

Week 3 Exercises

HD Sheets

10/1/2024

Please complete all exercises below. You may use any library that we have covered in class UP TO THIS POINT.

For each function, show that it works, by using the provided data as a test and by feeding in some test data that you create to test your function

Add comments to your function to explain what each line is doing

1.) Write a function that takes in a string with a person's name in the form

"Sheets, Dave"

and returns a string of the form

"Dave Sheets"

Note:

-assume no middle initial ever -remove the comma -be sure there is white space between the first and last name

You will probably want to use stringr

```
library(stringr)
name_in="Sheets, Dave"

reorder_name<-function(x){
  y = str_split(x, ",")
  z = sub("(\\w+)(\\W+)(\\w+)", "\\3\\2\\1", x)
  q = gsub(' ','',z)

  return(q)
}

reorder_name(name_in)
```

```
## [1] "Dave Sheets"
```

2.) Write a function that takes in a string of values x, and returns a data frame with three columns, x, x² and the square root of x

```
x=c(1,3,5,7,9,11,13)

powers_df<-function(x)
{
```

```

df <- as.data.frame(x)
df$x_square <- x^2
df$x_root <- x*0.5

return(df)
}

#your code here
powers_df(x)

```

```

##      x x_square x_root
## 1  1         1   0.5
## 2  3         9   1.5
## 3  5        25   2.5
## 4  7        49   3.5
## 5  9        81   4.5
## 6 11       121   5.5
## 7 13       169   6.5

```

3.) Write in a function that takes in a value x and returns

```

y= 0.3x if x<0
y=0.5x if x>=0

```

This is a variant on a relu function as used in some neural networks.

```

relu <-function(x)
{
  if(x<0)
  {
    return(y=0.5*x)
  }
  if(x>=0)
  {
    return(y=0.3*x)
  }
  return(y)
}

relu(2)

```

```
## [1] 0.6
```

4.) Write a function that takes in a value x and returns the first power of two greater than x (use a While loop)

```

power_of_two <- function(x) {
  result <- 1
  while(result <= x) {
    result <- result * 2
  }
}

```

```

    return(result)
}
power_of_two(3)

```

```
## [1] 4
```

5) Two Sum - Write a function named two_sum()

Given a vector of integers nums and an integer target, return indices of the two numbers such that they add up to target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

You can return the answer in any order.

Note: For the first problem I want you to use a brute force approach (loop inside a loop)

The brute force approach is simple. Loop through each element x and find if there is another value that equals to $\text{target} - x$

Use the function seq_along to iterate

```

two_sum <- function(nums_vector,target){
  n <- length(nums_vector)
  for(i in 1:(n-1)) {
    for(j in (i+1):n) {
      if(nums_vector[i] + nums_vector[j] == target) {
        return(c(i, j))
      }
    }
  }
  return(NULL)
}

# Test code
nums_vector <- c(5,7,12,34,6,10,8,9)
target <- 13

z=two_sum(nums_vector,target)
print(z)

```

```
## [1] 1 7
```

```

#expected answers
#[1] 1 7
#[1] 2 5
#[1] 5 2

```

6.) Write one piece of code that will use a regex command to extract a phone number written in the form

456-123-2329

The sentences to use are located below
use the `str_extract` function from `stringr`
use the same regex search pattern from each
-What does `\d` match to? or alternatively `[[:digit:]]`
-How do you specify a specific number of repeated characters

```
a="Please call me at 456-123-2329, asap"
b="Hey, we have a code 234 on machine a-234-12, call me at 678-321-98766"
c="On 12-23-2022, Joe over at 122 Turnpike, dialled 912-835-4756, tell me by 9:02 pm Wed"

extract_phone <- function(text) {
  phone_number <- str_extract(text, "\\d{3}-\\d{3}-\\d{4}")
  return(phone_number)
}

extract_phone(a)
```

```
## [1] "456-123-2329"
```

```
extract_phone(b)
```

```
## [1] "678-321-9876"
```

```
extract_phone(c)
```

```
## [1] "912-835-4756"
```

7.) For lines below, extract the domains (ie the part of the address after @)

```
d="jimmy.halibut@gmail.com"
e="His address is: c.brown@hopeles.org, do write him"
f="h.potter@hogwarts.edu is bouncing back on me, I wonder why?"

emails <- c(d, e, f)

# Extract domains using regular expression
domains <- sub(".*@", "", emails)
domains
```

```
## [1] "gmail.com"
```

```
## [2] "hopeles.org, do write him"
```

```
## [3] "hogwarts.edu is bouncing back on me, I wonder why?"
```