apply family

20 March, 2023

Write the R code to answer the following questions. You have until the beginning of next class to answer all of the questions below and commit to GitHub.

Question 1

Use an apply function instead of a for loop to create a vector containing the following elements: "num1", "num2", ... "num100"

```
out_vec <- sapply(1:100, function(x) paste0("num", x))
out_vec
```

```
##   [1] "num1"   "num2"   "num3"   "num4"   "num5"   "num6"   "num7"   "num8"
##   [9] "num9"   "num10"  "num11"  "num12"  "num13"  "num14"  "num15"  "num16"
##  [17] "num17"  "num18"  "num19"  "num20"  "num21"  "num22"  "num23"  "num24"
##  [25] "num25"  "num26"  "num27"  "num28"  "num29"  "num30"  "num31"  "num32"
##  [33] "num33"  "num34"  "num35"  "num36"  "num37"  "num38"  "num39"  "num40"
##  [41] "num41"  "num42"  "num43"  "num44"  "num45"  "num46"  "num47"  "num48"
##  [49] "num49"  "num50"  "num51"  "num52"  "num53"  "num54"  "num55"  "num56"
##  [57] "num57"  "num58"  "num59"  "num60"  "num61"  "num62"  "num63"  "num64"
##  [65] "num65"  "num66"  "num67"  "num68"  "num69"  "num70"  "num71"  "num72"
##  [73] "num73"  "num74"  "num75"  "num76"  "num77"  "num78"  "num79"  "num80"
##  [81] "num81"  "num82"  "num83"  "num84"  "num85"  "num86"  "num87"  "num88"
##  [89] "num89"  "num90"  "num91"  "num92"  "num93"  "num94"  "num95"  "num96"
##  [97] "num97"  "num98"  "num99"  "num100"
```

```
# equivalent
unlist(lapply(1:100, function(x) paste0("num", x)))
```

```
##   [1] "num1"   "num2"   "num3"   "num4"   "num5"   "num6"   "num7"   "num8"
##   [9] "num9"   "num10"  "num11"  "num12"  "num13"  "num14"  "num15"  "num16"
##  [17] "num17"  "num18"  "num19"  "num20"  "num21"  "num22"  "num23"  "num24"
##  [25] "num25"  "num26"  "num27"  "num28"  "num29"  "num30"  "num31"  "num32"
##  [33] "num33"  "num34"  "num35"  "num36"  "num37"  "num38"  "num39"  "num40"
##  [41] "num41"  "num42"  "num43"  "num44"  "num45"  "num46"  "num47"  "num48"
##  [49] "num49"  "num50"  "num51"  "num52"  "num53"  "num54"  "num55"  "num56"
##  [57] "num57"  "num58"  "num59"  "num60"  "num61"  "num62"  "num63"  "num64"
##  [65] "num65"  "num66"  "num67"  "num68"  "num69"  "num70"  "num71"  "num72"
##  [73] "num73"  "num74"  "num75"  "num76"  "num77"  "num78"  "num79"  "num80"
##  [81] "num81"  "num82"  "num83"  "num84"  "num85"  "num86"  "num87"  "num88"
##  [89] "num89"  "num90"  "num91"  "num92"  "num93"  "num94"  "num95"  "num96"
##  [97] "num97"  "num98"  "num99"  "num100"
```

```
sapply(1:100, paste0, "num")
```

```
##   [1] "1num"   "2num"   "3num"   "4num"   "5num"   "6num"   "7num"   "8num"
##   [9] "9num"   "10num"  "11num"  "12num"  "13num"  "14num"  "15num"  "16num"
##  [17] "17num"  "18num"  "19num"  "20num"  "21num"  "22num"  "23num"  "24num"
```

```
##  [25] "25num"  "26num"  "27num"  "28num"  "29num"  "30num"  "31num"  "32num"
##  [33] "33num"  "34num"  "35num"  "36num"  "37num"  "38num"  "39num"  "40num"
##  [41] "41num"  "42num"  "43num"  "44num"  "45num"  "46num"  "47num"  "48num"
##  [49] "49num"  "50num"  "51num"  "52num"  "53num"  "54num"  "55num"  "56num"
##  [57] "57num"  "58num"  "59num"  "60num"  "61num"  "62num"  "63num"  "64num"
##  [65] "65num"  "66num"  "67num"  "68num"  "69num"  "70num"  "71num"  "72num"
##  [73] "73num"  "74num"  "75num"  "76num"  "77num"  "78num"  "79num"  "80num"
##  [81] "81num"  "82num"  "83num"  "84num"  "85num"  "86num"  "87num"  "88num"
##  [89] "89num"  "90num"  "91num"  "92num"  "93num"  "94num"  "95num"  "96num"
##  [97] "97num"  "98num"  "99num"  "100num"
```

Question 2

Use a function in the "apply" family to find the fastest stopping speed for each group of cars, where the cars are grouped by their stopping distance

```
data(cars)
tapply(cars$speed, cars$dist, min)
```

```
##   2   4  10  14  16  17  18  20  22  24  26  28  32  34  36  40  42  46  48  50
##   4   7   4  12   8  11  10  12   7  12  10  11  16  10  14  16  18  13  20  17
##  52  54  56  60  64  66  68  70  76  80  84  85  92  93 120
##  20  15  18  14  20  22  19  24  18  14  18  25  24  24  24
```

Question 3

Using an apply function, find the means for each column in the `cars` dataset.

```
apply(cars, 2, mean)
```

```
## speed  dist
## 15.40 42.98
```