Problem set 2 sample solutions

2024-09-19

For these exercises, do not load any packages other than **dslabs**

Make sure to use vectorization whenever possible.

1. What is the sum of the first 100 positive integers? Use the functions `seq` and `sum` to compute the sum with R for any `n`.

```
# Sample answer
n <- 100
x <- seq(1, n)
sum(x)
```

```
[1] 5050
```

2. Load the US murders dataset in the **dslabs** package loaded. Use the function `str` to examine the structure of the `murders` object. What are the column names used by the data frame for these five variables? Show the subset of `murders` showing states with less than 1 per 100,000 deaths. Show all variables.

```
# Sample answer
library(dslabs)
str(murders)
```

```
'data.frame':   51 obs. of  5 variables:
 $ state     : chr  "Alabama" "Alaska" "Arizona" "Arkansas" ...
 $ abb       : chr  "AL" "AK" "AZ" "AR" ...
 $ region    : Factor w/ 4 levels "Northeast","South",..: 2 4 4 2 4 4 1 2 2 2 ...
 $ population: num  4779736 710231 6392017 2915918 37253956 ...
 $ total     : num  135 19 232 93 1257 ...
```

```
murders$rate <- with(murders, total/population*10^5)
murders[murders$rate < 1,]
```

```
           state abb        region population total      rate
12        Hawaii  HI          West    1360301     7 0.5145920
13         Idaho  ID          West    1567582    12 0.7655102
16          Iowa  IA North Central    3046355    21 0.6893484
20         Maine  ME     Northeast    1328361    11 0.8280881
24     Minnesota  MN North Central    5303925    53 0.9992600
30 New Hampshire  NH     Northeast    1316470     5 0.3798036
35  North Dakota  ND North Central     672591     4 0.5947151
38        Oregon  OR          West    3831074    36 0.9396843
42  South Dakota  SD North Central     814180     8 0.9825837
45          Utah  UT          West    2763885    22 0.7959810
46       Vermont  VT     Northeast     625741     2 0.3196211
51       Wyoming  WY          West     563626     5 0.8871131
```

3. Show the subset of `murders` showing states with less than 1 per 100,000 deaths and in the West of the US. Don't show the `region` variable.

```
# Sample answer
murders[murders$rate < 1 & murders$region == "West",]
```

```
     state abb region population total      rate
12  Hawaii  HI   West    1360301     7 0.5145920
13   Idaho  ID   West    1567582    12 0.7655102
38  Oregon  OR   West    3831074    36 0.9396843
45    Utah  UT   West    2763885    22 0.7959810
51 Wyoming  WY   West     563626     5 0.8871131
```

4. Show the largest state with a rate less than 1 per 100,000.

```
# Sample answer
dat <- murders[murders$rate < 1,]
dat[which.max(dat$population),]
```

```
       state abb        region population total    rate
24 Minnesota  MN North Central    5303925    53 0.99926
```

5. Show the state with a population of more than 10 million with the lowest rate.

```
# Sample answer
dat <- murders[murders$population >= 10^7,]
dat[which.min(dat$rate),]
```

```
      state abb    region population total    rate
33 New York  NY Northeast   19378102   517 2.66796
```

6. Compute the rate for each region of the US.

```
# Sample answer
indexes <- split(1:nrow(murders), murders$region)
sapply(indexes, function(ind) {
  sum(murders$total[ind])/sum(murders$population[ind])*10^5
})
```

```
    Northeast         South North Central          West
     2.655592      3.626558      2.731334      2.656175
```

7. Create a vector of numbers that starts at 6, does not pass 55, and adds numbers in increments of 4/7: 6, 6 + 4/7, 6 + 8/7, and so on. How many numbers does the list have? Hint: use `seq` and `length`.

```
# Sample answer
length(seq(6, 55, 4/7))
```

```
[1] 86
```

8. Make this data frame:

```
temp <- c(35, 88, 42, 84, 81, 30)
city <- c("Beijing", "Lagos", "Paris", "Rio de Janeiro",
          "San Juan", "Toronto")
city_temps <- data.frame(name = city, temperature = temp)
```

Convert the temperatures to Celsius.

```
# Sample answer
city_temps$temp <- (city_temps$temp - 32)*5/9
```

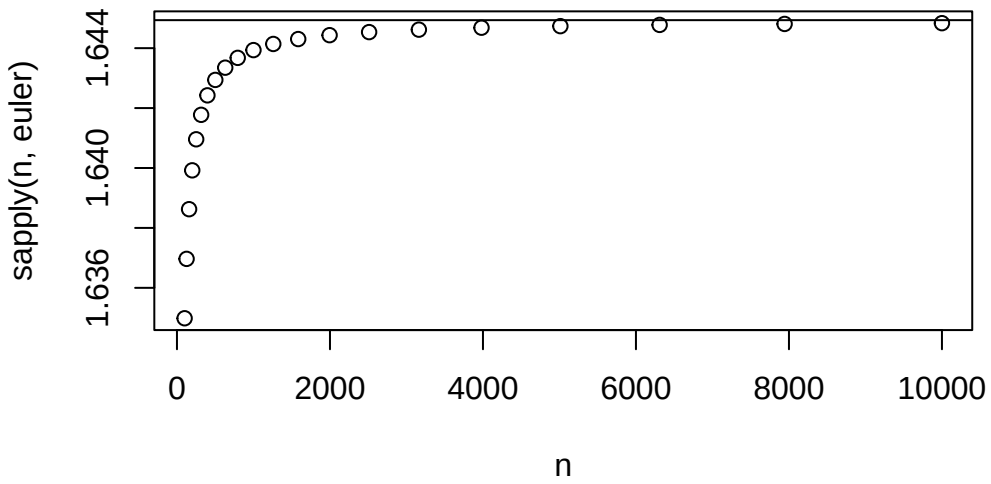
9. Write a function `euler` that compute the following sum for any n:

$$S_n = 1 + 1/2^2 + 1/3^2 + ... 1/n^2$$

```
# Sample answer
euler <- function(n){
  sum(1/seq(1,n)^2)
}
```

10. Show that as n gets bigger we get closer $\pi^2/6$ by plotting S_n versus n with a horizontal dashed line at $\pi^2/6$.

```
# Sample answer
n <- 10^seq(2, 4, 0.1)
plot(n, sapply(n, euler))
abline(h = pi^2/6)
```



11. Use the `%in%` operator and the predefined object `state.abb` to create a logical vector that answers the question: which of the following are actual abbreviations: MA, ME, MI, MO, MU?

```
# Sample answer
c("MA", "ME", "MI", "MO", "MU") %in% state.abb
```

```
[1]  TRUE  TRUE  TRUE  TRUE FALSE
```

12. Extend the code you used in the previous exercise to report the one entry that is **not** an actual abbreviation. Hint: use the `!` operator, which turns `FALSE` into `TRUE` and viceversa, then `which` to obtain an index.

```
# Sample answer
which(!c("MA", "ME", "MI", "MO", "MU") %in% state.abb)
```

```
[1] 5
```

13. In the `murders` dataset, use `%in%` to show all variables for New York, California, and Texas, in that order.

```
# Sample answer
library(dslabs)
result <- murders[murders$state %in% c("New York", "California", "Texas"),]
result[match(c("New York", "California", "Texas"), result$state),]
```

```
        state abb    region population total     rate
33   New York  NY Northeast   19378102   517 2.667960
5  California  CA      West   37253956  1257 3.374138
44      Texas  TX     South   25145561   805 3.201360
```

14. Write a function called `vandermonde_helper` that for any x and n, returns the vector $(1 x x^2 x^3 \dots x^n)$. Show the results for $x = 3$ and $n = 5$.

```
# Sample answer
vandermonde_helper <- function(x, n) x^(0:n)
vandermonde_helper(3, 5)
```

```
[1]   1   3   9  27  81 243
```

15. Create a vector using:

```
# Sample answer
n <- 10000
p <- 0.5
set.seed(2024-9-6)
x <- sample(c(0,1), n, prob = c(1 - p, p), replace = TRUE)
```

Compute the length of each stretch of 1s and then plot the distribution of these values. Check to see if the distribution follows a geometric distribution as the theory predicts. Do not use a loop!

```
# Sample answer
d <- diff(c(0,x,0))
start <- which(d == 1)
end <- which(d == -1)
y <- end - start
pr <- table(y)/length(y)
k <- as.numeric(names(pr))
pr <- as.numeric(pr)
plot(k, pr, type = "h")
lines(k, p^k)
```