Problem Sheet 1 - CT5102

Exploring Vectors

The goal of these short exercise is to practice key ideas from the lecture.

In most cases, the answers are provided as part of the output, the challenge is to write R code that will generate the results.

1. Predict what the types will be for the following variables, and then verify your results in R.

```
v1 <- c(1L, FALSE)
v2 <- c(1L, 2.0, FALSE)
v3 <- c(2.0, FALSE, "FALSE")
v4 <- c(1:20, seq(1,10,by=.5))
v5 <- unlist(list(1:10,list(11:20,"Hello")))
```

2. Create the following atomic vector, which is a combination of the character string Student and a sequence of numbers from 1 to 10. Explore how the R function **paste0()** can be used to generate the solution. Type **?paste0** to check out how this function can generate character strings.

```
# The output generated following the call to paste0()
slist
```

```
##  [1] "Student-1"  "Student-2"  "Student-3"  "Student-4"  "Student-5"
##  [6] "Student-6"  "Student-7"  "Student-8"  "Student-9"  "Student-10"
```

3. Generate a random sample of 20 temperatures (assume integer values in the range -5 to 30) using the **sample()** function (**set.seed(99)**). Assume that temperatures less than 4 are cold, temperatures greater that 25 are hot, and all others are medium, use the **ifelse()** function to generate the following vector. Note that an **ifelse()** call

can be nested within another **ifelse()** call.

```
# The temperature data set
temp
```

```
##  [1] 27 16 29 28 26  7 14 30 25 -2  3 12 18 24 16 14 26  8 -2  8
```

```
# The descriptions for each temperature generated by ifelse() call
des
```

```
##  [1] "Hot"    "Medium" "Hot"    "Hot"    "Hot"    "Medium" "Medium" "Hot"
##  [9] "Medium" "Cold"   "Cold"   "Medium" "Medium" "Medium" "Medium" "Medium"
## [17] "Hot"    "Medium" "Cold"   "Medium"
```

4. Use the expression **set.seed(100)** to ensure that you replicate the result as shown below. Configure a call to the function **sample()** that will generate a sample of 1000 for three categories of people: Young, Adult, and Elderly. Make use of the **prob** argument in **sample()** (which takes a vector probability weights for obtaining the elements of the vector being sampled) to ensure that 30% Young, 40% Adult and 30% Elderly are sampled. Use the **table()** function to generate the following output (assigned to the variable **ans**). Also, show the proportions for each category.

```
# A summary of the sample (1000 elements), based on the probability weights
ans
```

```
## pop
##   Adult Elderly   Young
##     399     300     301
```

```
# The proportions of each age
prop
```

```
## pop
##   Adult Elderly   Young
##   0.399   0.300   0.301
```