-min(rv)
h<-hmean_orig(rv)
g<-gmean(rv)
m<-mean(rv)
q<-qmean_orig(rv)
rvmax<-max(rv)

(rvmin <= h) && (h <= g) && (g<=m) && (m<=q) && (q <= rvmax)
```

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

Alternate Ending

```
hmean_alt = function (x,n=length(x)) { n / (sum(1/x)) }
gmean_alt = function (x,n=length(x)) { prod(x)^(1/(n)) }
qmean_alt = function (x,n=length(x)) { sqrt( (sum(x^2))/n ) }

h_alt<-hmean_alt(rv)
g_alt<-gmean_alt(rv)
```

```
q_alt<-qmean_alt(rv)

(rvmin <= h_alt) && (h_alt <= g_alt) && (g_alt<=m) && (m<=q_alt) && (q_alt <= rvmax)
```

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

Summary

```
# using signif() because digits=4 parameter was ignored
library(knitr)
kable(matrix(
    c(
        c('rvmin',rvmin,rvmin),
        c('h',signif(h),signif(h_alt)),
        c('g',signif(g),signif(g_alt)),
        c('m',signif(m),signif(m)),
        c('q',signif(q),signif(q_alt)),
        c('rvmax',rvmax,rvmax)
    ),
    c(6,3),byrow=TRUE),
    col.names=c('variable','Original','Alternate'),
    align=c('r','l','l'),
    caption="A Comparison of Results")
```

Table 1: A Comparison of Results

variable	Original	Alternate
rvmin	1	1
h	14.8622	14.8622
g	38.516	38.516
m	52.7667	52.7667
q	60.4508	60.4508
rvmax	100	100

Exercise 8.9

```
mad = function (x,x_bar = mean(x),n = length(x) ) {
    (sum(abs(x-x_bar)))/n
    }
mad(c(1,2,3))
```

```
## [1] 0.6666667
```

```
mad(c(10,20,30,40,50))
```

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