review

Azul Carrillo

2022-11-29

#1 Write a conditional statement that checks if surveys.csv exists in the working directory, if it doesn't then downloads it from https://ndownloader.figshare.com/files/2292172 using download.file(), and finally loads the file into a data frame and displays the first few rows using the head() function. The url needs to be in quotes since it is character data.

```
getwd()
```

```
## [1] "/Users/atziri/Bio 195-197/Data Science/documents"
```

```
"surveys.csv" == c(list.files("/Users/atziri/Bio 195-197/Data Science/raw-data"))
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [25]  TRUE FALSE FALSE
```

```
surveys <- list.files("/Users/atziri/Bio 195-197/Data Science/raw-data") == "surveys.csv"


is.element("surveys.csv", list.files("/Users/atziri/Bio 195-197/Data Science/raw-data") )
```

```
## [1] TRUE
```

```
is.element("surveys.csv", list.files("/Users/atziri/Bio 195-197/Data Science/raw-data") )
```

```
## [1] TRUE
```

```
surveys <- list.files("../raw-data")

if (is.element("surveys.csv", surveys)){
  print("file is downloaded")
} else {
   print("file is not downloaded")
  download.file("https://ndownloader.figshare.com/files/2292172",
                "../raw-data/surveys-download.csv")
surveys_data <- read.csv("../raw-data/surveys-download.csv")
 head(surveys_data)
}
```

```
## [1] "file is downloaded"
```

#2 Make a version of this conditional statement that is a function, where the name of the file is the first argument and the link for downloading the file is the second argumen

```
#This function tests if a file is in the raw-data directory and if not it download it and read it as da

reading_csv <- function(file_name, file_link) {

# 1. test if file_name is in the raw-data folder
# file_name <- "species.csv"

test <- !is.element(file_name, list.files(path = "../raw-data"))

# 2. if test is FALSE, download the file

if (test) {
  # Option 1: save it with a random name:
  # download.file(url = file_link, destfile = "../raw-data/temporary.csv")

  # result <- read.csv(file = "../raw-data/temporary.csv")

# Option 2: save it with the name given in file nome:
  destination_file <- stringr::str_c("../raw-data/", file_name)
  download.file(url = file_link, destfile = destination_file)
  result <- read.csv(file = destination_file)
}
  return(result)
}
```

```
reading_csv <- function(file_name, file_link) {
  test <- is.element(file_name, list.files(path = "../raw-data"))# removed the exclamation mark for it
  if (test) {
    destination_file <- stringr:: str_c("../raw-data/", file_name)
    download.file(url = file_link, destfile = destination_file)

    result <- read.csv(file = destination_file)
  }
   return(result)
}
```

```
reading_csv(file_name = "species.csv",
            file_link = "https://ndownloade.figshare.com/files/3299483")
```

```
##    species_id           genus         species    taxa
## 1          AB       Amphispiza       bilineata    Bird
## 2          AH Ammospermophilus         harrisi  Rodent
## 3          AS       Ammodramus      savannarum    Bird
## 4          BA          Baiomys         taylori  Rodent
## 5          CB  Campylorhynchus brunneicapillus    Bird
## 6          CM      Calamospiza     melanocorys    Bird
## 7          CQ       Callipepla        squamata    Bird
## 8          CS         Crotalus      scutalatus Reptile
## 9          CT    Cnemidophorus          tigris Reptile
## 10         CU    Cnemidophorus       uniparens Reptile
```

```
## 11         CV         Crotalus         viridis Reptile
## 12         DM        Dipodomys        merriami  Rodent
## 13         DO        Dipodomys           ordii  Rodent
## 14         DS        Dipodomys     spectabilis  Rodent
## 15         DX        Dipodomys             sp.  Rodent
## 16         EO          Eumeces       obsoletus Reptile
## 17         GS         Gambelia           silus Reptile
## 18         NL          Neotoma        albigula  Rodent
## 19         NX          Neotoma             sp.  Rodent
## 20         OL        Onychomys     leucogaster  Rodent
## 21         OT        Onychomys        torridus  Rodent
## 22         OX        Onychomys             sp.  Rodent
## 23         PB      Chaetodipus         baileyi  Rodent
## 24         PC           Pipilo       chlorurus    Bird
## 25         PE       Peromyscus        eremicus  Rodent
## 26         PF      Perognathus          flavus  Rodent
## 27         PG        Pooecetes       gramineus    Bird
## 28         PH      Perognathus         hispidus  Rodent
## 29         PI      Chaetodipus     intermedius  Rodent
## 30         PL       Peromyscus        leucopus  Rodent
## 31         PM       Peromyscus     maniculatus  Rodent
## 32         PP      Chaetodipus    penicillatus  Rodent
## 33         PU           Pipilo          fuscus    Bird
## 34         PX      Chaetodipus             sp.  Rodent
## 35         RF  Reithrodontomys      fulvescens  Rodent
## 36         RM  Reithrodontomys       megalotis  Rodent
## 37         RO  Reithrodontomys        montanus  Rodent
## 38         RX  Reithrodontomys             sp.  Rodent
## 39         SA       Sylvilagus       audubonii  Rabbit
## 40         SB         Spizella         breweri    Bird
## 41         SC       Sceloporus          clarki Reptile
## 42         SF         Sigmodon     fulviventer  Rodent
## 43         SH         Sigmodon         hispidus  Rodent
## 44         SO         Sigmodon    ochrognathus  Rodent
## 45         SS     Spermophilus       spilosoma  Rodent
## 46         ST     Spermophilus    tereticaudus  Rodent
## 47         SU       Sceloporus       undulatus Reptile
## 48         SX         Sigmodon             sp.  Rodent
## 49         UL           Lizard             sp. Reptile
## 50         UP           Pipilo             sp.    Bird
## 51         UR           Rodent             sp.  Rodent
## 52         US          Sparrow             sp.    Bird
## 53         ZL      Zonotrichia      leucophrys    Bird
## 54         ZM          Zenaida        macroura    Bird
```

#Exercise 2: Multi-file Analysis

##1. If individual_collar_data.zip is not already in your working directory download the zip file using download.file()

##2. Unzip it using unzip()

##3. Obtain a list of all of the files with file names matching the pattern "collar-data-.*.txt" (using list.files())

```
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
```

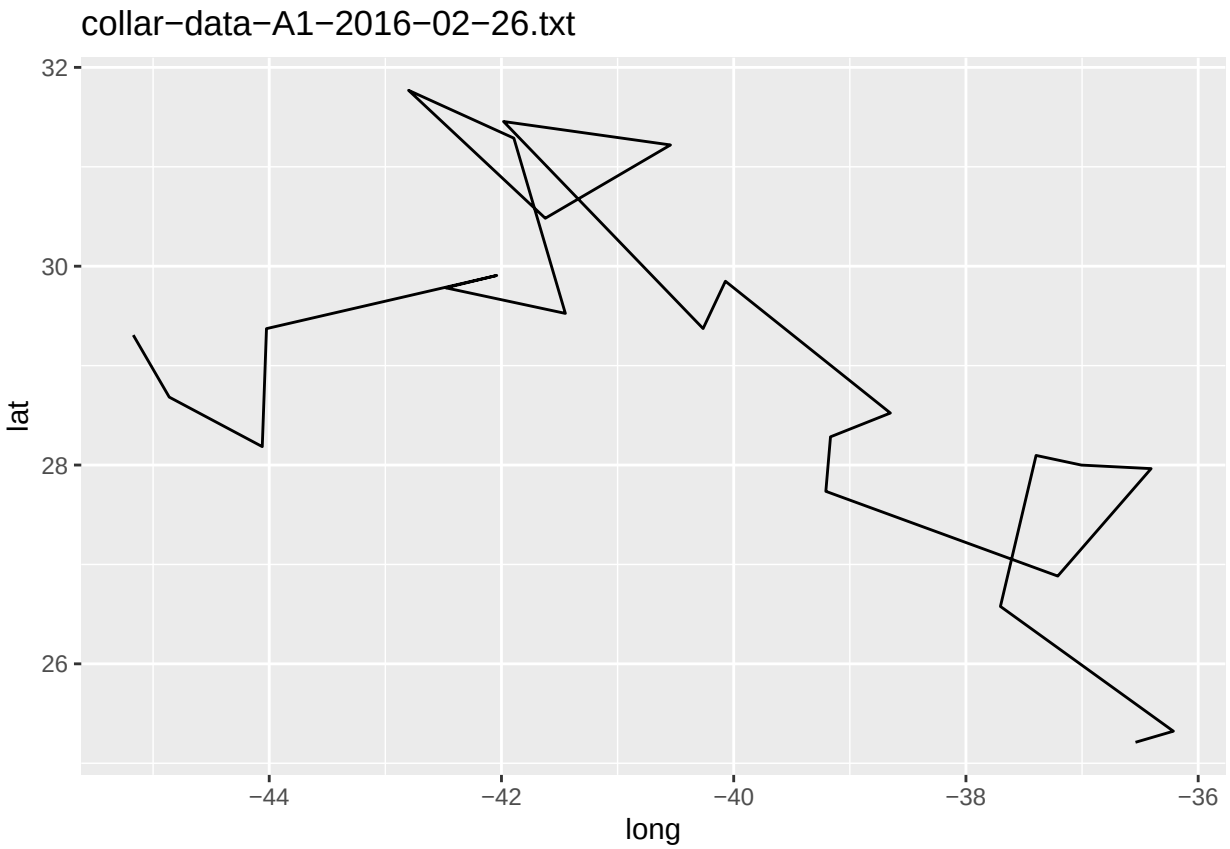
```
library(ggplot2)
download.file("http://www.datacarpentry.org/semester-biology/data/individual_collar_data.zip", "collar.z
unzip("collar.zip", exdir = "../raw-data")
collar_data_files <- list.files(pattern = "collar-")
collar_data_files
```

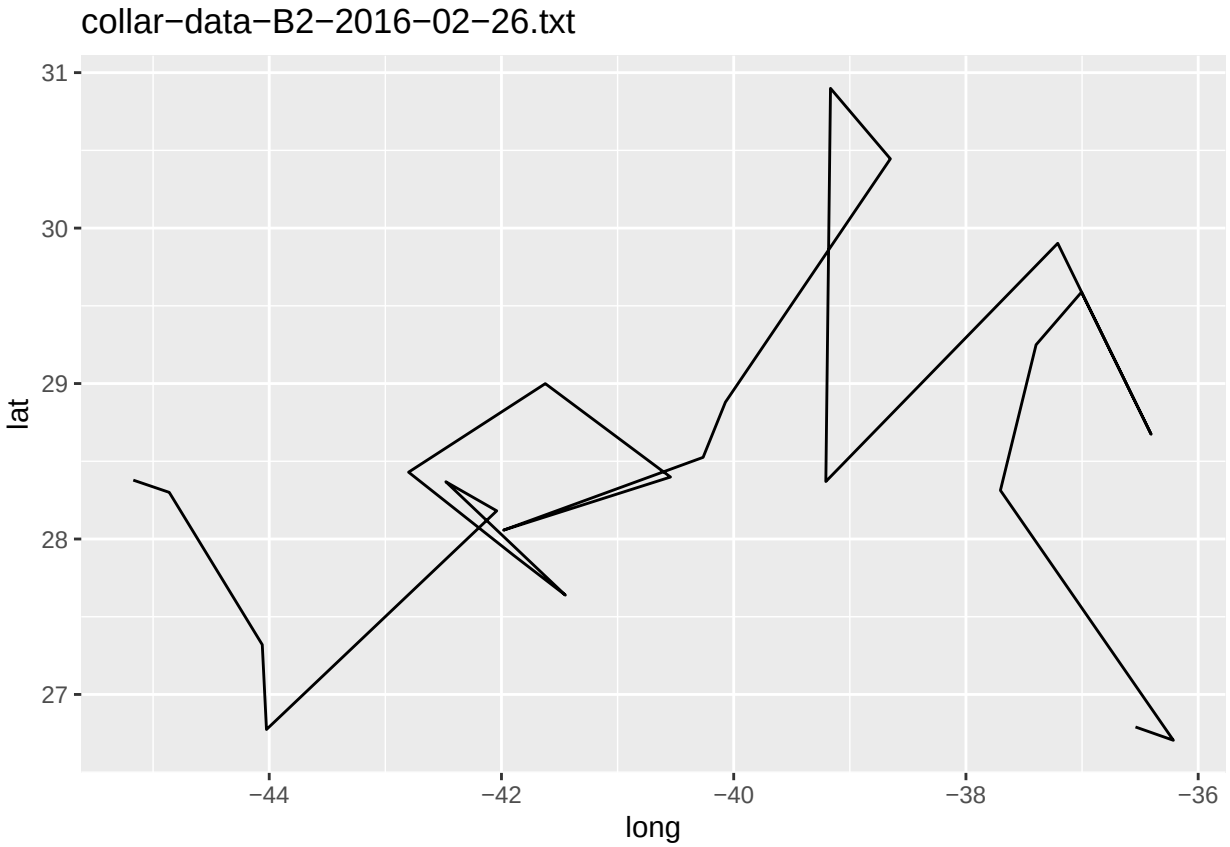
```
##  [1] "collar-data-A1-2016-02-26.txt"  "collar-data-B2-2016-02-26.txt"
##  [3] "collar-data-C3-2016-02-26.txt"  "collar-data-D4-2016-02-26.txt"
##  [5] "collar-data-E5-2016-02-26.txt"  "collar-data-F6-2016-02-26.txt"
##  [7] "collar-data-G7-2016-02-26.txt"  "collar-data-H8-2016-02-26.txt"
##  [9] "collar-data-I9-2016-02-26.txt"  "collar-data-J10-2016-02-26.txt"
```

##4. Use a loop to load each of these files into R and make a line plot (using geom_path()) for each file with long on the x axis and lat on the y axis. Graphs, like other types of output, won't display inside a loop unless you explicitly display them, so you need put your ggplot() command inside a print() statement. Include the name of the file in the graph as the graph title using labs().

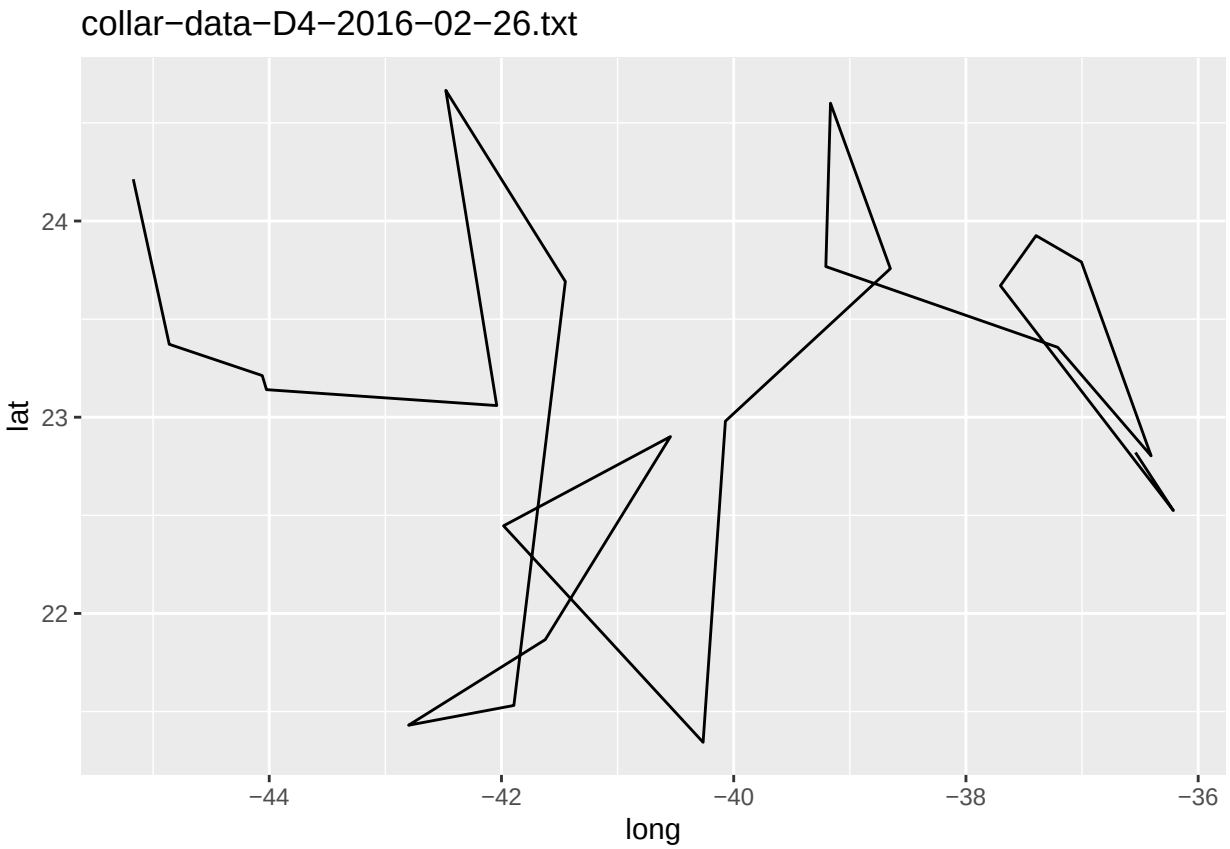
##5. Add code to the loop to calculate the minimum and maximum latitude in the file, and store these values, along with the name of the file, in a data frame. Show the data frame as output.

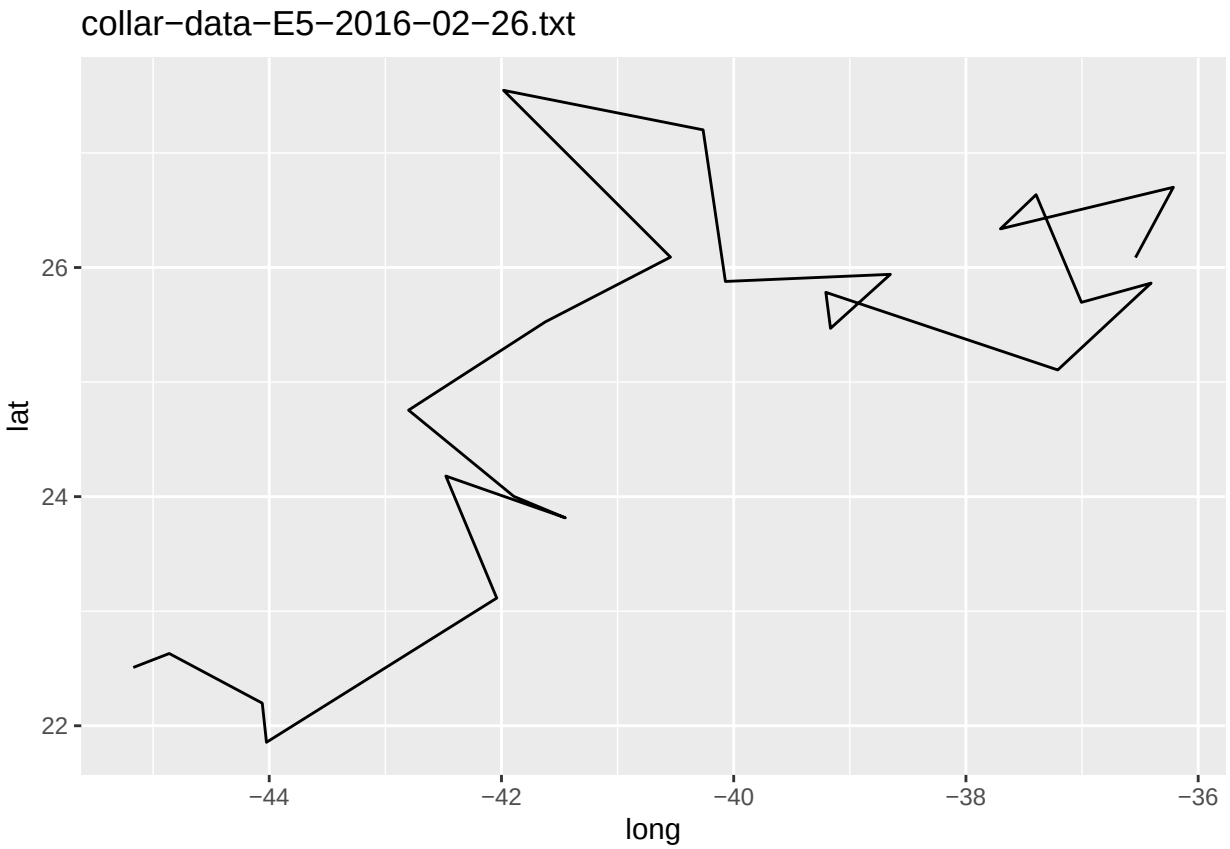
```
min_results <- vector(mode = "integer", length(collar_data_files))
max_results <- vector(mode = "integer", length(collar_data_files))


for(i in 1:length(collar_data_files)){
  filename <- collar_data_files[i]
  data <- read.csv(filename)
  data %>%
    ggplot(aes(x = long, y = lat )) +
    labs(title = filename) +
  geom_path() -> plots
    print(plots)
   min_results[i] <- min(data$lat)
   max_results[i] <- max(data$lat)
}
```

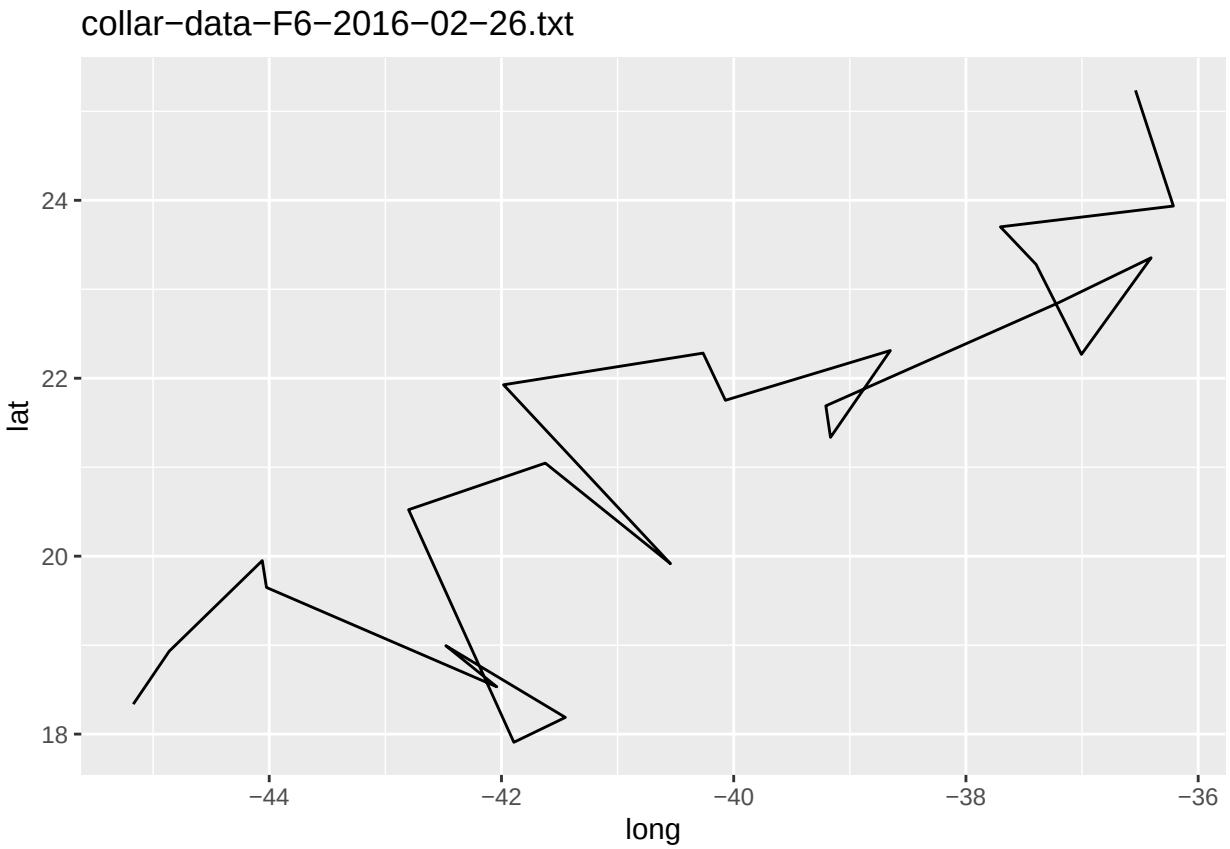

collar−data−A1−2016−02−26.txt
32
30
28
26
lat
−44
−42
−40
−38
−36
long


collar−data−B2−2016−02−26.txt
31
30
29
28
27
lat
−44
−42
−40
−38
−36
long

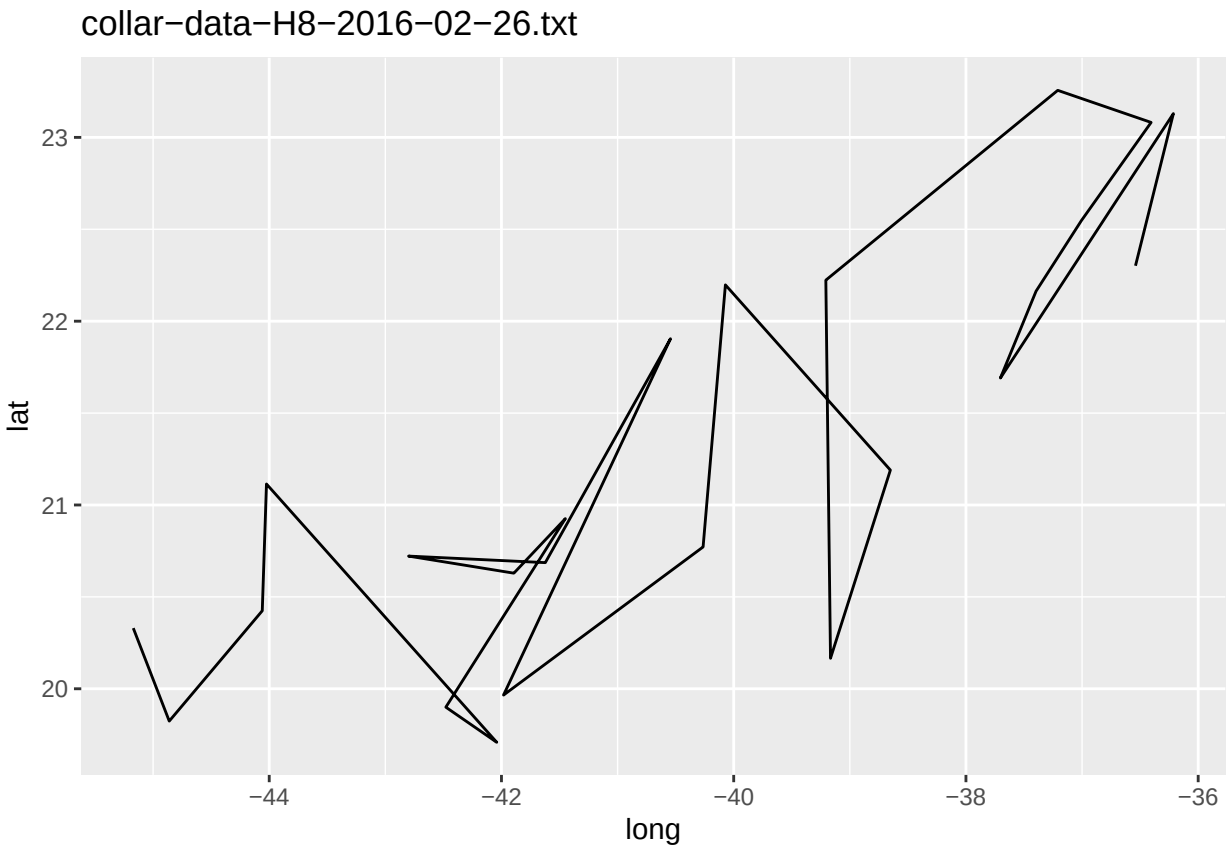

collar−data−C3−2016−02−26.txt
33
32
31
30
29
lat
−44
−42
−40
−38
−36
long


collar−data−D4−2016−02−26.txt
lat
24
23
22
−44
−42
−40
−38
−36
long

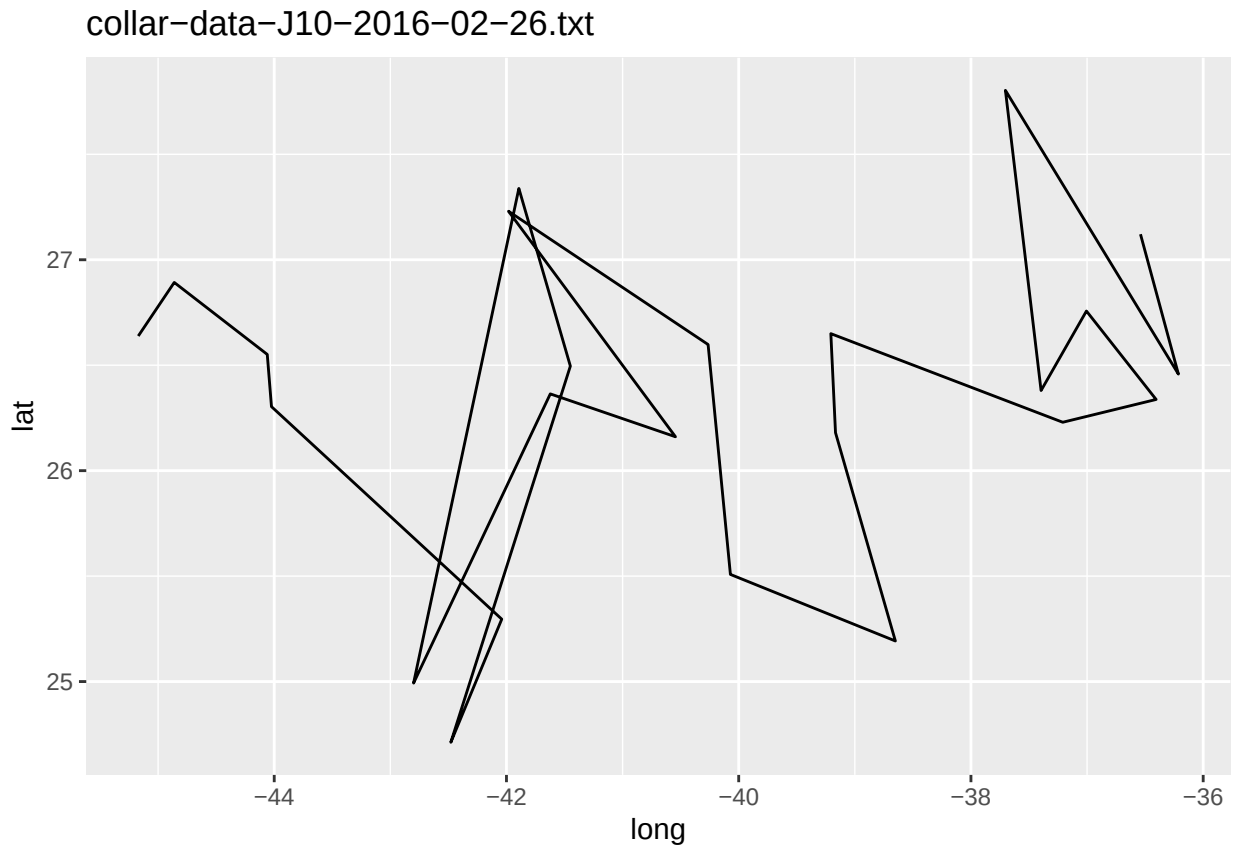

collar−data−E5−2016−02−26.txt
lat
26
24
22
−44
−42
−40
−38
−36
long


collar−data−F6−2016−02−26.txt
24
22
20
18
lat
−44
−42
−40
−38
−36
long


collar−data−G7−2016−02−26.txt
lat
31
30
29
28
−44
−42
−40
−38
−36
long


collar−data−H8−2016−02−26.txt
lat
23
22
21
20
−44
−42
−40
−38
−36
long


collar−data−I9−2016−02−26.txt
28.5
28.0
27.5
27.0
26.5
26.0
lat
−44
−42
−40
−38
−36
long




```
min_max_data <- data.frame(collar_data_files, min_results, max_results)

min_max_data
```

```
##                 collar_data_files min_results max_results
## 1   collar-data-A1-2016-02-26.txt    25.21080    31.76912
## 2   collar-data-B2-2016-02-26.txt    26.70509    30.89907
## 3   collar-data-C3-2016-02-26.txt    28.86998    33.44421
## 4   collar-data-D4-2016-02-26.txt    21.34315    24.66598
## 5   collar-data-E5-2016-02-26.txt    21.85565    27.54663
## 6   collar-data-F6-2016-02-26.txt    17.90788    25.23623
## 7   collar-data-G7-2016-02-26.txt    27.67120    31.63272
## 8   collar-data-H8-2016-02-26.txt    19.70875    23.25601
## 9   collar-data-I9-2016-02-26.txt    25.70252    28.49172
## 10 collar-data-J10-2016-02-26.txt    24.71200    27.80325
```

##Solution 2

```
all_min <-  vector()
all_max <- vector()
length(all_min)
```

```
## [1] 0
```

```
file_name <- vector()

for(i in collar_data_files){
  print(i)
  print(getwd())
  file_name <- c(file_name,i)
  collar_data_table <- read.csv(file = i )
  min_lat <- min (collar_data_table[,"lat"])
  all_min <- c(all_min,min_lat)
  max_lat <- max(collar_data_table[,"lat"])
  all_max <- c(all_max, max_lat)
  print(all_min)
  print(all_max)
}
```

```
## [1] "collar-data-A1-2016-02-26.txt"
## [1] "/Users/atziri/Bio 195-197/Data Science/documents"
## [1] 25.2108
## [1] 31.76912
## [1] "collar-data-B2-2016-02-26.txt"
## [1] "/Users/atziri/Bio 195-197/Data Science/documents"
## [1] 25.21080 26.70509
## [1] 31.76912 30.89907
## [1] "collar-data-C3-2016-02-26.txt"
## [1] "/Users/atziri/Bio 195-197/Data Science/documents"
## [1] 25.21080 26.70509 28.86998
## [1] 31.76912 30.89907 33.44421
## [1] "collar-data-D4-2016-02-26.txt"
## [1] "/Users/atziri/Bio 195-197/Data Science/documents"
## [1] 25.21080 26.70509 28.86998 21.34315
## [1] 31.76912 30.89907 33.44421 24.66598
## [1] "collar-data-E5-2016-02-26.txt"
## [1] "/Users/atziri/Bio 195-197/Data Science/documents"
## [1] 25.21080 26.70509 28.86998 21.34315 21.85565
## [1] 31.76912 30.89907 33.44421 24.66598 27.54663
## [1] "collar-data-F6-2016-02-26.txt"
## [1] "/Users/atziri/Bio 195-197/Data Science/documents"
## [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788
## [1] 31.76912 30.89907 33.44421 24.66598 27.54663 25.23623
## [1] "collar-data-G7-2016-02-26.txt"
## [1] "/Users/atziri/Bio 195-197/Data Science/documents"
## [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120
## [1] 31.76912 30.89907 33.44421 24.66598 27.54663 25.23623 31.63272
## [1] "collar-data-H8-2016-02-26.txt"
## [1] "/Users/atziri/Bio 195-197/Data Science/documents"
## [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120 19.70875
## [1] 31.76912 30.89907 33.44421 24.66598 27.54663 25.23623 31.63272 23.25601
## [1] "collar-data-I9-2016-02-26.txt"
## [1] "/Users/atziri/Bio 195-197/Data Science/documents"
## [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120 19.70875
## [9] 25.70252
## [1] 31.76912 30.89907 33.44421 24.66598 27.54663 25.23623 31.63272 23.25601
## [9] 28.49172
```

```
## [1] "collar-data-J10-2016-02-26.txt"
## [1] "/Users/atziri/Bio 195-197/Data Science/documents"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120 19.70875
##  [9] 25.70252 24.71200
##  [1] 31.76912 30.89907 33.44421 24.66598 27.54663 25.23623 31.63272 23.25601
##  [9] 28.49172 27.80325
```

```
min_max_dataframe <- data_frame(file_name, all_min, all_max)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## i Please use `tibble()` instead.
```

```
min_max_dataframe
```

```
## # A tibble: 10 x 3
##    file_name                       all_min all_max
##    <chr>                             <dbl>   <dbl>
##  1 collar-data-A1-2016-02-26.txt      25.2    31.8
##  2 collar-data-B2-2016-02-26.txt      26.7    30.9
##  3 collar-data-C3-2016-02-26.txt      28.9    33.4
##  4 collar-data-D4-2016-02-26.txt      21.3    24.7
##  5 collar-data-E5-2016-02-26.txt      21.9    27.5
##  6 collar-data-F6-2016-02-26.txt      17.9    25.2
##  7 collar-data-G7-2016-02-26.txt      27.7    31.6
##  8 collar-data-H8-2016-02-26.txt      19.7    23.3
##  9 collar-data-I9-2016-02-26.txt      25.7    28.5
## 10 collar-data-J10-2016-02-26.txt     24.7    27.8
```

```
#solution 3
library(stringr)
all_min_lat <- vector(mode = "integer", length = length(collar_data_files))
all_max_lat <- all_min_lat
all_file_names <- all_min_lat
length(all_min)
```

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

```
length(all_max)
```

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

```
all_file_names
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

```
for (i in 1:length(collar_data_files)){
  file_name_and_path <- str_c("../raw-data/",collar_data_files[i])
  all_file_names[i] <- file_name_and_path
  print(file_name_and_path)
  collar_data_table <- read.csv(file = file_name_and_path)
```

```
  min_lat <- min(collar_data_table$lat)
  all_min_lat[i] <- min_lat
  max_lat <- max(collar_data_table$lat)
  all_max_lat[i] <- max_lat
  print(all_min_lat)
  print(all_max_lat)
}
```

```
## [1] "../raw-data/collar-data-A1-2016-02-26.txt"
##  [1] 25.2108  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
## [10]  0.0000
##  [1] 31.76912  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
## [1] "../raw-data/collar-data-B2-2016-02-26.txt"
##  [1] 25.21080 26.70509  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
##  [1] 31.76912 30.89907  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
## [1] "../raw-data/collar-data-C3-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998  0.00000  0.00000  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
##  [1] 31.76912 30.89907 33.44421  0.00000  0.00000  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
## [1] "../raw-data/collar-data-D4-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315  0.00000  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
##  [1] 31.76912 30.89907 33.44421 24.66598  0.00000  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
## [1] "../raw-data/collar-data-E5-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
##  [1] 31.76912 30.89907 33.44421 24.66598 27.54663  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
## [1] "../raw-data/collar-data-F6-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788  0.00000  0.00000
##  [9]  0.00000  0.00000
##  [1] 31.76912 30.89907 33.44421 24.66598 27.54663 25.23623  0.00000  0.00000
##  [9]  0.00000  0.00000
## [1] "../raw-data/collar-data-G7-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120  0.00000
##  [9]  0.00000  0.00000
##  [1] 31.76912 30.89907 33.44421 24.66598 27.54663 25.23623 31.63272  0.00000
##  [9]  0.00000  0.00000
## [1] "../raw-data/collar-data-H8-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120 19.70875
##  [9]  0.00000  0.00000
##  [1] 31.76912 30.89907 33.44421 24.66598 27.54663 25.23623 31.63272 23.25601
##  [9]  0.00000  0.00000
## [1] "../raw-data/collar-data-I9-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120 19.70875
##  [9] 25.70252  0.00000
##  [1] 31.76912 30.89907 33.44421 24.66598 27.54663 25.23623 31.63272 23.25601
##  [9] 28.49172  0.00000
```

```
## [1] "../raw-data/collar-data-J10-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120 19.70875
##  [9] 25.70252 24.71200
##  [1] 31.76912 30.89907 33.44421 24.66598 27.54663 25.23623 31.63272 23.25601
##  [9] 28.49172 27.80325
```

```
min_max_dataframee <- data_frame(all_file_names, all_min_lat, all_max_lat)
min_max_dataframee
```

```
## # A tibble: 10 x 3
##    all_file_names                              all_min_lat all_max_lat
##    <chr>                                             <dbl>       <dbl>
##  1 ../raw-data/collar-data-A1-2016-02-26.txt          25.2        31.8
##  2 ../raw-data/collar-data-B2-2016-02-26.txt          26.7        30.9
##  3 ../raw-data/collar-data-C3-2016-02-26.txt          28.9        33.4
##  4 ../raw-data/collar-data-D4-2016-02-26.txt          21.3        24.7
##  5 ../raw-data/collar-data-E5-2016-02-26.txt          21.9        27.5
##  6 ../raw-data/collar-data-F6-2016-02-26.txt          17.9        25.2
##  7 ../raw-data/collar-data-G7-2016-02-26.txt          27.7        31.6
##  8 ../raw-data/collar-data-H8-2016-02-26.txt          19.7        23.3
##  9 ../raw-data/collar-data-I9-2016-02-26.txt          25.7        28.5
## 10 ../raw-data/collar-data-J10-2016-02-26.txt         24.7        27.8
```

#Exercise 3: A function for the UHURU data set

##1. Explain what each line of code in the body of the function is doing. Add the explanations to your Rmd file as comments, before each line of code.

```
report_rsquared <- function(data, species, formula){
  subset <- dplyr::filter(data, ANT == species)#subset the ANT row name by a logical condition changing
  test <- lm(formula, data = subset) #lm fun stands for linear models is used to fit linear models to d
  rsquared <- round(summary(test)$r.squared, 3)#round function rounds off values to a specific number o
  output <- data.frame(species = species, r2 = rsquared)#creates a data frame from the arguments given
  return(output)
}
```

##2. Execute the function using the UHURU data and specifying species = "CM" and formula = "AXIS1~CIRC".

```
UHURU_data <- read.csv(file = "../raw-data/ACACIA_DREPANOLOBIUM_SURVEY.txt", sep = "\t")
UHURU_data
```

```
##     SURVEY YEAR  SITE BLOCK TREATMENT    PLOT   ID HEIGHT AXIS1 AXIS2 CIRC
## 1        1 2012 SOUTH     1     TOTAL S1TOTAL  581   2.25  2.75  2.15 20.0
## 2        1 2012 SOUTH     1     TOTAL S1TOTAL  582   2.65  4.10  3.90 28.0
## 3        1 2012 SOUTH     1     TOTAL S1TOTAL 3111    1.5  1.70  0.85 17.0
## 4        1 2012 SOUTH     1     TOTAL S1TOTAL 3112   2.01  1.80  1.60 12.0
## 5        1 2012 SOUTH     1     TOTAL S1TOTAL 3113   1.75  1.84  1.42 13.0
## 6        1 2012 SOUTH     1     TOTAL S1TOTAL 3114   1.65  1.62  0.85 15.0
## 7        1 2012 SOUTH     1     TOTAL S1TOTAL 3115    1.2  1.95  0.90  9.0
## 8        1 2012 SOUTH     1     TOTAL S1TOTAL 3199   1.45  2.00  1.75 12.2
## 9        1 2012 SOUTH     1      MESO  S1MESO  941   1.87  2.15  1.82 13.0
## 10       1 2012 SOUTH     1      MESO  S1MESO  942   2.38  5.55  4.82 35.0
```

```
## 11       1 2012 SOUTH     1      MESO  S1MESO  943   2.58  4.90  4.24 24.0
## 12       1 2012 SOUTH     1      MESO  S1MESO  944   2.65  3.75  3.10 27.0
## 13       1 2012 SOUTH     1      MESO  S1MESO  946   2.35  2.34  2.05 20.0
## 14       1 2012 SOUTH     1      MESO  S1MESO  947   1.88  2.10  1.85 28.0
## 15       1 2012 SOUTH     1      MESO  S1MESO 3116   2.32  3.05  2.63 30.0
## 16       1 2012 SOUTH     1      MESO  S1MESO 3117   2.39  2.21  2.10 13.0
## 17       1 2012 SOUTH     1      MESO  S1MESO 3118    2.2  1.80  1.50 10.0
## 18       1 2012 SOUTH     1      MESO  S1MESO 3119   1.05  0.90  0.55  8.0
## 19       1 2012 SOUTH     1      MESO  S1MESO 3120      2  1.25  1.20 10.0
## 20       1 2012 SOUTH     1      MESO  S1MESO 3131   1.28  1.14  1.00 10.0
## 21       1 2012 SOUTH     2      OPEN  S2OPEN  341   dead    NA    NA   NA
## 22       1 2012 SOUTH     2     TOTAL S2TOTAL 3178    1.4  2.50  2.15 18.0
## 23       1 2012 SOUTH     2     TOTAL S2TOTAL  101    1.9  3.31  2.65 15.0
## 24       1 2012 SOUTH     2     TOTAL S2TOTAL  102   1.75  2.70  2.55 16.0
## 25       1 2012 SOUTH     2     TOTAL S2TOTAL  103    1.8  2.75  2.30 16.0
## 26       1 2012 SOUTH     2     TOTAL S2TOTAL  104    2.7  4.05  4.00 35.2
## 27       1 2012 SOUTH     2     TOTAL S2TOTAL  105   2.02  2.85  1.49 17.0
## 28       1 2012 SOUTH     2     TOTAL S2TOTAL  108    1.9  3.10  2.85 19.0
## 29       1 2012 SOUTH     2     TOTAL S2TOTAL  109   1.85  2.45  1.90 19.0
## 30       1 2012 SOUTH     2     TOTAL S2TOTAL  110   1.65  1.90  1.54 17.0
## 31       1 2012 SOUTH     2     TOTAL S2TOTAL  111    1.4  2.35  1.45 14.0
## 32       1 2012 SOUTH     2     TOTAL S2TOTAL  113    2.5  3.25  2.30 22.0
## 33       1 2012 SOUTH     2     TOTAL S2TOTAL  115   2.05  5.40  4.50 33.0
## 34       1 2012 SOUTH     2     TOTAL S2TOTAL  116   2.26  3.50  3.10 33.0
## 35       1 2012 SOUTH     2     TOTAL S2TOTAL  117   2.13  2.40  2.30 20.0
## 36       1 2012 SOUTH     2     TOTAL S2TOTAL  118    1.8  3.15  2.55 22.0
## 37       1 2012 SOUTH     2     TOTAL S2TOTAL 1211   1.85  2.00  2.27 20.0
## 38       1 2012 SOUTH     2     TOTAL S2TOTAL 1212    1.5  2.15  1.80 15.0
## 39       1 2012 SOUTH     2     TOTAL S2TOTAL 1213   1.87  2.34  2.05 13.0
## 40       1 2012 SOUTH     2     TOTAL S2TOTAL 1214   1.58  1.28  0.75 11.0
## 41       1 2012 SOUTH     2     TOTAL S2TOTAL 1215   2.05  2.10  1.75 17.0
## 42       1 2012 SOUTH     2     TOTAL S2TOTAL 1216   1.75  2.45  3.28 16.0
## 43       1 2012 SOUTH     2     TOTAL S2TOTAL 1217   1.49  1.50  1.45 13.0
## 44       1 2012 SOUTH     2     TOTAL S2TOTAL 1218   1.28  2.00  0.90 10.0
## 45       1 2012 SOUTH     2     TOTAL S2TOTAL 1219   1.49  2.35  1.65 13.0
## 46       1 2012 SOUTH     2     TOTAL S2TOTAL 1220   1.07  1.20  0.95 11.0
## 47       1 2012 SOUTH     2     TOTAL S2TOTAL 1231   1.48  1.25  1.20  9.0
## 48       1 2012 SOUTH     2     TOTAL S2TOTAL 1232   1.25  1.25  0.90 10.0
## 49       1 2012 SOUTH     2     TOTAL S2TOTAL 1233   1.41  1.41  1.40 14.0
## 50       1 2012 SOUTH     2     TOTAL S2TOTAL 1234    1.6  1.60  1.30 13.0
## 51       1 2012 SOUTH     2     TOTAL S2TOTAL 1235    1.2  1.20  1.30 14.0
## 52       1 2012 SOUTH     2     TOTAL S2TOTAL 1236   1.49  1.49  1.20  8.0
## 53       1 2012 SOUTH     2     TOTAL S2TOTAL 1237    1.5  1.50  1.50 14.0
## 54       1 2012 SOUTH     2     TOTAL S2TOTAL 1238   1.65  1.65  2.00 20.0
## 55       1 2012 SOUTH     2     TOTAL S2TOTAL 1239   1.13  1.13  1.20 10.0
## 56       1 2012 SOUTH     2     TOTAL S2TOTAL 1240   1.25  1.25  0.90 10.0
## 57       1 2012 SOUTH     2     TOTAL S2TOTAL 1251    1.1  1.20  1.10 10.0
## 58       1 2012 SOUTH     2     TOTAL S2TOTAL 1252    2.2  2.70  2.40 25.0
## 59       1 2012 SOUTH     2     TOTAL S2TOTAL 1253   1.45  1.65  1.25 10.0
## 60       1 2012 SOUTH     2     TOTAL S2TOTAL 1254    1.6  2.45  2.10 13.0
## 61       1 2012 SOUTH     2     TOTAL S2TOTAL 1255   1.55  2.40  1.80 13.0
## 62       1 2012 SOUTH     2     TOTAL S2TOTAL 1256    1.5  2.40  2.15 13.0
## 63       1 2012 SOUTH     2     TOTAL S2TOTAL 1257   1.03  1.20  1.00 10.0
## 64       1 2012 SOUTH     2     TOTAL S2TOTAL 1258   2.14  1.90  1.70 13.0
```

```
## 65       1 2012 SOUTH     2     TOTAL S2TOTAL 1259    1.2  1.90  1.65 12.0
## 66       1 2012 SOUTH     2     TOTAL S2TOTAL 1260   1.05  1.10  1.00  9.0
## 67       1 2012 SOUTH     2     TOTAL S2TOTAL 2131    1.8  2.60  2.40 15.0
## 68       1 2012 SOUTH     2     TOTAL S2TOTAL 2132    1.2  1.00  0.95  7.0
## 69       1 2012 SOUTH     2     TOTAL S2TOTAL 2133   1.75  1.40  1.10 10.0
## 70       1 2012 SOUTH     2     TOTAL S2TOTAL 2134   1.45  3.10  1.80 10.0
## 71       1 2012 SOUTH     2     TOTAL S2TOTAL 2135   1.17  1.20  1.10  5.0
## 72       1 2012 SOUTH     2     TOTAL S2TOTAL 2136   2.15  3.10  2.58 22.0
## 73       1 2012 SOUTH     2     TOTAL S2TOTAL 2137    1.7  1.70  1.40 12.0
## 74       1 2012 SOUTH     2     TOTAL S2TOTAL 3132   1.98  2.85  2.70 12.0
## 75       1 2012 SOUTH     2     TOTAL S2TOTAL 3133   1.26  1.95  1.75 17.0
## 76       1 2012 SOUTH     2     TOTAL S2TOTAL 3134   1.11  1.95  1.50 10.0
## 77       1 2012 SOUTH     2     TOTAL S2TOTAL 3135   1.14  1.32  1.05 10.0
## 78       1 2012 SOUTH     2     TOTAL S2TOTAL 3136   1.26  1.60  1.40 10.0
## 79       1 2012 SOUTH     2     TOTAL S2TOTAL 3137    1.3  1.40  0.80 10.0
## 80       1 2012 SOUTH     2     TOTAL S2TOTAL 3138   1.29  1.44  1.35 13.0
## 81       1 2012 SOUTH     2     TOTAL S2TOTAL 3139   1.31  1.35  1.15  7.0
## 82       1 2012 SOUTH     2     TOTAL S2TOTAL 3140   1.15  1.70  1.28 10.0
## 83       1 2012 SOUTH     2     TOTAL S2TOTAL 3151   1.87  3.40  1.85 15.0
## 84       1 2012 SOUTH     2     TOTAL S2TOTAL 3152   1.47  2.10  1.61  8.0
## 85       1 2012 SOUTH     2     TOTAL S2TOTAL 3153   1.05  1.79  1.50 10.0
## 86       1 2012 SOUTH     2     TOTAL S2TOTAL 3154    2.1  4.90  3.75 25.0
## 87       1 2012 SOUTH     2     TOTAL S2TOTAL 3155   1.99  1.80  1.35 13.0
## 88       1 2012 SOUTH     2     TOTAL S2TOTAL 3156   1.42  1.90  1.80 14.0
## 89       1 2012 SOUTH     2     TOTAL S2TOTAL 3157    1.5  2.11  1.75 12.0
## 90       1 2012 SOUTH     2     TOTAL S2TOTAL 3158   1.06  1.05  0.85  4.0
## 91       1 2012 SOUTH     2     TOTAL S2TOTAL 3159   1.49  1.50  1.15 13.0
## 92       1 2012 SOUTH     2     TOTAL S2TOTAL 3160    1.8  1.60  1.50 14.0
## 93       1 2012 SOUTH     2     TOTAL S2TOTAL 3171   1.93  1.74  1.20 14.0
## 94       1 2012 SOUTH     2     TOTAL S2TOTAL 3172    1.2  1.60  1.30 10.0
## 95       1 2012 SOUTH     2     TOTAL S2TOTAL 3173   1.65  1.25  1.10 11.0
## 96       1 2012 SOUTH     2     TOTAL S2TOTAL 3174   1.52  1.49  1.10 12.0
## 97       1 2012 SOUTH     2     TOTAL S2TOTAL 3175   1.43  2.05  1.54 13.0
## 98       1 2012 SOUTH     2     TOTAL S2TOTAL 3176   1.25  1.40  1.25 13.0
## 99       1 2012 SOUTH     2     TOTAL S2TOTAL 3177   1.88  2.65  2.64 20.0
## 100      1 2012 SOUTH     2     TOTAL S2TOTAL 3179   1.03  1.40  0.60 13.0
## 101      1 2012 SOUTH     2     TOTAL S2TOTAL 3180    1.1  1.30  1.20 10.0
## 102      1 2012 SOUTH     2     TOTAL S2TOTAL 3191    1.4  1.05  1.00 10.0
## 103      1 2012 SOUTH     2     TOTAL S2TOTAL 3192   1.05  1.55  0.90 10.0
## 104      1 2012 SOUTH     2     TOTAL S2TOTAL 3193   1.18  1.20  1.00  7.0
## 105      1 2012 SOUTH     2     TOTAL S2TOTAL 3194    1.4  1.30  1.85 13.0
## 106      1 2012 SOUTH     2     TOTAL S2TOTAL 3195   1.37  2.67  2.19 19.0
## 107      1 2012 SOUTH     2     TOTAL S2TOTAL 3196   1.32  2.15  1.55 11.0
## 108      1 2012 SOUTH     2      MEGA  S2MEGA  182   1.55  2.20  1.20 20.0
## 109      1 2012 SOUTH     2      MEGA  S2MEGA  183    1.3  1.80  0.90  8.0
## 110      1 2012 SOUTH     2      MEGA  S2MEGA  184   1.24  1.20  1.20 25.0
## 111      1 2012 SOUTH     2      MEGA  S2MEGA  185    1.5  2.10  1.75 16.0
## 112      1 2012 SOUTH     2      MEGA  S2MEGA  186   1.65  2.50  2.20 15.0
## 113      1 2012 SOUTH     2      MEGA  S2MEGA  187   2.17  2.00  1.20 15.0
## 114      1 2012 SOUTH     2      MEGA  S2MEGA  188   1.28  1.60  1.50 10.0
## 115      1 2012 SOUTH     2      MEGA  S2MEGA  189   1.07  1.50  1.50 10.0
## 116      1 2012 SOUTH     2      MEGA  S2MEGA  190   0.67  1.00  0.80  8.0
## 117      1 2012 SOUTH     2      MEGA  S2MEGA  191   0.68  0.70  0.60  4.0
## 118      1 2012 SOUTH     2      MEGA  S2MEGA  192   1.87  1.60  1.40  9.0
```

```
## 119      1 2012 SOUTH     2      MEGA  S2MEGA  193   1.35  1.90  1.50 14.0
## 120      1 2012 SOUTH     2      MEGA  S2MEGA  194   1.75  2.10  2.10 15.0
## 121      1 2012 SOUTH     2      MESO  S2MESO  462   1.75  3.30  2.50 23.0
## 122      1 2012 SOUTH     2      MESO  S2MESO  463   1.64  2.30  2.00 14.0
## 123      1 2012 SOUTH     2      MESO  S2MESO 2138   1.42  0.90  0.80 10.0
## 124      1 2012 SOUTH     3      OPEN  S3OPEN 1301   dead    NA    NA   NA
## 125      1 2012 SOUTH     3      OPEN  S3OPEN 1302    0.9  1.30  1.10 11.0
## 126      1 2012 SOUTH     3     TOTAL S3TOTAL 1061   dead    NA    NA   NA
## 127      1 2012 SOUTH     3     TOTAL S3TOTAL 1062    1.8  2.60  2.60 15.0
## 128      1 2012 SOUTH     3     TOTAL S3TOTAL 1063   2.47  3.10  2.20 18.0
## 129      1 2012 SOUTH     3     TOTAL S3TOTAL 1064   2.15  1.60  1.10 17.0
## 130      1 2012 SOUTH     3     TOTAL S3TOTAL 1066    1.7  2.50  2.15 15.0
## 131      1 2012 SOUTH     3     TOTAL S3TOTAL 1066    1.9  1.80  1.50 20.0
## 132      1 2012 SOUTH     3     TOTAL S3TOTAL 1067   1.95  2.10  1.90 13.0
## 133      1 2012 SOUTH     3     TOTAL S3TOTAL 1068    1.8  1.70  1.40 13.0
## 134      1 2012 SOUTH     3     TOTAL S3TOTAL 1069    1.4  2.00  1.60 14.0
## 135      1 2012 SOUTH     3     TOTAL S3TOTAL 1070      1  1.30  1.20  7.0
## 136      1 2012 SOUTH     3     TOTAL S3TOTAL 2139   1.75  1.20  1.10 13.0
## 137      1 2012 SOUTH     3     TOTAL S3TOTAL 2140   1.28  1.50  0.95  4.0
## 138      1 2012 SOUTH     3     TOTAL S3TOTAL 2151      1  1.40  1.20  4.0
## 139      1 2012 SOUTH     3     TOTAL S3TOTAL 2152   1.45  1.50  1.30 10.0
## 140      1 2012 SOUTH     3     TOTAL S3TOTAL 2153      1  1.00  0.75  8.0
## 141      1 2012 SOUTH     3     TOTAL S3TOTAL 2154   1.03  1.00  0.90  6.0
## 142      1 2012 SOUTH     3     TOTAL S3TOTAL 2155   1.51  2.00  1.80 12.0
## 143      1 2012 SOUTH     3     TOTAL S3TOTAL 2156   1.17  1.10  0.90 10.0
## 144      1 2012 SOUTH     3     TOTAL S3TOTAL 2157   1.33  1.90  1.85 14.0
## 145      1 2012 SOUTH     3     TOTAL S3TOTAL 2158    1.3  1.10  0.85  8.0
## 146      1 2012 SOUTH     3     TOTAL S3TOTAL 2159   1.13  1.10  0.90 10.0
## 147      1 2012 SOUTH     3     TOTAL S3TOTAL 2160   1.58  1.40  1.40 13.0
## 148      1 2012 SOUTH     3     TOTAL S3TOTAL 2171   1.06  1.40  1.00  5.0
## 149      1 2012 SOUTH     3     TOTAL S3TOTAL 2172   1.05  1.40  0.95  7.0
## 150      1 2012 SOUTH     3     TOTAL S3TOTAL 2173   1.45  1.60  1.10  6.0
## 151      1 2012 SOUTH     3     TOTAL S3TOTAL 2174   1.15  1.10  0.90  5.0
## 152      1 2012 SOUTH     3     TOTAL S3TOTAL 2175   1.42  1.45  1.30 13.0
## 153      1 2012 SOUTH     3     TOTAL S3TOTAL 2176   1.02  1.20  1.00  8.0
## 154      1 2012 SOUTH     3     TOTAL S3TOTAL 2177    1.4  1.20  1.00  9.0
## 155      1 2012 SOUTH     3     TOTAL S3TOTAL 2178   1.45  2.10  2.05 15.0
## 156      1 2012 SOUTH     3      MESO  S3MESO 1421   1.95  2.20  1.60 13.0
## 157      1 2012 SOUTH     3      MESO  S3MESO 1422   dead    NA    NA   NA
##     FLOWERS BUDS FRUITS   ANT
## 1         0    0     10    CS
## 2         0    0    150    TP
## 3         2    1     50    TP
## 4         0    0     75    CS
## 5         0    0     20    CS
## 6         0    0      0     E
## 7         0    0      0    CS
## 8         0    0     25    CS
## 9         0    0      0    TP
## 10        0    0     50    TP
## 11        0    0      5    CS
## 12        0    0     60    TP
## 13        0    0     60    TP
## 14        2    0     60    CS
```

```
## 15        2    0      0    CS
## 16        0    0      0    TP
## 17        0    0      0    TP
## 18        0    0      0    CS
## 19        0    0      0    CM
## 20        0    0      0    TP
## 21       NA   NA     NA
## 22        0    0      5    CS
## 23        0    0     45    CS
## 24       40   50     35    CS
## 25        8    2     65    CS
## 26        0    0     20    TP
## 27        0    0     70    CS
## 28        0    0    125    CM
## 29        0    0    200    CM
## 30        0    0     10    CS
## 31        0    0      0    CS
## 32        0    0     35    TP
## 33        0    0    300    CM
## 34        2    2    100    CS
## 35        0    0     30    CM
## 36        0    0     50    TP
## 37        0    0     10    CM
## 38        0    0     25    CS
## 39        0    0     15    TP
## 40        0    0      0    TP
## 41        0    0     15    TP
## 42        0    0      0    TP
## 43        0    0     40    TP
## 44        0    0      0    TP
## 45        0    0     15    CM
## 46        0    0      0    CM
## 47        0    0      0    TP
## 48        0    0      0    TP
## 49        0    0      1    TP
## 50        0    0     20    TP
## 51        0    0      0    TP
## 52        0    0      0    TP
## 53        0    0     20    TP
## 54        0    0      0    TP
## 55        0    0      0    CN
## 56        0    0      0    CN
## 57        0    0      0    TP
## 58        0    0      5    TP
## 59        0    0      0    TP
## 60        0    0     25    TP
## 61        0    0     25    TP
## 62        0    0     20    TP
## 63        0    0      0    TP
## 64        0    0     10    CS
## 65        1    0     25    CS
## 66        0    0      0    TP
## 67        0    0     10    TP
## 68        0    0      0    TP
```

```
## 69        0    0      0    TP
## 70        0    0      0    TP
## 71        0    0      0    TP
## 72        0    0      0    CS
## 73        0    0      0    CS
## 74        0    0     25 AB_TP
## 75        0    0      0    TP
## 76        0    0      0    TP
## 77        0    0      0    TP
## 78        0    0      0    CS
## 79        0    0      0    CS
## 80        0    0      0    CS
## 81        0    0      0    CS
## 82        0    0      5    CS
## 83        6    0      0    CS
## 84        0    0      0    CS
## 85        0    0      1    CS
## 86        0    0     25    CS
## 87        0    0      0    CS
## 88        0    0      0    CS
## 89        0    0     10    CS
## 90        0    0      0    CS
## 91        0    0     35    CS
## 92        0    0      0    CS
## 93        0    0      0    CS
## 94        0    0      0    CS
## 95        0    0      0    CS
## 96        0    0     20    CS
## 97        0    0      0    CS
## 98        0    0      0    CM
## 99        0    0    100    CM
## 100       0    0      0    CS
## 101       0    0      0    CS
## 102       0    0      0    CS
## 103       0    0      0    CM
## 104       0    0      0    TP
## 105       0    0     30    CS
## 106       0    0     50    TP
## 107       0    0     10    CS
## 108       0    0      0    CS
## 109       0    0     15    CS
## 110       0    0     10    CS
## 111       5    0    200    CS
## 112       0    0     80    CS
## 113       0    0    150    TP
## 114       0    0     40    TP
## 115       0    0     60    TP
## 116       0    0      0    CS
## 117       0    0      0    TP
## 118       0    0     40    CS
## 119       0    0     20    CS
## 120       0    0     75    TP
## 121       0    0     20    CM
## 122       0    0      0    TP
```

```
## 123       0    0      0     E
## 124      NA   NA     NA
## 125       0    0      0    TP
## 126      NA   NA     NA
## 127       0    0     50    TP
## 128       0    0      0    TP
## 129       0    0      0    TP
## 130       0    0      2    TP
## 131       0    0     25    TP
## 132       0    0      0    TP
## 133       0    0      0    TP
## 134       0    0      0    TP
## 135       0    0      0    TP
## 136       0    0      0    TP
## 137       0    0      0    TP
## 138       0    0      0    TP
## 139       0    0      0    TP
## 140       0    0      0    TP
## 141       0    0      0    TP
## 142       0    0      0    TP
## 143       0    0      0    TP
## 144       0    0      0    TP
## 145       0    0      0    TP
## 146       0    0      0    TP
## 147       0    0      0    TP
## 148       0    0      8    TP
## 149       0    0      0    TP
## 150       0    0      0    TP
## 151       0    0      0    TP
## 152       0    0      0    TP
## 153       0    0      0    TP
## 154       0    0      0    TP
## 155       0    0     20    TP
## 156       0    0      2    CS
## 157      NA   NA     NA
```

```
report_rsquared(UHURU_data,"CM","AXIS1~CIRC")
```

```
##   species    r2
## 1      CM 0.866
```

##3. Modify the function so that it also determines if() the rsquared is significant based on a given threshold. The modified function should return() the species, rsquared and a significance value of "S" for a relationship with an rsquared > threshold or "NS" for an rsquared < threshold.

```
report_rsquared_modified <- function(data, species, formula, threshold){
  subset <- dplyr::filter(data, ANT == species)
  test <- lm(formula, data = subset)
  rsquared <- round(summary(test)$r.squared, 3)
   if (rsquared > threshold ){
     print("S")
} else if (rsquared < threshold)
   print("NS")
```

```
  output <- data.frame(species = species, r2 = rsquared)
  return(output)
}
```

##4. Execute your modified function for species of "CM", "CS", and "TP" given a threshold = 0.667.

```
report_rsquared_modified(UHURU_data,"CM","AXIS1~CIRC", 0.667)
```

```
## [1] "S"

##   species    r2
## 1      CM 0.866
```

```
report_rsquared_modified(UHURU_data,"CS","AXIS1~CIRC", 0.667)
```

```
## [1] "NS"

##   species    r2
## 1      CS 0.437
```

```
report_rsquared_modified(UHURU_data,"TP","AXIS1~CIRC", 0.667)
```

```
## [1] "S"

##   species    r2
## 1      TP 0.701
```

#Exercise 4: Multi-file Analysis DNA or RNA Iteration

Write a function, dna_or_rna(sequence), that determines if a sequence of base pairs is DNA, RNA, or if it is not possible to tell given the sequence provided. Since all the function will know about the material is the sequence the only way to tell the difference between DNA and RNA is that RNA has the base Uracil ("u") instead of the base Thymine ("t"). Have the function return one of three outputs: "DNA", "RNA", or "UNKNOWN".

##1. Use the function and a for loop to print the type of the sequences in the following list. ##2.Use the function and sapply to print the type of the sequences in the following list.

```
library(dplyr)
library(stringr)
sequences <- c("ttgaatgccttacaactgatcattacacaggcggcatgaagcaaaaatatactgtgaaccaatgcaggcg", "gauuauuccccaca

dna_or_rna <- function(sequences){
  if (str_detect(sequences, "u")) {
 result <-  print("RNA")
   } else if (str_detect(sequences, "t")) {
   result <- print("DNA")
   } else {
 result <- print("UNKNOWN")
  }
     return(result)
}
sapply(sequences, dna_or_rna)
```

```
## [1] "DNA"
## [1] "RNA"
## [1] "UNKNOWN"
## [1] "RNA"
## [1] "RNA"
```

```
##                                 ttgaatgccttacaactgatcattacacaggcggcatgaagcaaaaatatactgtgaaccaatgcaggcg
##                                                                                                   "DNA"
##                                 gauuauuccccacaaagggaguggguauuaggagcugcaucauuuacaagagcagaauguuucaaaugcau
##                                                                                                   "RNA"
##                                                                  gaaagcaagaaaaggcaggcgaggaagggaagaagggggggaaacc
##                                                                                                "UNKNOWN"
## guuuccuacaguauuugaugagaaugagaguuuacuccuggaagauaauauuuagaauguuuacaacugcaccugaucagguggauaaggaagaugaagacu
##                                                                                                   "RNA"
##                       gauaaggaagaugaagacuuucaggaaucuaauaaaaugcacuccaugaauggauucauguauggggaaucagccgggu
##                                                                                                   "RNA"
```

#Exercise 5:Energy Conversion Challenge

1. Write a function with the form convert_energy_units(energy_value, input_unit, output_unit) to convert units between the following energy values:

Joules(J), Kilojoules(KJ), Calories(CAL), and Kilocalories (KCAL; this unit is used for labeling the amount of energy contained in food). To write the equations to convert between units, consider the following:

A Kilojoule is 1000 Joules, a Calorie is 4.1868 Joules, a Kilocalorie is 4186.8 Joules.

```
convert_energy_units <- function(energy_value, input_unit, output_unit){
   if (str_detect(input_unit, "KJ")) {
     J <- energy_value * 1000
   } else if (str_detect(input_unit, "CAL")) {
     J <- energy_value * 4.1868
   } else {
      (str_detect(input_unit, "KCAL"))
     J <- energy_value * 4186.8
  }
     return(J)
}
```

An example of a call to this function would look like:

energy_in_cal <- 200 energy_in_j <- convert_energy_units(energy_in_cal, "CAL", "J")

2. Test your function by running the example call above.

```
energy_in_cal <- 200
energy_in_j <- convert_energy_units(energy_in_cal, "CAL", "J")
energy_in_j
```

```
## [1] 837.36
```

3. If either the input unit or the output unit do not match the four types given above, have the function print - "Sorry, I don't know how to convert " + the name of the unit provided.

```
convert_energy_units <- function(energy_value, input_unit, output_unit){
   if (str_detect(input_unit, "KJ")) {
     J <- energy_value * 1000
   } else if (input_unit == "CAL") {
     J <- energy_value * 4.1868
     print("CAL")
   } else if (input_unit == "KCAL") {
     J <- energy_value * 4186.8
     print("this should be KCAL")
   } else {
   message("Sorry, I don't know how to convert", input_unit)
     return(NA)
   }
     return(J)
}
```

```
convert_to_J <- function(energy_value, input_unit){
   if (str_detect(input_unit, "KJ")) {
     J <- energy_value * 1000
   } else if (input_unit == "CAL") {
     J <- energy_value * 4.1868
     message("input units are CAL")
   } else if (input_unit == "KCAL") {
     J <- energy_value * 4186.8
     message("input units are KCAL")
   } else {
   message("Sorry, I don't know how to convert", input_unit)
    # return(NA)
     J <- NA
   }
     return(J)
}
```

4. Use your function to answer the following questions:

a) What is the daily metabolic energy used by a human (~2500 KCALs) in Joules.

```
convert_energy_units(2500, "CAL", "J")
```

```
## [1] "CAL"
```

```
## [1] 10467
```

```
convert_energy_units(2500, "KCAL", "J")
```

```
## [1] "this should be KCAL"
```

```
## [1] 10467000
```

b) How many times more energy does a common seal use than a human? The common seal uses ~52,500 KJ/day (Nagy et al. 1999). Use the daily human metabolic cost calculated in (4a).

```
convert_energy_units(52500, "KCAL", "J")
```

```
## [1] "this should be KCAL"
```

```
## [1] 219807000
```

c) How many ergs (ERG) are there in one kilocalorie. Since we didn't include the erg conversion this should trigger our 'don't know how to convert' message.

```
convert_energy_units(1, "ERG", "J")
```

```
## Sorry, I don't know how to convertERG
```

```
## [1] NA
```

5. Make the function more efficient, and instead of writing an individual conversion between each of the different units (which requires 12 if statements) you could choose to convert all of the input units to a common scale and then convert from that common scale to the output units. This approach is especially useful if you need to add new units later.

```
convert_energy_efficient <- function(energy_value, input_unit, output_unit){
  #convert input to J
 energy_to_J <- convert_energy_units(energy_value, input_unit)
  #convert J to output unit
  message("output unit is", output_unit)
    if (str_detect(output_unit, "KJ")) {
     energy <- energy_to_J / 1000
   } else if (output_unit == "CAL") {
     energy <- energy_to_J / 4.1868
     print("CAL")
   } else if (output_unit == "KCAL") {
     energy <- energy_to_J / 4186.8
     print(str_c("this should be KCAL"))
   } else {
     message("Sorry, I don't know how to convert", input_unit)
     print("sorry again")
     return(NA)
   }

  return(energy)
}
```

```
convert_energy_efficient(200, "Azul","End of the semester!" )
```

```
## Sorry, I don't know how to convertAzul
```

```
## output unit isEnd of the semester!
```

```
## Sorry, I don't know how to convertAzul
```

```
## [1] "sorry again"
```

```
## [1] NA
```

```
convert_energy_efficient(1200, "CAL",output_unit = "KCAL" )
```

```
## [1] "CAL"
```

```
## output unit isKCAL
```

```
## [1] "this should be KCAL"
```

```
## [1] 1.2
```

#thank you for your patience luna i loved having you a an instructor truly have seen progress in myself can't wait for you to become a professor. Share the news when you do!!!! <3