

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
```

```
## [13] FALSE 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 argument

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

```
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 name:
    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://ndownloads.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	Poocetes	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
```

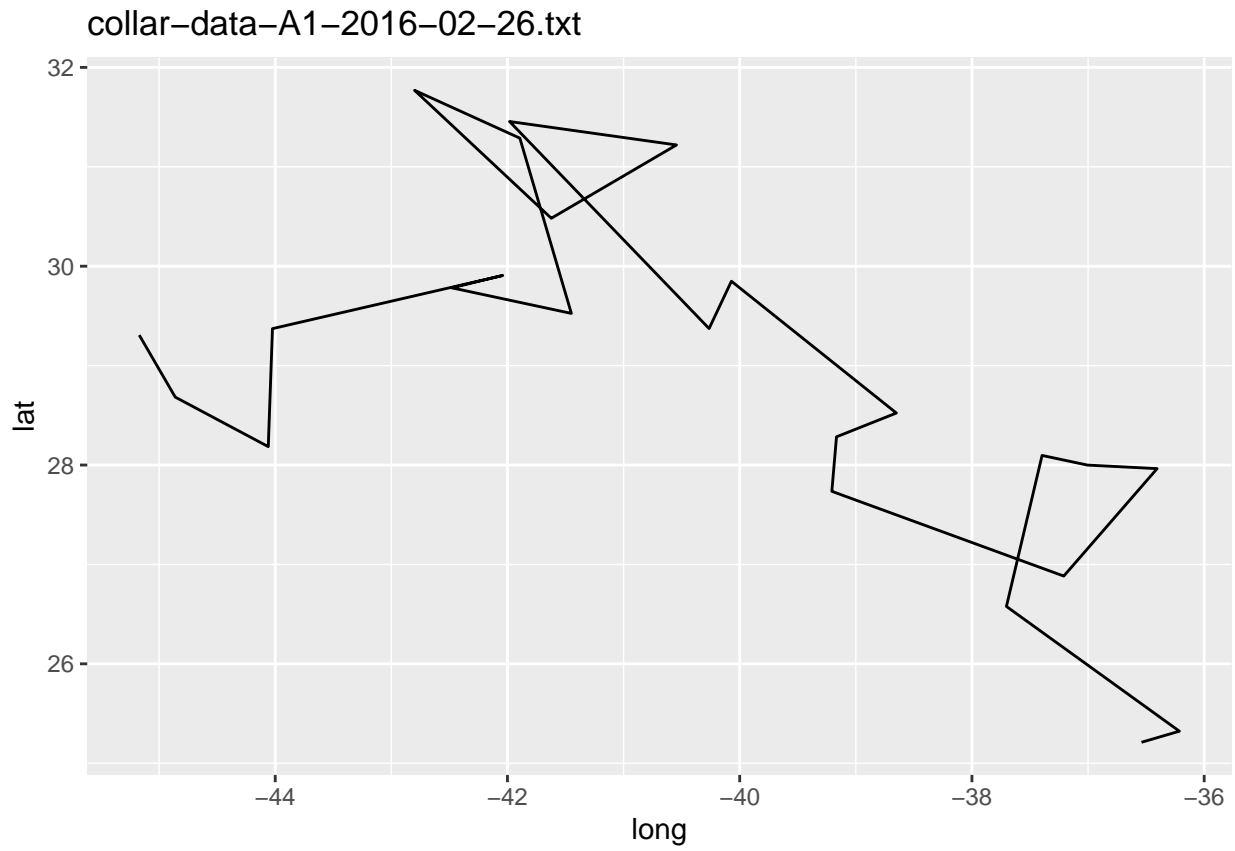
```
library(ggplot2)  
collar_data_files <- list.files(pattern = "collar-")  
collar_data_files
```

```
## [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"
```

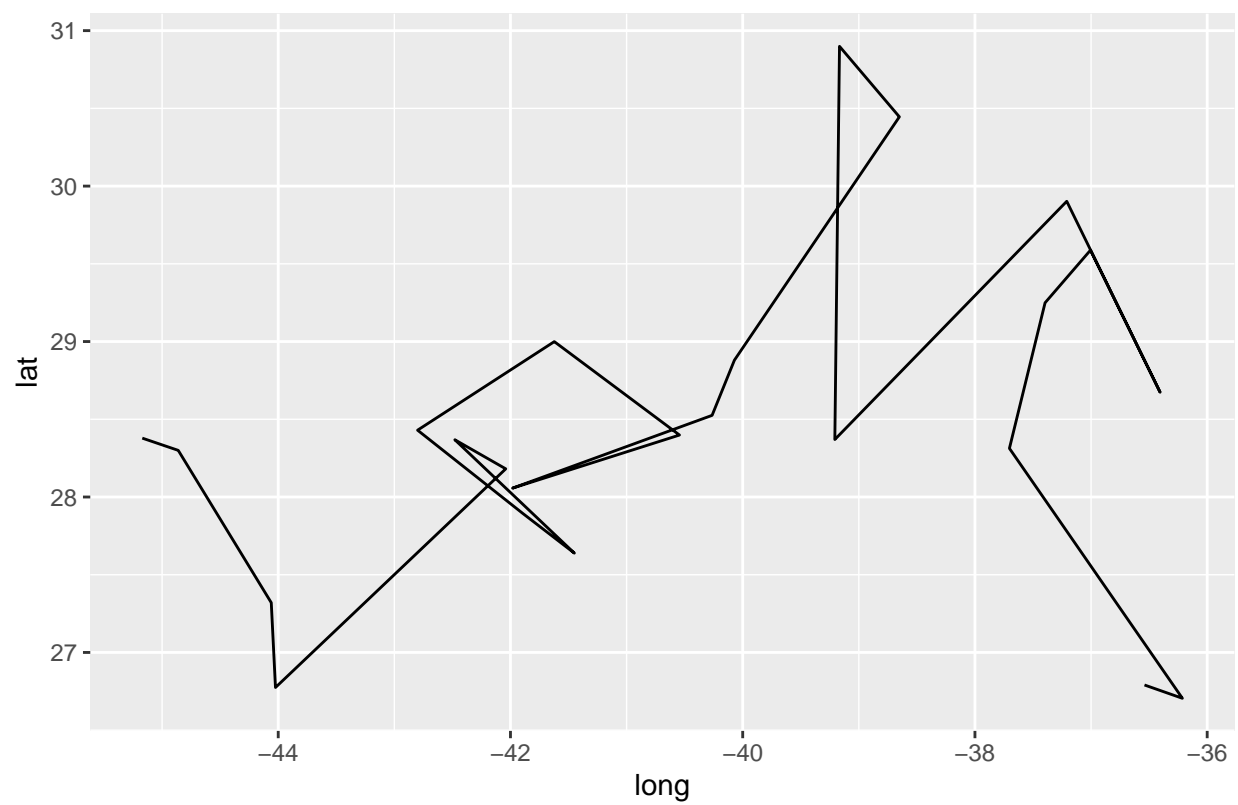
##4. Use a loop to load each of these files into R and make a line plot (using `geom_path()`) for each file with `long` on the x axis and `lat` on the y axis. Graphs, like other types of output, won't display inside a loop unless you explicitly display them, so you need put your `ggplot()` command inside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

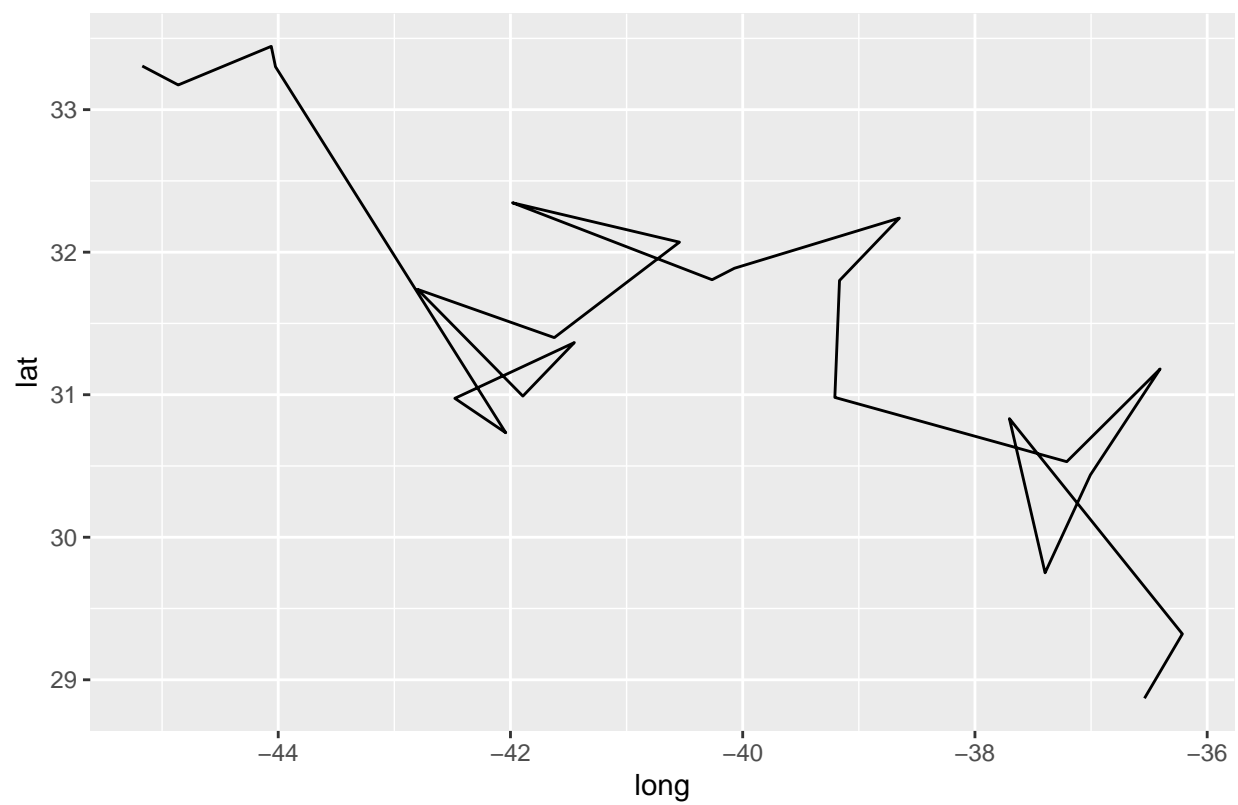
```
min_results <- vector(mode = "integer", length(collar_data_files))  
max_results <- vector(mode = "integer", length(collar_data_files))  
  
for(i in 1:length(collar_data_files)){  
  filename <- collar_data_files[i]  
  data <- read.csv(filename)  
  data %>%  
    ggplot(aes(x = long, y = lat ))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



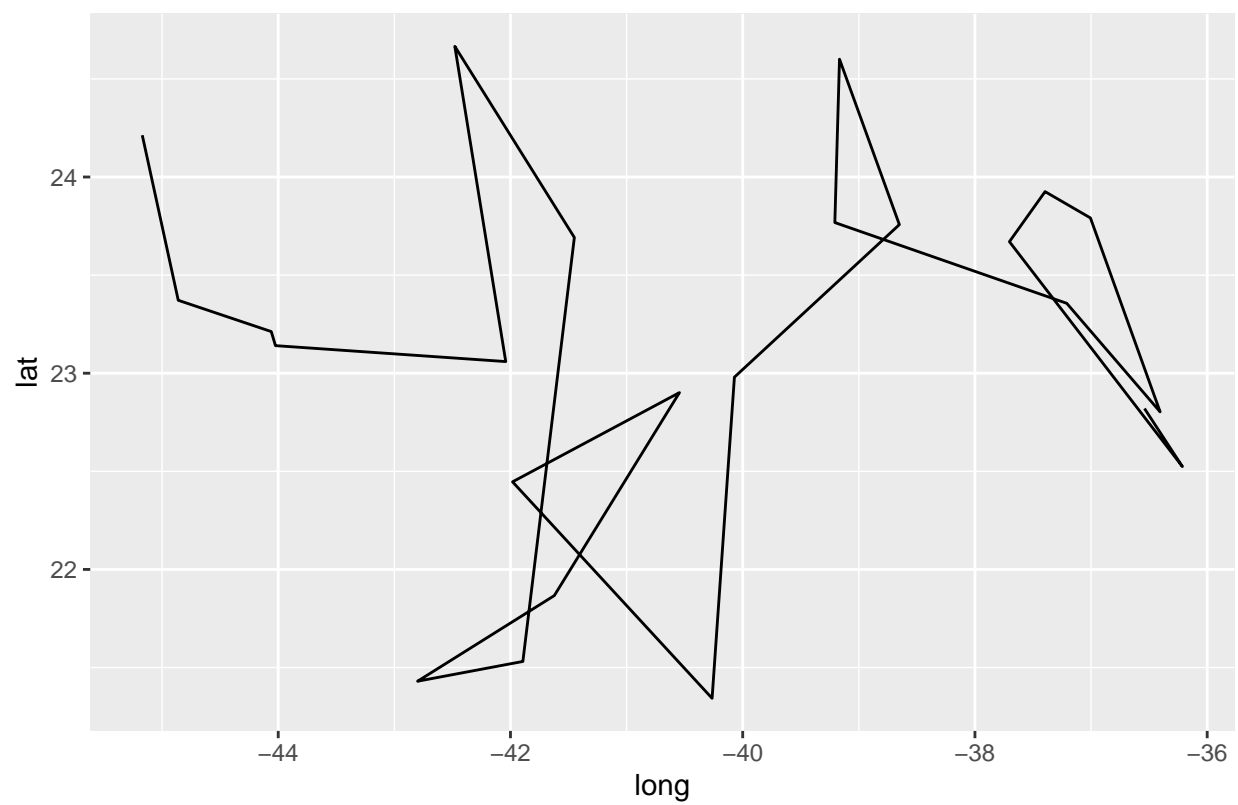
collar-data-B2-2016-02-26.txt



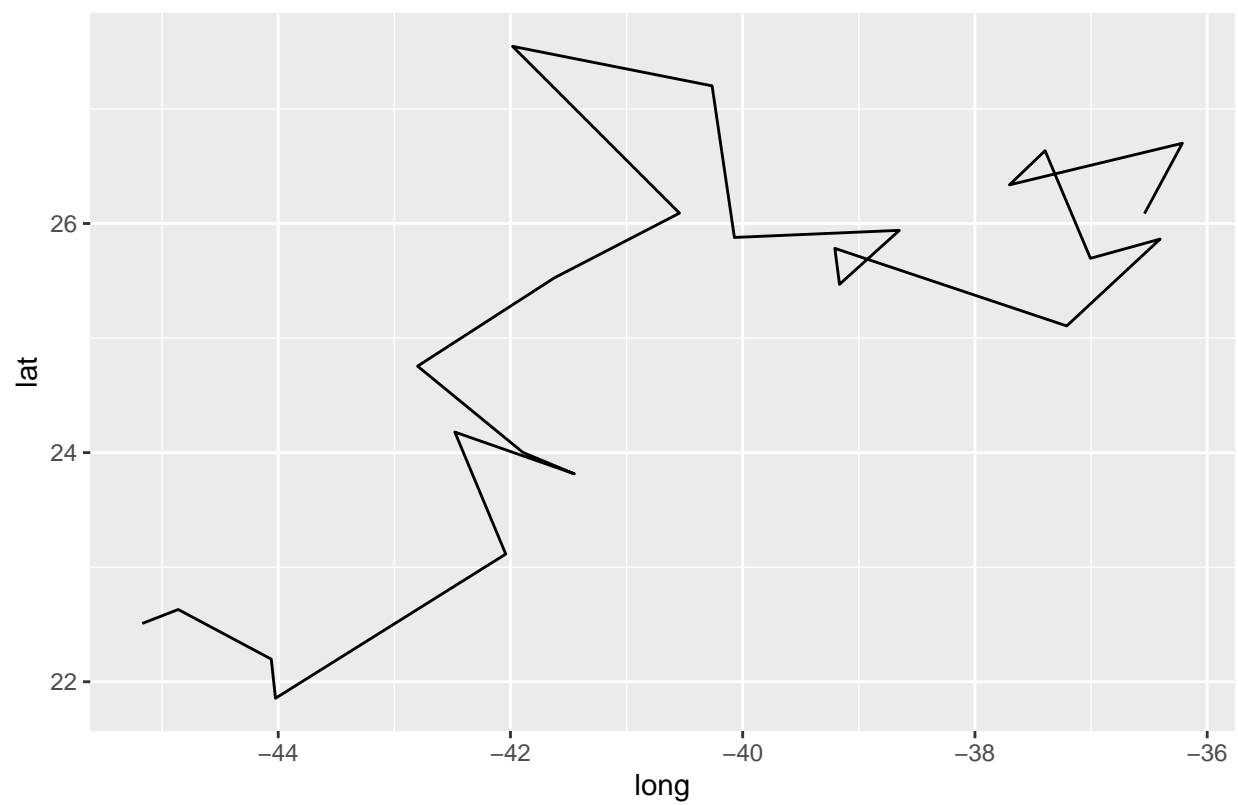
collar-data-C3-2016-02-26.txt



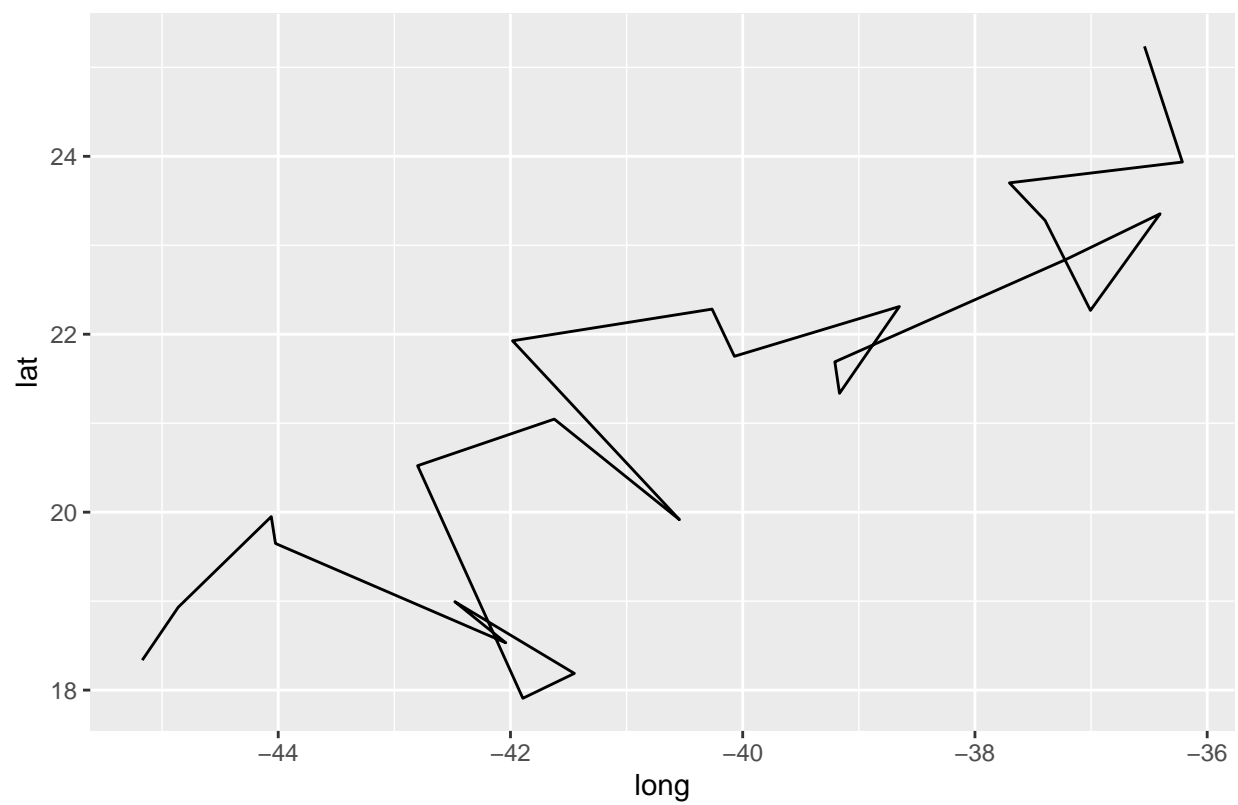
collar-data-D4-2016-02-26.txt



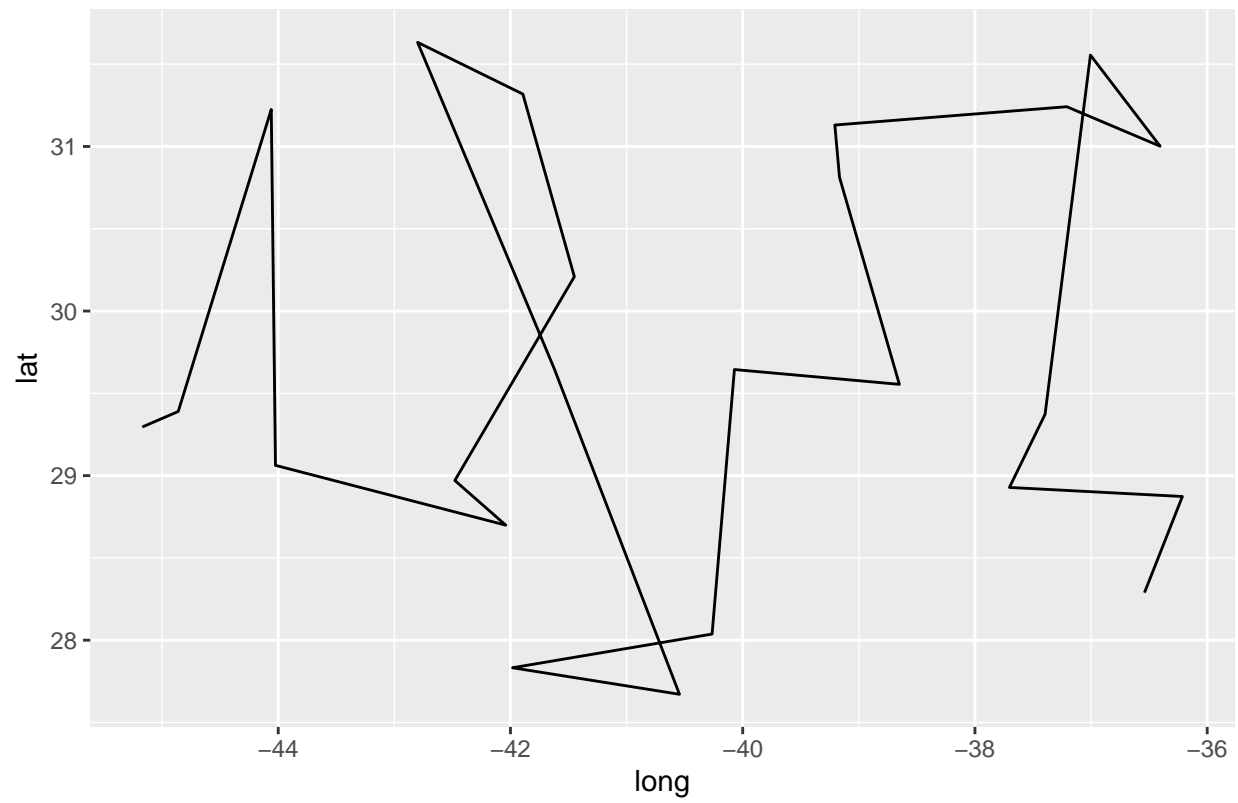
collar-data-E5-2016-02-26.txt



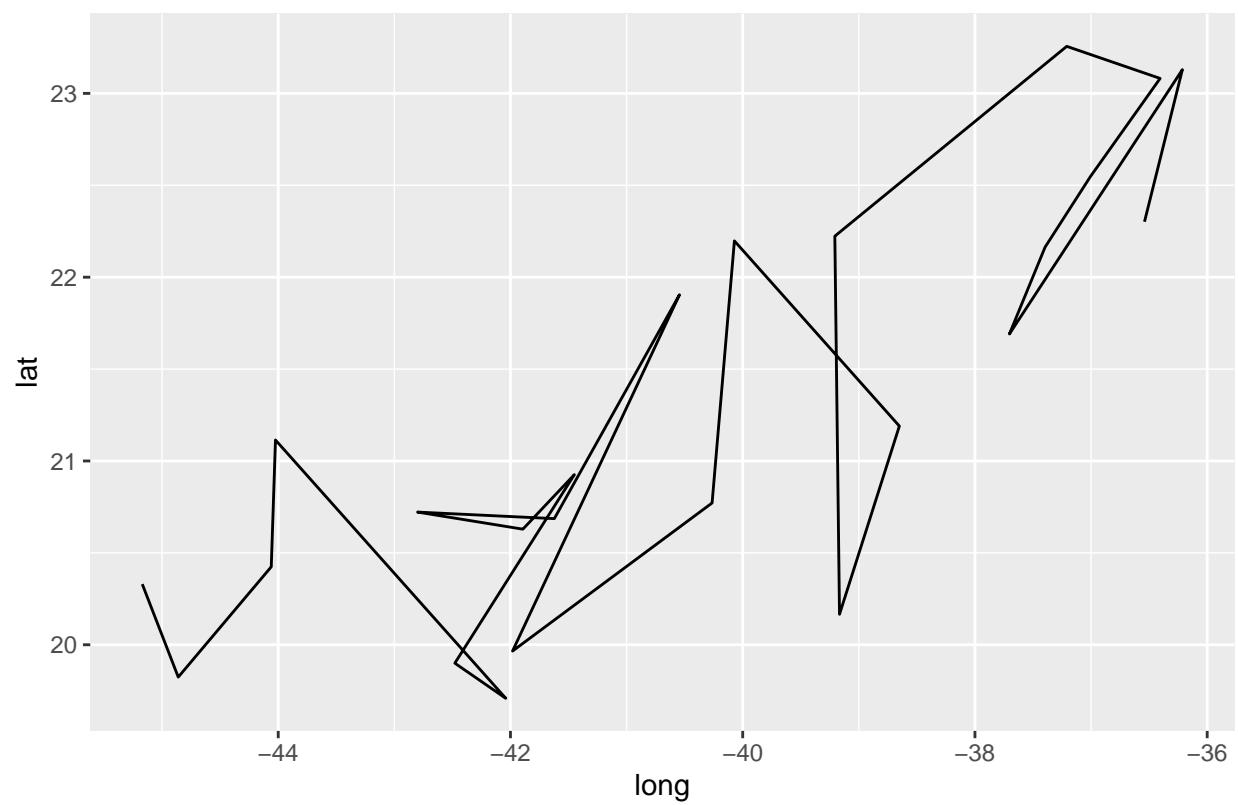
collar-data-F6-2016-02-26.txt

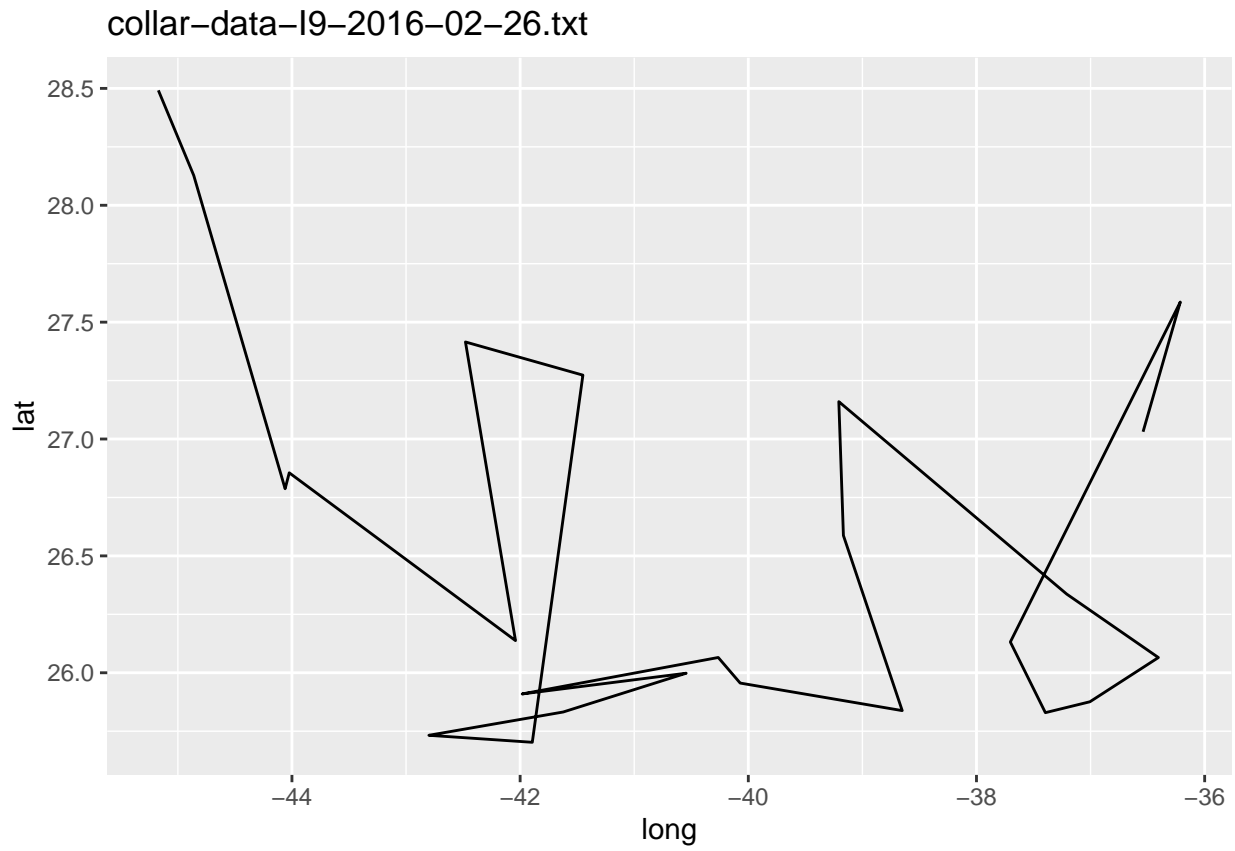


collar-data-G7-2016-02-26.txt

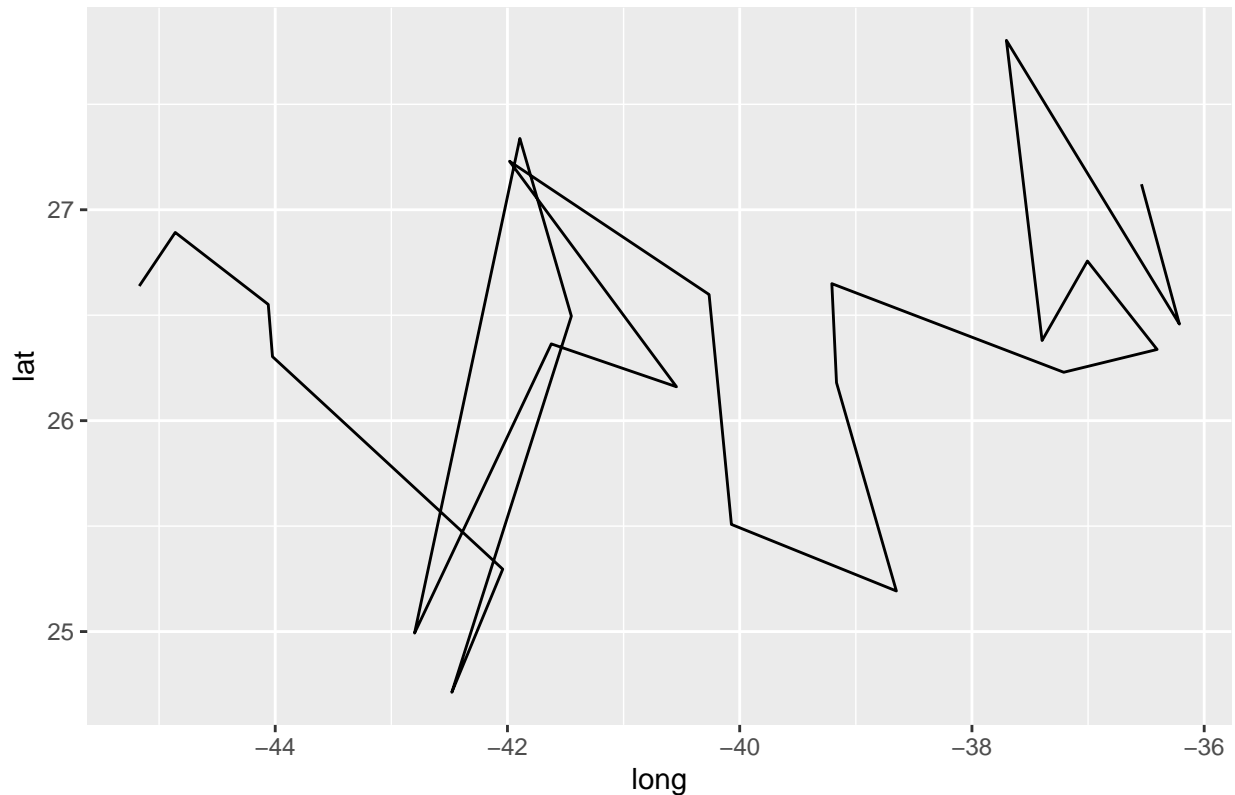


collar-data-H8-2016-02-26.txt





collar-data-J10-2016-02-26.txt



#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)
  test <- lm(formula, data = subset)
  rsquared <- round(summary(test)$r.squared, 3)
  output <- data.frame(species = species, r2 = rsquared)
  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")
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){
  subset <- dplyr::filter(data, ANT == species)
  test <- lm(formula, data = subset)
  rsquared <- round(summary(test)$r.squared, 3)
  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.