A3_Tianyi_Zuo

Tianyi Zuo

2022/1/26

GitHub username: Lydia12138

Date: 2022-01-26 Repo:https://github.com/Lydia12138/Assignment3_TianyiZuo

```
library(dplyr)#package loading
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
MyData<-read.csv("./InputData/FallopiaData.csv")#loading the data
str(MyData)#check the structure of the Data
```

```
## 'data.frame':    123 obs. of  13 variables:
##  $ PotNum      : int  1 2 3 5 6 7 8 9 10 11 ...
##  $ Scenario    : chr  "low" "low" "low" "low" ...
##  $ Nutrients   : chr  "low" "low" "low" "low" ...
##  $ Taxon       : chr  "japon" "japon" "japon" "japon" ...
##  $ Symphytum   : num  9.81 8.64 2.65 1.44 9.15 ...
##  $ Silene      : num  36.4 29.6 36 21.4 23.9 ...
##  $ Urtica      : num  16.08 5.59 17.09 12.39 5.19 ...
##  $ Geranium    : num  4.68 5.75 5.13 5.37 0 9.05 3.51 9.64 7.3 6.36 ...
##  $ Geum        : num  0.12 0.55 0.09 0.31 0.17 0.97 0.4 0.01 0.47 0.33 ...
##  $ All_Natives : num  67 50.2 61 40.9 38.4 ...
##  $ Fallopia    : num  0.01 0.04 0.09 0.77 3.4 0.54 2.05 0.26 0 0 ...
##  $ Total       : num  67.1 50.2 61.1 41.7 41.8 ...
##  $ Pct_Fallopia: num  0.01 0.08 0.15 1.85 8.13 1.12 3.7 0.61 0 0 ...
```

```
head(MyData)
```

```
##   PotNum Scenario Nutrients Taxon Symphytum Silene Urtica Geranium Geum
## 1      1      low       low japon      9.81  36.36  16.08     4.68 0.12
## 2      2      low       low japon      8.64  29.65   5.59     5.75 0.55
## 3      3      low       low japon      2.65  36.03  17.09     5.13 0.09
## 4      5      low       low japon      1.44  21.43  12.39     5.37 0.31
## 5      6      low       low japon      9.15  23.90   5.19     0.00 0.17
## 6      7      low       low japon      6.31  24.40   7.00     9.05 0.97
##   All_Natives Fallopia Total Pct_Fallopia
## 1       67.05     0.01 67.06         0.01
## 2       50.18     0.04 50.22         0.08
## 3       60.99     0.09 61.08         0.15
## 4       40.94     0.77 41.71         1.85
## 5       38.41     3.40 41.81         8.13
## 6       47.73     0.54 48.27         1.12
```

```
class(MyData)
```

```
## [1] "data.frame"
```

```
dim(MyData)
```

```
## [1] 123  13
```

```
tail(MyData)
```

```
##     PotNum     Scenario Nutrients Taxon Symphytum Silene Urtica Geranium Geum
## 118    143 fluctuations      high bohem      5.06  12.81  23.82     3.64 0.16
## 119    144 fluctuations      high bohem     19.93  21.07   6.08     2.80 0.43
## 120    145 fluctuations      high bohem      4.89  32.93   6.30     9.64 0.00
## 121    147 fluctuations      high bohem      7.84  31.16  13.61     6.58 0.03
## 122    148 fluctuations      high bohem      4.15  38.70  23.59     5.11 1.36
## 123    149 fluctuations      high bohem      1.72  10.41  23.48     8.51 0.43
##     All_Natives Fallopia Total Pct_Fallopia
## 118       45.49    21.31 66.80        31.90
## 119       50.31     0.00 50.31         0.00
## 120       53.76     2.36 56.12         4.21
## 121       59.22     3.74 62.96         5.94
## 122       72.91     5.89 78.80         7.47
## 123       44.55    19.70 64.25        30.66
```

```
summary(MyData)
```

```
##      PotNum         Scenario          Nutrients            Taxon
##  Min.   :  1.00   Length:123         Length:123         Length:123
##  1st Qu.: 38.50   Class :character   Class :character   Class :character
##  Median : 77.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 75.48
##  3rd Qu.:112.50
##  Max.   :149.00
##    Symphytum          Silene          Urtica         Geranium
##  Min.   : 0.000   Min.   : 0.00   Min.   : 0.00   Min.   : 0.000
##  1st Qu.: 5.715   1st Qu.:19.34   1st Qu.: 4.76   1st Qu.: 3.980
##  Median : 9.040   Median :26.64   Median : 8.59   Median : 6.430
##  Mean   : 9.113   Mean   :26.35   Mean   :10.29   Mean   : 6.513
##  3rd Qu.:11.870   3rd Qu.:32.43   3rd Qu.:15.10   3rd Qu.: 8.885
##  Max.   :21.820   Max.   :60.93   Max.   :41.08   Max.   :24.640
##       Geum         All_Natives       Fallopia          Total
##  Min.   :0.0000   Min.   :24.07   Min.   : 0.000   Min.   :29.54
##  1st Qu.:0.0650   1st Qu.:45.09   1st Qu.: 0.000   1st Qu.:49.65
##  Median :0.2100   Median :53.37   Median : 2.240   Median :56.70
##  Mean   :0.3431   Mean   :52.61   Mean   : 3.727   Mean   :56.33
##  3rd Qu.:0.4750   3rd Qu.:60.12   3rd Qu.: 5.405   3rd Qu.:63.13
##  Max.   :1.8000   Max.   :77.05   Max.   :24.580   Max.   :78.80
##   Pct_Fallopia
##  Min.   : 0.000
##  1st Qu.: 0.000
##  Median : 4.050
##  Mean   : 6.508
##  3rd Qu.: 9.415
##  Max.   :43.630
```

Remove rows with 'Total' biomass < 60

```
MyData <- MyData %>%
  filter(Total >= 60)
```

Reorder the columns so that they are in the order: 'Total', 'Taxon', 'Scenario', 'Nutrients', and remove the other columns

```
SubData <- select(MyData, Total, Taxon, Scenario, Nutrients)
head(SubData)
```

```
##   Total Taxon Scenario Nutrients
## 1 67.06 japon      low       low
## 2 61.08 japon      low       low
## 3 60.82 japon     high      high
## 4 66.74 japon     high      high
## 5 63.18 japon     high      high
## 6 61.31 japon     high      high
```

Make a new column TotalG, which converts the 'Total' column from mg to grams AND replace Total with TotalG, and add it to the dataset.

We know that converts the 1 mg equal to 0.001 grams. Here I add a new column TotalG with different units(grams), and TotalG is Total divide 1000.

```
NewData <- transmute(SubData,TotalG = Total / 1000, Taxon, Scenario, Nutrients)
head(NewData)
```

```
##    TotalG Taxon Scenario Nutrients
## 1 0.06706 japon      low       low
## 2 0.06108 japon      low       low
## 3 0.06082 japon     high      high
## 4 0.06674 japon     high      high
## 5 0.06318 japon     high      high
## 6 0.06131 japon     high      high
```

Write a custom function that will take two inputs from the user:

1. a vector of data to process (e.g. column from a data.frame object) and
2. a string that defines what calculation to perform. if string #2 is "Average" then calculate the average value for the column named in vector #1 if string #2 is "Sum" then calculate the sum of values for the column named in vector #1 if string #2 is "Observations" then count the number of observed values for the column named in vector #1 if string #2 is anything else, then output an error to the user

```
# Write a custom function have two inputs from the user: vec1 is a vector, and str1 is a string.
# used else if to achieve the function
my_function <- function(vec1, str1){
  if (str1 == "Average"){
    Average <- mean(vec1)
    return(Average)
  }else if (str1 == "Sum"){
    Sum <- sum(vec1)
    return(Sum)
  }else if (str1 == "Observation"){
    Count <- length(vec1)
    return (Count)
  }else {
    cat("Error! You can choose from Average/Sum/Observation.")
  }
}
```

Uses the function to count the total number of observations in the 'Taxon' column

```
cat("The total number of observations is", my_function(NewData$Taxon,"Observation"))
```

```
## The total number of observations is 45
```

Uses the function to calculate the average TotalG for each of the two Nutrient concentrations

```
NewData %>% group_by(Nutrients) %>%
  summarise(Average_TotalG = my_function(TotalG, "Average"))
```

```
## # A tibble: 2 x 2
##   Nutrients Average_TotalG
##   <chr>              <dbl>
## 1 high              0.0665
## 2 low               0.0641
```

Write (i.e. save) the new data to a file called "WrangledData.csv" in the Output folder.

```
write.csv(NewData, "Output/WrangledData.csv", row.names = FALSE)
```

Test the function

```
Vector <- c(1,2,3,4,5,6,7,8,9,10)
# here I used this vector to test my function.  The output should be average = 5.5, sum = 55, Observati
my_function(Vector, "Average")
```

```
## [1] 5.5
```

```
my_function(Vector, "Sum")
```

```
## [1] 55
```

```
my_function(Vector, "Observation")
```

```
## [1] 10
```