

VIT-AP UNIVERSITY, ANDHRA PRADESH

CSE4027 – Data Analytics - Lab Sheet: 8

Academic year: 2021-2022

Semester: Fall

Faculty Name: Dr. Syed Khasim

Branch/ Class: B.Tech

Date: 28-10-22

School: SCOPE

NAME: Majjiga Jaswanth

REGNO:20BCD7171

1. Write a R function to find the sum of prime numbers within the given limit. Use nested function to check whether a number is a prime or not.

Code:

```
> sumOfPrime <- function(limit){
+   if (limit>=2){
+     x = seq(2,limit)
+     primes <- c()
+     for( i in seq(2,limit)){
+       flag <- is.prime(i)
+       if(flag){
+         primes <- c(primes,i)
+       }
+     }
+     return (sum(primes))
+   }else{
+     stop("Input should be at least 2.")
+   }
+ }
```

```
> print(sumOfPrime(15))
```

output:

```
> is.prime <- function(num) {
+   if (num == 2) {
+     TRUE
+   } else if (any(num %% 2:(num-1) == 0)) {
+     FALSE
+   } else {
+     TRUE
+   }
+ }
> sumOfPrime <- function(limit){
+   if (limit>=2){
+     x = seq(2,limit)
+     primes <- c()
+     for( i in seq(2,limit)){
+       flag <- is.prime(i)
+       if(flag){
+         primes <- c(primes,i)
+       }
+     }
+     return (sum(primes))
+   }else{
+     stop("Input should be at least 2.")
+   }
+ }
> print(sumOfPrime(15))
[1] 41
> |
```

2. Create a function that given a vector and an integer will return the occurrences of the integer inside the vector.

Code:

```
> occurrences <- function(v,n){
+   count <- 0
+   for(i in v){
+     if(i == n){
+       count <- count + 1
+     }
+   }
+   return (count)
+ }
> vec <- c(25,16,52,84,75,36,15,24,37)
> n <- 24
> print(occurrences(vec,n))
```

Output: _

```
> occurrences <- function(v,n){
+   count <- 0
+   for(i in v){
+     if(i == n){
+       count <- count + 1
+     }
+   }
+   return (count)
+ }
> vec <- c(25,16,52,84,75,36,15,24,37)
> n <- 24
> print(occurrences(vec,n))
[1] 1
> |
```

3. Write a R function to return the factorial values of individual digits

Code:

```
> indiFact <- function(num){
+   vec <- as.numeric(strsplit(as.character(num), ""))
+   [[1]]
+   fact <- c()
+   for(i in 1:length(vec)){
+     f <- 1
+     for(j in 1:vec[i]){
+       f <- f * j
+     }
+     fact[i] <- f
+   }
+   print(fact)
+ }
> indiFact(352)
```

Output:

```

> indiFact <- function(num){
+   vec <- as.numeric(strsplit(as.character(num), "")
+                         [[1]])
+   fact <- c()
+   for(i in 1:length(vec)){
+     f <- 1
+     for(j in 1:vec[i]){
+       f <- f * j
+     }
+     fact[i] <- f
+   }
+   print(fact)
+ }
> indiFact(352)
[1] 6 120 2
>

```

4. Use COVID'19 Dataset to answer the following.

a. Draw positive cases line plot for last 10 days of AP, TN and KL states

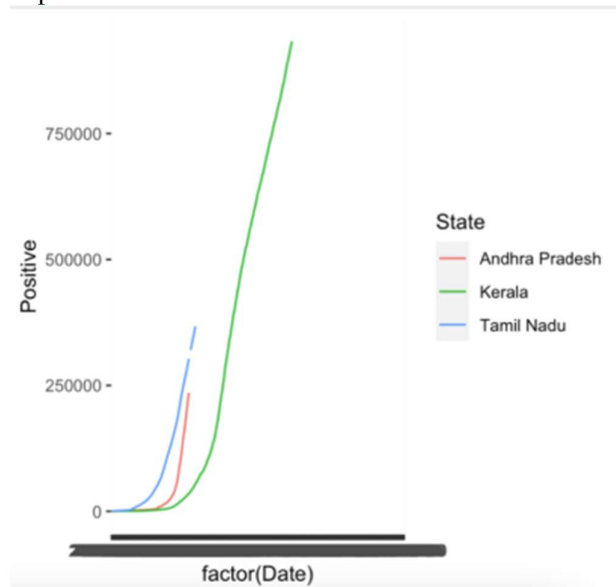
code:

```

library(ggplot2)
ne <- read.csv("APKLTN.csv")
ggplot(ne, aes(x = factor(Date), y = Positive, colour =
                State, group = State)) +
  geom_line()

```

output:



b. Draw negative cases line plot for last 10 days of AP, TN and KL states

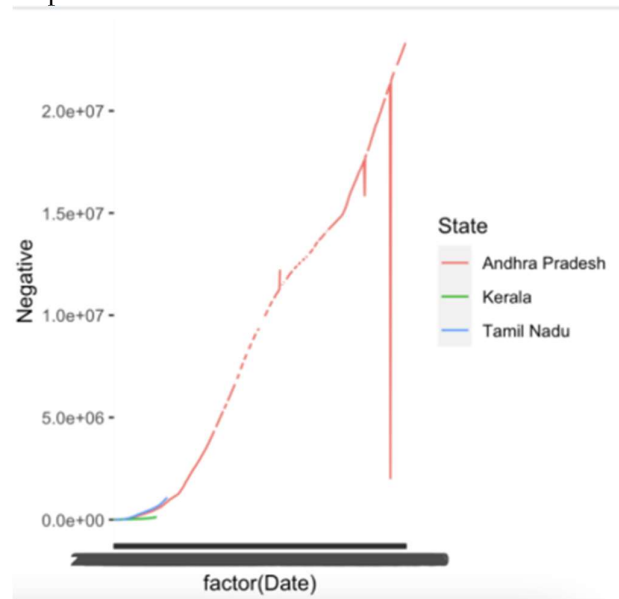
code:

```

library(ggplot2)
ne <- read.csv("APKLTN.csv")
ggplot(ne, aes(x = factor(Date), y = Negative, colour =
                State,
                group = State)) +
  geom_line()

```

output:



c. Draw positive cases line plot for last 10 days of top 3 highest positive cases states

code:

```
library(ggplot2)
df <- read.csv("c19.csv")
ne1 <- df[which(df$State == "Kerala"|df$State ==
               "Maharashtra"|df$State == "Karnataka"),]
ggplot(ne1, aes(x = factor(Date), y = Positive, colour
               = State, group = State)) +
  geom_line()
```

output:

