BIOL 432 - R-eproducibility

Roshael Chellappah (20103016)

26/01/2022

Project Information
git hub username: RoshaelC
git hub repository: https://github.com/RoshaelC/r-eproducibility.git (https://github.com/RoshaelC/r-eproducibility.git)

Project Set Up

```
library(dplyr)
```

```
##
## 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
```

```
data <- read.csv("./InputData/FallopiaData.csv")
```

Dataset Prep

Remove rows with “Total” biomass < 60

```
MyData <- subset(data, Total > 60)
```

Reorder the columns in the order “Total”, “Taxon”, “Scenario”, “Nutrients” and remove other columns

```
colnames(MyData) #identify what the current column names are
```

```
##  [1] "PotNum"       "Scenario"     "Nutrients"    "Taxon"        "Symphytum"
##  [6] "Silene"       "Urtica"       "Geranium"     "Geum"         "All_Natives"
## [11] "Fallopia"     "Total"        "Pct_Fallopia"
```

```
MyData <- transmute(MyData, Total, Taxon, Scenario, Nutrients) #transmute = mutate() + select()
colnames(MyData)
```

```
## [1] "Total"     "Taxon"     "Scenario"  "Nutrients"
```

Replace “Total” with new column with “TotalG”, converting values from mg to g

```
MyData <- transmute(MyData, TotalG = Total*0.001, Taxon, Scenario, Nutrients) # 1 mg is 0.001 g
```

Custom Function

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)
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 s2 is “Sum” then calculate the sum of values for the column named in v1
- if s2 is “Observations” then count the number of observed values for the column named in v1
- if s2 is anything else, then outut an error to the user

```
my.function <- function(datacol = 0, calc = 0){
  cat("\n**User input** \nvector data:", datacol, "\ncalculation:", calc, "\n")

  if (calc == "Average"){
    cat("\nCalculating average...\n")
    mean <- mean(datacol)
    return(average = mean)
  }

  else if (calc == "Sum"){
    cat("\nCalculating sum...\n")
    sumvector <- sum(datacol)
    return(sum = sumvector)
  }

  else if (calc == "Observations"){
    cat("\nCalculating number of observations...\n")
    obs <- NROW(datacol)
    return(observations = obs)
  }

  else{
    cat("\nUnrecognized calculation, please use 'Average', 'Sum' or 'Observations'...\n")
  }
}
```

Write some R code that uses your function to count the total number of observations in the “Taxon” column

```
my.function(datacol = MyData$Taxon, calc = "Observations")
```

```
##
## **User input**
## vector data: japon japon japon japon japon japon bohem bohem bohem bohem bohem bohem japon ja
pon japon japon japon japon bohem bohem bohem bohem bohem japon japon japon japon bohem bohem bo
hem bohem bohem bohem bohem japon japon japon japon japon japon japon bohem bohem bohem bohem
## calculation: Observations
##
## Calculating number of observations...
```

```
## [1] 45
```

Write some R code that uses your function to calculate the average TotalG for each of the two Nutrient concentrations

```
WData <- MyData %>%
  group_by(Nutrients) %>% #groups MyData by Nutrient values 'high' and 'low'
  summarize("Average TotalG" = my.function(datacol = TotalG, calc = "Average")) #summarizes the
 results of my.function() calculating TotalG averages for 'high' vs. 'low' Nutrient observations
```

```
##
## **User input**
## vector data: 0.06082 0.06674 0.06318 0.06131 0.06471 0.06106 0.0642 0.06855 0.0666 0.07482 0.
06606 0.06701 0.07041 0.06343 0.07705 0.06179 0.06256 0.06954 0.07591 0.06497 0.06071 0.06744 0.
07774 0.06568 0.07107 0.0677 0.06086 0.06219 0.06553 0.06089 0.06037 0.06786 0.069 0.0654 0.0630
8 0.06093 0.06873 0.06988 0.06948 0.0668 0.06296 0.0788 0.06425
## calculation: Average
##
## Calculating average...
##
## **User input**
## vector data: 0.06706 0.06108
## calculation: Average
##
## Calculating average...
```

```
WData
```

```
## # 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(WData, "./Output/WrangledData.csv")
```