

# Functions and control flow

*Jonas Schöley*

*September 15, 2017*

We already know how to calculate a life-table. Below, some familiar code.

```
swe <- read.table('swe_dxn.txt', skip = 3, header = TRUE)

periods <- unique(swe$period)
age_groups <- unique(swe$age_group)
D <- matrix(swe$deaths, nrow = length(age_groups), dimnames = list(age_groups, periods))
E <- matrix(swe$exposure, nrow = length(age_groups), dimnames = list(age_groups, periods))

x <- c(0, 1, seq(5, 110, 5))
nx <- c(diff(x), Inf)
nmx <- D[,1] / E[,1]
npx <- exp(-nx*nmx)
nqx <- 1-npx
lx <- cumprod(c(1, npx[-length(npx)]))
ndx <- c(diff(-lx), lx[length(lx)])
nLx <- -nx*ndx/log(npx); nLx[is.nan(nLx)] <- 0
Tx <- rev(cumsum(rev(nLx)))
ex <- Tx/lx
```

We can turn this code into a function.

```
LTFun <- function(x, Dx, Nx) {
  nx <- c(diff(x), Inf)
  nmx <- Dx / Nx
  npx <- exp(-nx*nmx)
  nqx <- 1-npx
  lx <- cumprod(c(1, npx[-length(npx)]))
  ndx <- c(diff(-lx), lx[length(lx)])
  nLx <- -nx*ndx/log(npx); nLx[is.nan(nLx)] <- 0
  Tx <- rev(cumsum(rev(nLx)))
  ex <- Tx/lx

  data.frame(x, nx, nmx, npx, nqx, lx, ndx, nLx, Tx, ex)
}

LTFun(x = x, Dx = D[,1], Nx = E[,1])
```

##	x	nx	nmx	npx	nqx	lx
## 0	0	1	0.246673006	0.7813961639	0.21860384	1.000000e+00
## 1-4	1	4	0.040349807	0.8509522792	0.14904772	7.813962e-01
## 5-9	5	5	0.012922361	0.9374312989	0.06256870	6.649308e-01
## 10-14	10	5	0.006535550	0.9678503999	0.03214960	6.233270e-01
## 15-19	15	5	0.006410377	0.9684563347	0.03154367	6.032873e-01
## 20-24	20	5	0.007992651	0.9608247432	0.03917526	5.842574e-01
## 25-29	25	5	0.009424996	0.9539681623	0.04603184	5.613689e-01
## 30-34	30	5	0.011527944	0.9439899881	0.05601001	5.355281e-01
## 35-39	35	5	0.011365367	0.9447576540	0.05524235	5.055332e-01
## 40-44	40	5	0.015960088	0.9233005802	0.07669942	4.776063e-01

```
## 45-49    45    5 0.016714464 0.9198245623 0.08017544 4.409742e-01
## 50-54    50    5 0.021696673 0.8971938193 0.10280618 4.056189e-01
## 55-59    55    5 0.027245664 0.8726433664 0.12735663 3.639188e-01
## 60-64    60    5 0.038280417 0.8258004786 0.17419952 3.175713e-01
## 65-69    65    5 0.050050188 0.7786053761 0.22139462 2.622505e-01
## 70-74    70    5 0.083510384 0.6586572937 0.34134271 2.041897e-01
## 75-79    75    5 0.120732198 0.5468061143 0.45319389 1.344910e-01
## 80-84    80    5 0.151401100 0.4690689536 0.53093105 7.354051e-02
## 85-89    85    5 0.200507910 0.3669463788 0.63305362 3.449557e-02
## 90-94    90    5 0.283582050 0.2422196669 0.75778033 1.265802e-02
## 95-99    95    5 0.387714960 0.1439089026 0.85609110 3.066023e-03
## 100-104 100    5 0.574341861 0.0566020935 0.94339791 4.412279e-04
## 105-109 105    5 1.413828689 0.0008509617 0.99914904 2.497443e-05
## 110+    110 Inf      NaN      NaN      NaN 2.125228e-08
##          ndx      nLx      Tx      ex
## 0          2.186038e-01 8.862090e-01 3.607786e+01 36.0778633
## 1-4          1.164653e-01 2.886391e+00 3.519165e+01 45.0368915
## 5-9          4.160386e-02 3.219525e+00 3.230526e+01 48.5843957
## 10-14        2.003971e-02 3.066263e+00 2.908574e+01 46.6620880
## 15-19        1.902989e-02 2.968608e+00 2.601948e+01 43.1294958
## 20-24        2.288843e-02 2.863685e+00 2.305087e+01 39.4532772
## 25-29        2.584084e-02 2.741735e+00 2.018718e+01 35.9606347
## 30-34        2.999494e-02 2.601933e+00 1.744545e+01 32.5761586
## 35-39        2.792684e-02 2.457188e+00 1.484352e+01 29.3621003
## 40-44        3.663213e-02 2.295233e+00 1.238633e+01 25.9341790
## 45-49        3.535530e-02 2.115252e+00 1.009109e+01 22.8836394
## 50-54        4.170013e-02 1.921960e+00 7.975843e+00 19.6633904
## 55-59        4.634747e-02 1.701095e+00 6.053883e+00 16.6352595
## 60-64        5.532077e-02 1.445145e+00 4.352788e+00 13.7064903
## 65-69        5.806086e-02 1.160053e+00 2.907643e+00 11.0872703
## 70-74        6.969866e-02 8.346107e-01 1.747590e+00 8.5586590
## 75-79        6.095051e-02 5.048405e-01 9.129792e-01 6.7884024
## 80-84        3.904494e-02 2.578907e-01 4.081386e-01 5.5498474
## 85-89        2.183755e-02 1.089111e-01 1.502479e-01 4.3555703
## 90-94        9.592002e-03 3.382443e-02 4.133674e-02 3.2656548
## 95-99        2.624795e-03 6.769908e-03 7.512306e-03 2.4501796
## 100-104      4.162535e-04 7.247487e-04 7.423981e-04 1.6825726
## 105-109      2.495317e-05 1.764936e-05 1.764936e-05 0.7066974
## 110+        2.125228e-08 0.000000e+00 0.000000e+00 0.0000000
```

A function can only return a single object. If we want to return multiple objects we store them in a single list.

```
LTFun <- function (x, Dx, Nx) {
  nx <- c(diff(x), Inf)
  nmx <- Dx / Nx
  npx <- exp(-nx*nmx)
  nqx <- 1-npx
  lx <- cumprod(c(1, npx[-length(npx)]))
  ndx <- c(diff(-lx), lx[length(lx)])
  nLx <- -nx*ndx/log(npx); nLx[is.nan(nLx)] <- 0
  Tx <- rev(cumsum(rev(nLx)))
  ex <- Tx/lx

  list(
    lt = data.frame(x, nx, nmx, npx, nqx, lx, ndx, nLx, Tx, ex),
```

```

summary = c(e0 = ex[[1]], m = x[which.max(ndx)], ltr = 1/ex[[1]])
)
}

LTFun(x = x, Dx = D[,1], Nx = E[,1])

```

```

## $lt
##      x  nx      nmX      npX      nqx      lx
## 0      0   1 0.246673006 0.7813961639 0.21860384 1.000000e+00
## 1-4     1   4 0.040349807 0.8509522792 0.14904772 7.813962e-01
## 5-9     5   5 0.012922361 0.9374312989 0.06256870 6.649308e-01
## 10-14   10  5 0.006535550 0.9678503999 0.03214960 6.233270e-01
## 15-19   15  5 0.006410377 0.9684563347 0.03154367 6.032873e-01
## 20-24   20  5 0.007992651 0.9608247432 0.03917526 5.842574e-01
## 25-29   25  5 0.009424996 0.9539681623 0.04603184 5.613689e-01
## 30-34   30  5 0.011527944 0.9439899881 0.05601001 5.355281e-01
## 35-39   35  5 0.011365367 0.9447576540 0.05524235 5.055332e-01
## 40-44   40  5 0.015960088 0.9233005802 0.07669942 4.776063e-01
## 45-49   45  5 0.016714464 0.9198245623 0.08017544 4.409742e-01
## 50-54   50  5 0.021696673 0.8971938193 0.10280618 4.056189e-01
## 55-59   55  5 0.027245664 0.8726433664 0.12735663 3.639188e-01
## 60-64   60  5 0.038280417 0.8258004786 0.17419952 3.175713e-01
## 65-69   65  5 0.050050188 0.7786053761 0.22139462 2.622505e-01
## 70-74   70  5 0.083510384 0.6586572937 0.34134271 2.041897e-01
## 75-79   75  5 0.120732198 0.5468061143 0.45319389 1.344910e-01
## 80-84   80  5 0.151401100 0.4690689536 0.53093105 7.354051e-02
## 85-89   85  5 0.200507910 0.3669463788 0.63305362 3.449557e-02
## 90-94   90  5 0.283582050 0.2422196669 0.75778033 1.265802e-02
## 95-99   95  5 0.387714960 0.1439089026 0.85609110 3.066023e-03
## 100-104 100  5 0.574341861 0.0566020935 0.94339791 4.412279e-04
## 105-109 105  5 1.413828689 0.0008509617 0.99914904 2.497443e-05
## 110+    110 Inf      NaN      NaN      NaN      NaN 2.125228e-08
##
##      ndx      nLx      Tx      ex
## 0      2.186038e-01 8.862090e-01 3.607786e+01 36.0778633
## 1-4     1.164653e-01 2.886391e+00 3.519165e+01 45.0368915
## 5-9     4.160386e-02 3.219525e+00 3.230526e+01 48.5843957
## 10-14    2.003971e-02 3.066263e+00 2.908574e+01 46.6620880
## 15-19    1.902989e-02 2.968608e+00 2.601948e+01 43.1294958
## 20-24    2.288443e-02 2.863685e+00 2.305087e+01 39.4532772
## 25-29    2.584084e-02 2.741735e+00 2.018718e+01 35.9606347
## 30-34    2.999494e-02 2.601933e+00 1.744545e+01 32.5761586
## 35-39    2.792684e-02 2.457188e+00 1.484352e+01 29.3621003
## 40-44    3.663213e-02 2.295233e+00 1.238633e+01 25.9341790
## 45-49    3.535530e-02 2.115252e+00 1.009109e+01 22.8836394
## 50-54    4.170013e-02 1.921960e+00 7.975843e+00 19.6633904
## 55-59    4.634747e-02 1.701095e+00 6.053883e+00 16.6352595
## 60-64    5.532077e-02 1.445145e+00 4.352788e+00 13.7064903
## 65-69    5.806086e-02 1.160053e+00 2.907643e+00 11.0872703
## 70-74    6.969866e-02 8.346107e-01 1.747590e+00 8.5586590
## 75-79    6.095051e-02 5.048405e-01 9.129792e-01 6.7884024
## 80-84    3.904494e-02 2.578907e-01 4.081386e-01 5.5498474
## 85-89    2.183755e-02 1.089111e-01 1.502479e-01 4.3555703
## 90-94    9.592002e-03 3.382443e-02 4.133674e-02 3.2656548
## 95-99    2.624795e-03 6.769908e-03 7.512306e-03 2.4501796

```

```
## 100-104 4.162535e-04 7.247487e-04 7.423981e-04 1.6825726
## 105-109 2.495317e-05 1.764936e-05 1.764936e-05 0.7066974
## 110+ 2.125228e-08 0.000000e+00 0.000000e+00 0.0000000
##
## $summary
##      e0      m      ltdr
## 36.07786328 0.00000000 0.02771783
```

Once you have written a function it is easy to *iterate* it over your data.

```
lts <- vector(mode = 'list', length = ncol(D))
for (i in 1:ncol(D)) {
  lts[[i]] <- LTFun(x = x, Dx = D[,i], Nx = E[,i])
}
sapply(lts, function (x) {x$summary['e0']})
```

```
##      e0      e0      e0      e0      e0      e0      e0      e0
## 36.07786 35.46542 33.09399 35.74394 38.45461 35.49436 37.19603 41.18388
##      e0      e0      e0      e0      e0      e0      e0      e0
## 40.98771 43.36909 41.63890 44.65841 46.48102 49.58567 52.01687 55.12746
##      e0      e0      e0      e0      e0      e0      e0      e0
## 56.82192 61.58237 64.45614 68.73963 72.22870 73.68003 74.94856 76.62122
##      e0      e0      e0
## 78.53245 80.42706 81.74005
```