Chapter 11 Answers

Highlights

- Review these functions
  - append
  - substr
  - table
- Don't forget these functions
  - array

- Also introduced this chapter
  - cat
  - factor
  - grep
  - levels
  - names
  - nchar
  - paste
  - print
  - sprintf
  - strsplit
  - sub
  - tolower
  - toupper
  - various data sets (including state.name)

11.1 Create a $2x3x2$ array named `abb` of U.S. state abbreviations. Then use the `substr` function to create a second $2x3x2$ array named `abb1` which contains just the first letters of each string in `abb`.

```
abb <- array(state.name,c(2,3,2))
abb
```

```
## , , 1
##
##      [,1]      [,2]       [,3]
## [1,] "Alabama" "Arizona"  "California"
## [2,] "Alaska"  "Arkansas" "Colorado"
##
## , , 2
##
##      [,1]          [,2]      [,3]
## [1,] "Connecticut" "Florida" "Hawaii"
## [2,] "Delaware"    "Georgia" "Idaho"
```

```
abb1 <- substr(abb,1,3)
abb1
```

```
## , , 1
##
##      [,1]  [,2]  [,3]
## [1,] "Ala" "Ari" "Cal"
## [2,] "Ala" "Ark" "Col"
##
## , , 2
```

```
##
##      [,1]  [,2]  [,3]
## [1,] "Con" "Flo" "Haw"
## [2,] "Del" "Geo" "Ida"
```

11.2 Discrete-event simulations involve a simulation clock, entities that pass through a system, attributes associated with the entities, and a calendar of future events. Consider a coffee shop with a single server. Currently, Arthur is being served at simulated times 57 minutes. Waiting in line behind Arthur, in order, are Bob, Charise, and Daniel. The event calendar contains, in order, the arrival of Emma at time 60, and the end of service of Arthur at time 62. The current state of the coffee shop is captured by the five R commands

```
# simluated time
> time          = 57

# Names of customers currently in shop as character strings
# ordered by their position in the queue
> coffee        = c( "Arthur",  "Bob",   "Charise",  "Daniel")

# arrival times of customers currently being served (in this case Arthur) and
# ...the customers currently waiting in line
> arrival       = c( 50,       52,      55,         56      )

# times of future events
> calendar      = c( 60,             62              )

# names of future events for the times in calendar
> event         = c( "arrival",      "end.of.service")
```

- Write the R commands that update the data structures to process the arrival of Emma at time 60, and schedule the arrival of Flip at time 64.
- Write the R commands that update the data structures to process the end of service of Arthur at time 62, and schedule the end of service of Bob at time 66.

```
# simluated time
time          = 57

# Names of customers currently in shop as character strings
# ordered by their position in the queue
coffee        = c( "Arthur",  "Bob",   "Charise",  "Daniel")

# arrival times of customers currently being served (in this case Arthur) and
# ...the customers currently waiting in line
arrival       = c( 50,       52,      55,         56      )

# times of future events
calendar      = c( 60,             62              )

# names of future events for the times in calendar
event         = c( "arrival",      "end.of.service")

# PART A
# Emma Arrives @ time 60
coffee      <- append(coffee,"Emma")
arrival     <- c(arrival,60)
# Flip Arrives @ time 64
coffee      <- c(coffee,"Flip")
```

```
arrival     <- append(arrival,64)

# PART B
# end of service for Arthur @ time 62
calendar    <- c(calendar,62)
event       <- c(event,"end.of.service")
# end of service for Bob @ time 66
calendar    <- c(calendar,66)
event       <- c(event,"end.of.service")
# only scheduling
# time has not lapsed (no increment to the time object)
# no need to remove Bob or Arthur from the coffee queue
# may find it odd that someone else has been scheudle for end.of.service @ 62
# but is not currently in the queue at time 57

coffee
```

```
## [1] "Arthur"  "Bob"     "Charise" "Daniel"  "Emma"    "Flip"
```

```
arrival
```

```
## [1] 50 52 55 56 60 64
```

```
calendar
```

```
## [1] 60 62 62 66
```

```
event
```

```
## [1] "arrival"        "end.of.service" "end.of.service" "end.of.service"
```

```
table(calendar)
```

```
## calendar
## 60 62 66
##  1  2  1
```

```
table(event)
```

```
## event
##        arrival end.of.service
##              1              3
```