Exercises Data Science with R

Thomas Kirschstein

Chapter 1

There are quite a lot standard basic mathematical functions. Try to google for R cheat sheets if you are interested in a comprehensive overview.

```
3^2     #1
## [1] 9
sqrt(9) #2
## [1] 3
9^.5
## [1] 3
pi^2    #3
## [1] 9.869604
(abs(3^2-4^2))^.5 #4
## [1] 2.645751
log(exp(4), base = exp(1)) #5
## [1] 4
log(exp(4))
## [1] 4
log(100, base = 10) #6
## [1] 2
log10(100)
## [1] 2
factorial(8) #7
## [1] 40320
exp(factorial(3)) #8
## [1] 403.4288
```

Chapter 2

1. Formulate the EOQ formula in R

```
cost_eoq_fun <- function(q, d, co, cl) {
  # returns total cost per period
  # d...demand
  # q...lot size
  # co...ordering cost
  # cl...stock holding cost
  ((1/2)*cl*q)+((d/q)*co)
}
```

```r
# test cost function
cost_eoq_fun(d=100, q=20, cl = .1, co = 50)
## [1] 251

eoq_fun <- function(co, d, cl) {
  # return optimal lot size
  # d...demand
  # co...ordering cost
  # cl...stock holding cost
  sqrt((2*co*d)/cl)
}
# optimal lot size
q.star <-  eoq_fun(d = 100, cl = .1, co = 50)
# optimal cost
cost_eoq_fun(d=100, q=q.star, cl = .1, co = 50)
## [1] 31.62278
```

2. Derive a function for calculating weighted Euclidean distance between two points.

```r
weuc_d2_func <- function(x,y,w) {
  # calculates weighted Euclidean distance between x and y
  # y,x...vectors
  # w.. weight vector
  sqrt(sum(w*(x-y)^2))
}
# test distacne function
weuc_d2_func(x = c(1,2,3), y= c(3,2,1), w=c(1,1,1) )
## [1] 2.828427
# result should be sqrt(8)
```

3. Formulate a function for the Geometric Poisson density distribution.

```r
geom_pois_dens_fun <- function(n, lambda, theta){
  # calculates density value of geometric Poisson distribution
  # n...integer, demand/successes, theta,lambda..parameters
  k.vec <- 1:n
  sum(exp(-lambda)*lambda^k.vec/factorial(k.vec)*(1-theta)^(n-k.vec)*choose(n-1, k.vec-1))
}
# test function
geom_pois_dens_fun(n=3, lambda=.5, theta = 2)
## [1] 0.1642687
```