

VIT-AP UNIVERSITY, ANDHRA PRADESH

CSE2047 – Data Analytics - Lab Sheet : 8

Academic year: 2020-2021

Semester: Fall

Faculty Name: Prof. S.Gopikrishnan

Student name: Valiveti Manikanta bhuvanesh

Branch/ Class: B.Tech/M.Tech

Date:

School: SCOPE

Reg. no.: 19BCD7088

LAB 8

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

Example:

Input: 8

Output: 18 (1+2+3+5+7)

```
prime<-function(n){  
  s=1  
  for (j in 2:n) {  
    f = 1  
    i = 2  
    n = j  
    while (i <= n / 2) {  
      if (n %% i == 0) {  
        f = 0  
        break  
      }  
      i = i + 1  
    }  
  
    if (f == 1) {  
      s=s+n  
    }  
  }  
}
```

```

    }
    print(paste("The sum of primes :",s))
  }
prime(8)

> prime<-function(n){
+   s=1
+   for (j in 2:n) {
+     f = 1
+     i = 2
+     n = j
+     while (i <= n / 2) {
+       if (n %% i == 0) {
+         f = 0
+         break
+       }
+       i = i + 1
+     }
+     if (f == 1) {
+       s=s+n
+     }
+   }
+   print(paste("The sum of primes :",s))
+ }
> prime(8)
[1] "The sum of primes : 18"
> |

```

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

Example:

Input: inp=c(34,23,12,32,34,23,12,24,23), n=23

Output: 3

```

occur<-function(v,n){
  count=0
  for(i in v){
    if(i==n){
      count=count+1
    }
  }
  print(paste("Number of occurrences of ", n , " is ",count))
}

```

```

v<-sample(1:1000, 100, replace=TRUE)

```

```

v1<-c(3,5,6,6,8,7,6,3,2,45)
n=readline()
n=as.integer(n)
occur(v1,n)

> occur<-function(v,n){
+   count=0
+   for(i in v){
+     if(i==n){
+       count=count+1
+     }
+   }
+   print(paste("Number of occurrences of ", n , " is ",count))
+ }
> v<-sample(1:1000, 100, replace=TRUE)
> v1<-c(3,5,6,6,8,7,6,3,2,45)
> n=readline()
6
> n=as.integer(n)
> occur(v1,n)
[1] "Number of occurrences of 6 is 3"
> |

```

3. Write a R function to return the factorial values of individual digits in the given input:

Example:

Input: n=453

Output: 24 120 6

```

fact<-function(n){
  factorial = 1
  if(n == 0) {
    return(1)
  } else {
    for(i in 1:n) {
      factorial = factorial * i
    }
    return(factorial)
  }
}

extract<-function(n){
  while (n !=0) {
    rem=n%%10

```

```

    print(fact(rem))
    n=n%%10
  }
}
n=readline("Enter the number ")
n=as.integer(n)
extract(n)

> fact<-function(n){
+   factorial = 1
+   if(n == 0) {
+     return(1)
+   } else {
+     for(i in 1:n) {
+       factorial = factorial * i
+     }
+     return(factorial)
+   }
+ }
> extract<-function(n){
+   while (n !=0) {
+     rem=n%%10
+     print(fact(rem))
+     n=n%%10
+   }
+ }
> n=readline("Enter the number ")
Enter the number 453
> n=as.integer(n)
> extract(n)
[1] 6
[1] 120
[1] 24

```

4. Write a R recursive function to find the odd man out. Assume there are n candidates are attending an interview. In every next round even numbered position candidate alone will be shortlisted. Find the candidate id who is selected and how many rounds of interview will be conducted.

Example:

Input: candID= 23, 12, 34, 56, 43, 54, 65, 27, 45, 21

Output: 27 and 3

Explanation: 1st round result: 12, 56, 54, 27, 21

2nd round result: 56, 27 and 3rd round result: 27

```
win<-function(v,n = 0){  
  if(length(v)==1){  
    print(paste(v," and ",n-1))  
  }  
  else{  
    v=v[c(TRUE, FALSE)]  
    n=n+1  
    return(win(v,n))  
  }  
}
```

```
v<-c(23, 12, 34, 56, 43, 54, 65, 27, 45, 21)
```

```
win(v)
```

```
> win<-function(v,n = 0){  
+   if(length(v)==1){  
+     print(paste(v," and ",n-1))  
+   }  
+   else{  
+     v=v[c(TRUE, FALSE)]  
+     n=n+1  
+     return(win(v,n))  
+   }  
+ }  
> v<-c(23, 12, 34, 56, 43, 54, 65, 27, 45, 21)  
> win(v)  
[1] "23 and 3"  
> |
```

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

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

```
df<-read.csv('complete.csv')
```

```
dfap<-subset(df,state=="Andhra Pradesh")
```

```
dfap<-dfap[1:10,]
```

```
plot(dfap$TotalConfirmedcases,type = "o", col = "red", xlab = "Date", ylab = "Positive cases",main = "Andhran pradesh")
```

```
dftn<-subset(df,state=="Telangana")
```

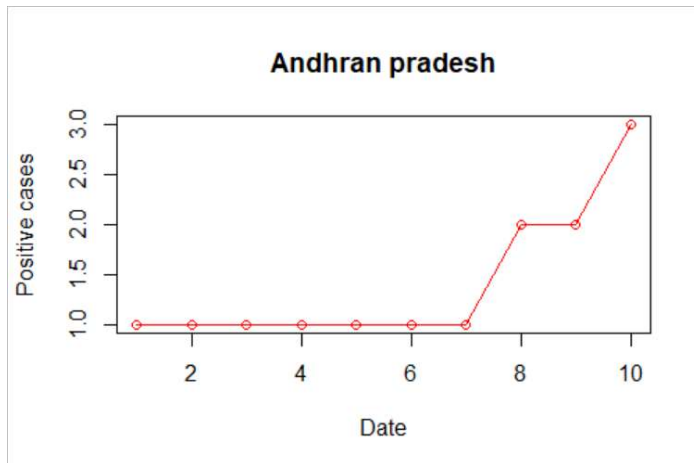
```
dftn<-dftn[1:10,]
```

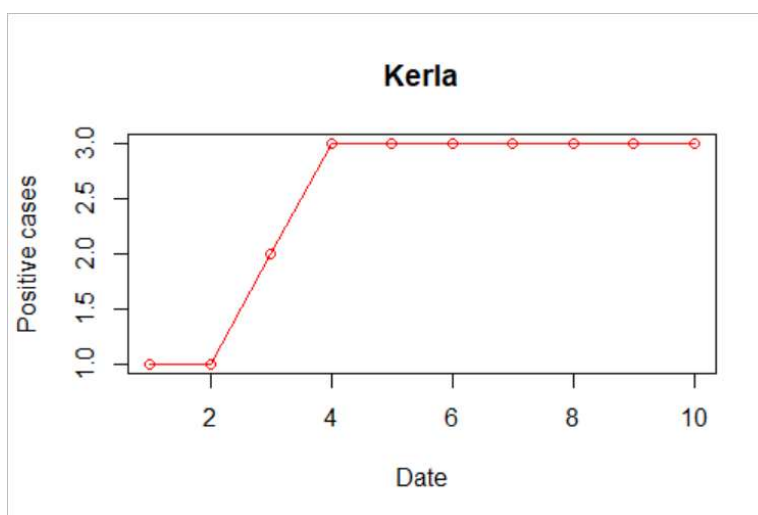
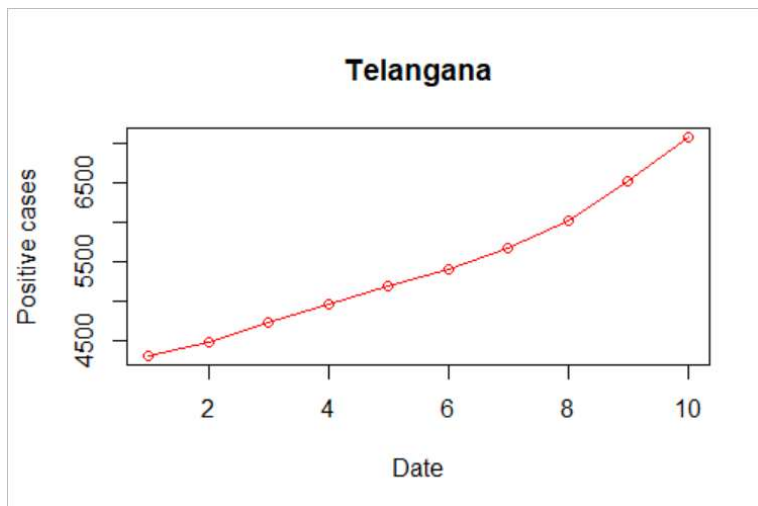
```
plot(dftn$TotalConfirmedcases,type = "o", col = "red", xlab = "Date", ylab = "Positive cases",main="Telangana")
```

```
dfk<-subset(df,state=="Kerala")
```

```
dfk<-dfk[1:10,]
```

```
plot(dfk$TotalConfirmedcases,type = "o", col = "red", xlab = "Date", ylab = "Positive cases",main = "Kerla")
```





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

```
dfap<-subset(df,state=="Andhra Pradesh")
```

```
dfap<-dfap[1:10,]
```

```
plot(dfap$Cured,type = "o", col = "red", xlab = "Date", ylab = "Negative cases",main = "Andhran pradesh")
```

```
dftn<-subset(df,state=="Telangana")
```

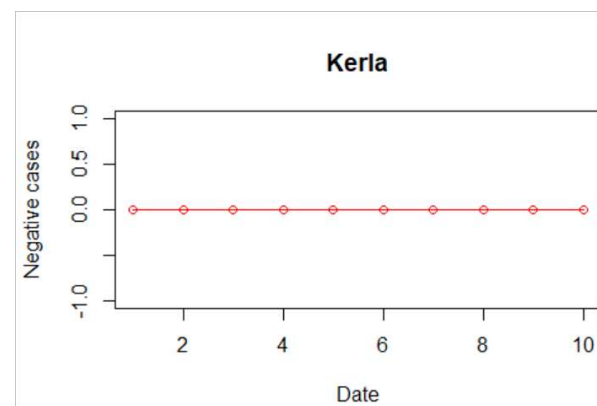
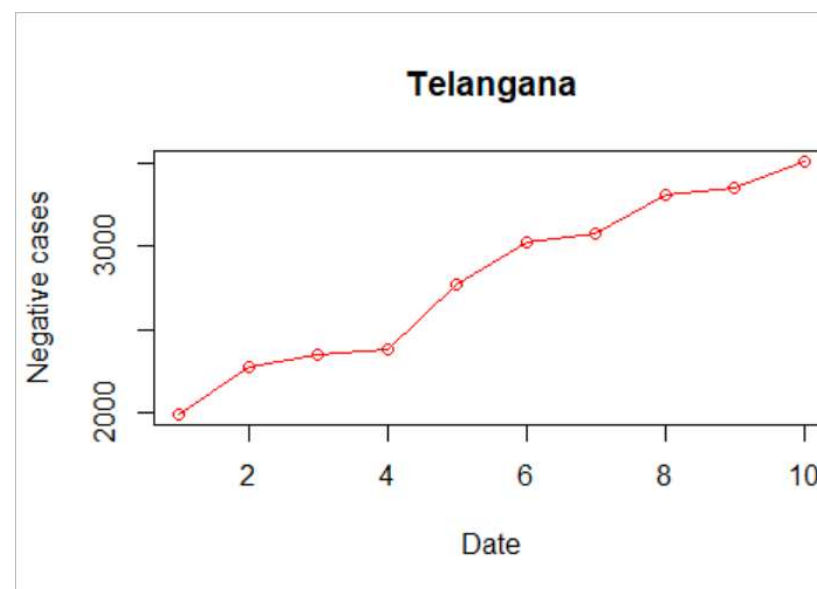
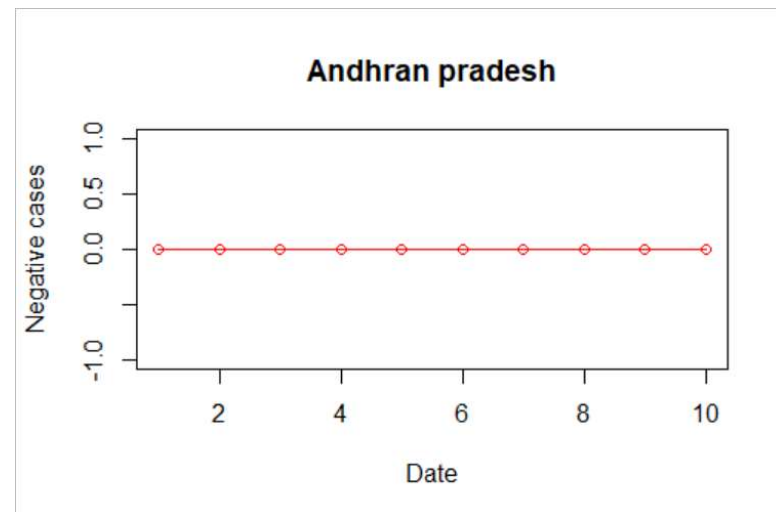
```
dftn<-dftn[1:10,]
```

```
plot(dftn$Cured,type = "o", col = "red", xlab = "Date", ylab = "Negative cases",main="Telangana")
```

```
dfk<-subset(df,state=="Kerala")
```

```
dfk<-dfk[1:10,]
```

```
plot(dfk$Cured,type = "o", col = "red", xlab = "Date", ylab = "Negative cases",main =  
"Kerla")
```

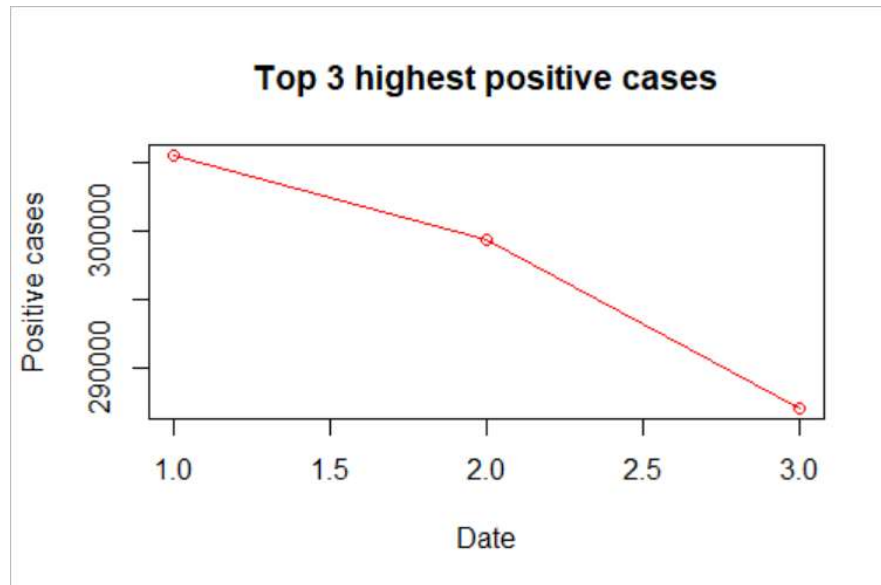


- c. Draw positive cases line plot for last 10 days of top 3 highest **positive cases states**

```
df1<-df[order(df$TotalConfirmedcases,decreasing = TRUE),]
```

```
df1<-df1[1:3,]
```

```
plot(df1$Cured,type = "o", col = "red", xlab = "Date", ylab = "Positive cases",main="Top 3 highest positive cases")
```



- d. Draw negative cases line plot for last 10 days of top 3 highest **recovery cases states**

```
df2<-df[order(df$Cured,decreasing = TRUE),]
```

```
df2<-df2[1:3,]
```

```
plot(df2$Cured,type = "o", col = "red", xlab = "Date", ylab = "Negative cases",main="Top 3 negative cases")
```

