

# Week12\_Assignment

Ellen Bledsoe

2024-04-21

## Assignment

### Purpose

The goal of this assignment is to practice writing and using for loops for iteration.

### Task

Write R code to successfully answer each question below.

### Criteria for Success

- Code is within the provided code chunks or new code chunks are created where necessary
- Code chunks run without errors
- Code chunks have brief comments indicating which code is answering which part of the question
- Code will be assessed as follows:
  - Produces the correct answer using the requested approach: 100%
  - Generally uses the right approach, but a minor mistake results in an incorrect answer: 90%
  - Attempts to solve the problem and makes some progress using the core concept, but returns the wrong answer and does not demonstrate comfort with the core concept: 50%
  - Answer demonstrates a lack of understanding of the core concept: 0%
- Any questions requiring written answers are answered with sufficient detail

### Due Date

April 15 at midnight MST

## Assignment Exercises

### Set Up

Be sure to load the `tidyverse` to use later in the assignment.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr    1.5.1
## v ggplot2    3.5.0      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

## 1. For Loop Basics (30 pts)

Complete the following tasks.

I recommend copying the commented code and making changes to the copy. That way, if you have made edits that you haven't kept track of, you have the original code to start over as needed.

- The code below prints the numbers 1 through 5 one line at a time. Modify it to print each of these numbers multiplied by 3.

```
numbers <- c(1, 2, 3, 4, 5)
# for (number in numbers){
#   print(number)
# }

for (number in numbers){
  multiplied <- number*3
  print(multiplied)
}
```

```
## [1] 3
## [1] 6
## [1] 9
## [1] 12
## [1] 15
```

- Write a for loop that loops over the following vector and prints out the mass in kilograms (`mass_kg = 2.2 * mass_lb`)

```
mass_lbs <- c(2.2, 3.5, 9.6, 1.2)

for (mass in mass_lbs){
  mass_kg <- 2.2 * mass
  print(mass_kg)
}
```

```
## [1] 4.84
## [1] 7.7
## [1] 21.12
## [1] 2.64
```

c. Complete the code below so that it prints out the name of each bird one line at a time.

```
# birds = c('robin', 'woodpecker', 'blue jay', 'sparrow')
# for (i in 1:length(____)){
#   print(birds[____])
# }

birds = c('robin', 'woodpecker', 'blue jay', 'sparrow')
for (i in 1:length(birds)){
  print(birds[i])
}
```

```
## [1] "robin"
## [1] "woodpecker"
## [1] "blue jay"
## [1] "sparrow"
```

d. Complete the code below so that it stores one area for each radius.

```
#radius <- c(1.3, 2.1, 3.5)
# areas <- vector(_____ = "numeric", length = _____)
# for (__ in 1:length(_____)){
#   areas[____] <- pi * radius[i] ^ 2
# }
#areas

radius <- c(1.3, 2.1, 3.5)
areas <- vector(mode = "numeric", length = length(radius))
for (i in 1:length(radius)){
  areas[i] <- pi * radius[i] ^ 2
}
areas
```

```
## [1] 5.309292 13.854424 38.484510
```

e. Complete the code below to calculate an area for each pair of **lengths** and **widths**, store the areas in a vector, and after they are all calculated print them out:

```
lengths = c(1.1, 2.2, 1.6)
widths = c(3.5, 2.4, 2.8)
# areas <- vector(length = _____)
# for (i in _____) {
#   areas[____] <- lengths[____] * widths[____]
# }

areas <- vector(length = length(lengths))
for (i in 1:length(lengths)) {
  areas[i] <- lengths[i] * widths[i]
}
areas
```

```
## [1] 3.85 5.28 4.48
```

## 2. Size Estimates by Name (30 pts)

This is a followup to “Size Estimates by Name” from last week.

Download the `dinosaur_lengths.csv` file from D2L and place it in the correct directory. Read the file into R.

```
dinosaur_lengths <- read_csv("/Users/graceadams/Documents/Git/Adams_DataWrangling/data_clean/dinosaur_l
```

```
## Rows: 500 Columns: 2
## -- Column specification -----
## Delimiter: ","
## chr (1): species
## dbl (1): lengths
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

Write a function `mass_from_length()` that uses the equation  $\text{mass} \leftarrow a * \text{length}^b$  to estimate the size of a dinosaur from its length. This function should take two arguments, `length` and `species`. For each of the following inputs for `species`, so use the associated `a` and `b` values to estimate the species mass using these equations:

- *Stegosauria*:  $\text{mass} = 10.95 * \text{length} ^ 2.64$  (Seebacher 2001).
- *Theropoda*:  $\text{mass} = 0.73 * \text{length} ^ 3.63$  (Seebacher 2001).
- *Sauropoda*:  $\text{mass} = 214.44 * \text{length} ^ 1.46$  (Seebacher 2001).
- For any other value of `species`:  $\text{mass} = 25.37 * \text{length} ^ 2.49$

```
mass_from_length <- function(length, species) {
  case_when(species == "Stegosauria" ~ mass <- 10.95 * length ^ 2.64,
            species == "Theropoda" ~ mass <- 0.73 * length ^ 3.63,
            species == "Sauropoda" ~ mass <- 214.44 * length ^ 1.46)
  return(mass)
}

mass_from_length
```

```
## function(length, species) {
##   case_when(species == "Stegosauria" ~ mass <- 10.95 * length ^ 2.64,
##             species == "Theropoda" ~ mass <- 0.73 * length ^ 3.63,
##             species == "Sauropoda" ~ mass <- 214.44 * length ^ 1.46)
##   return(mass)
## }
```

- Use this function and a for loop to calculate the estimated mass for each dinosaur, store the masses in a vector, and after all of the calculations are complete show the first few items in the vector using `head()`.
- Add the results in the vector back to the original data frame. Show the first few rows of the data frame using `head()`.
- Calculate the mean mass for each `species` using `dplyr` (no for loops).

### 3. Multi-file Analysis (40 pts)

You have satellite collars on a number of different individuals and want to be able to quickly look at all of their recent movements at once. The data is posted daily to a URL that contains one csv file for each individual: zip file

Start your solution by:

- If `individual-collar-data.zip` is not already in your working directory, download it from the link above and place the zip file in the correct directory.
- Unzip it using `unzip()`
- Obtain a list of all of the files with file names matching the pattern `"collar-data-*.txt"` (using `list.files()`)

```
download.file("http://www.datacarpentry.org/semester-biology/data/individual-collar-data.zip",
              "../Assignments/data_raw/individual-collar-data.zip")
```

```
unzip(zipfile = "../Assignments/data_raw/individual-collar-data.zip", exdir = "../Assignments/data_raw")
```

```
list.files("../Assignments/data_raw")
```

```
## [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"
## [11] "individual-collar-data.zip"      "locations-2016-01-01.txt"
## [13] "locations-2016-01-02.txt"       "locations-2016-01-03.txt"
## [15] "locations-2016-01-04.txt"       "locations-2016-01-05.txt"
## [17] "locations.zip"
```

```
data_files <- list.files(path = "../Assignments/data_raw",
                         pattern = "collar-")
```

- a. 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.

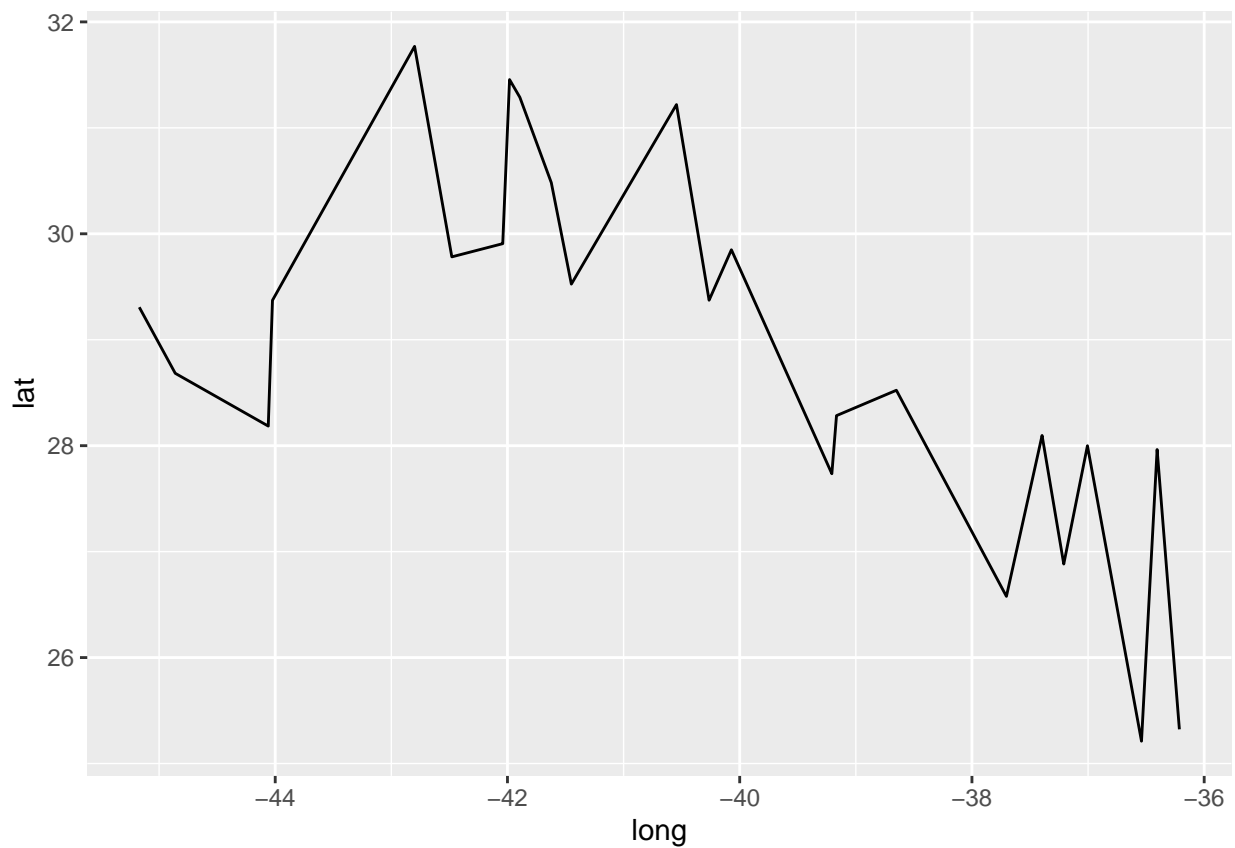
```
num_files <- length(data_files)
results <- vector(mode = "integer", length = num_files)
```

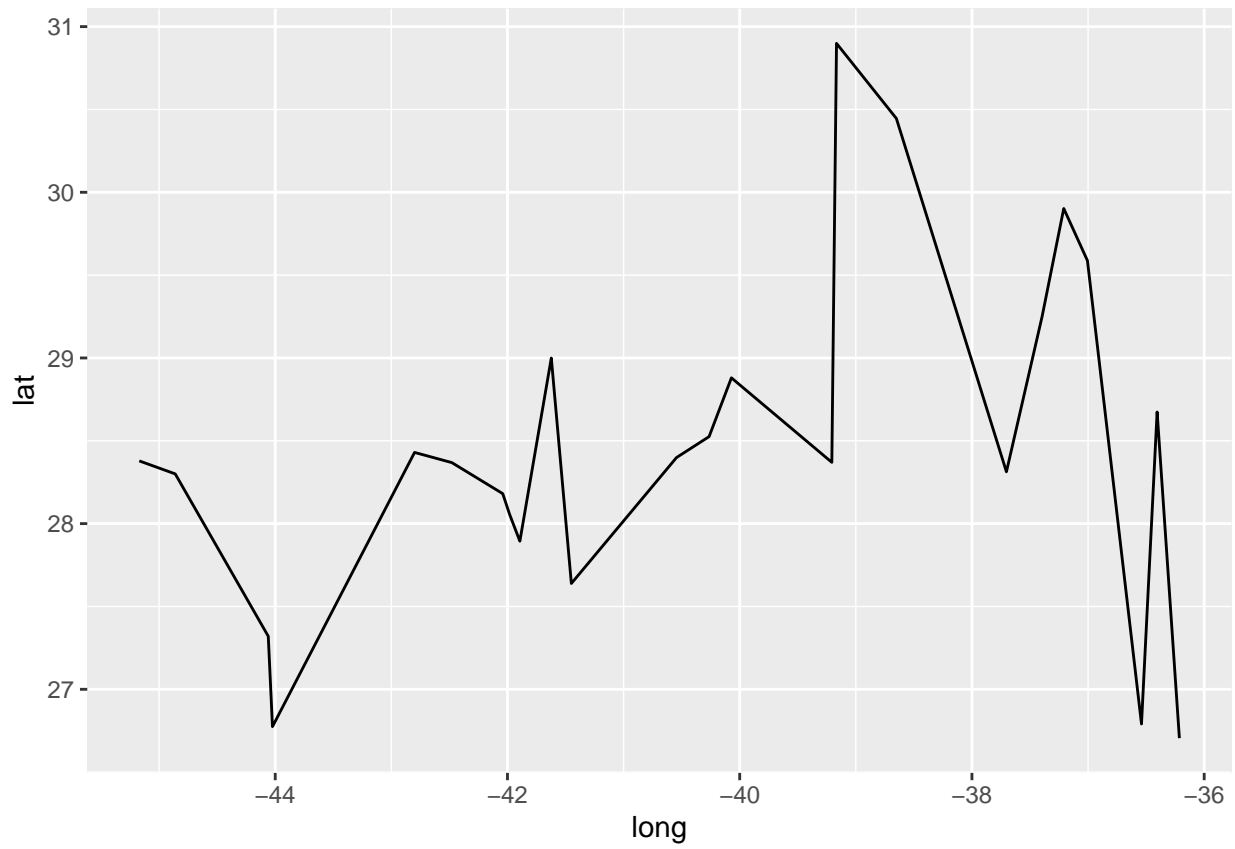
```
library(ggplot2)

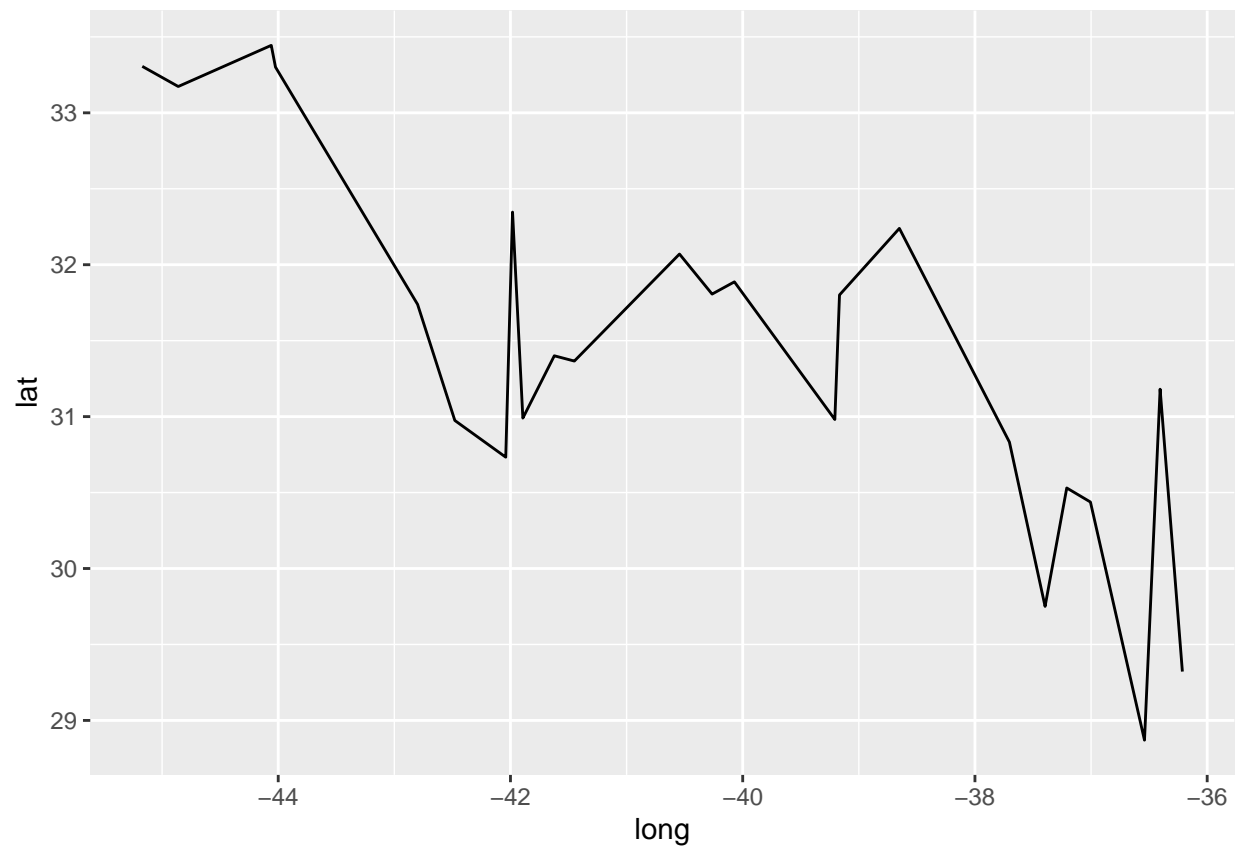
for (i in 1:num_files){
  # specify the file
  filename <- data_files[i]
  # read in the text file
  data <- read.csv(paste0("../Assignments/data_raw/", filename))
  # make a plot
  plot <- ggplot(data, aes(x= long, y = lat))+
    geom_line()
  plot
}
```

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.

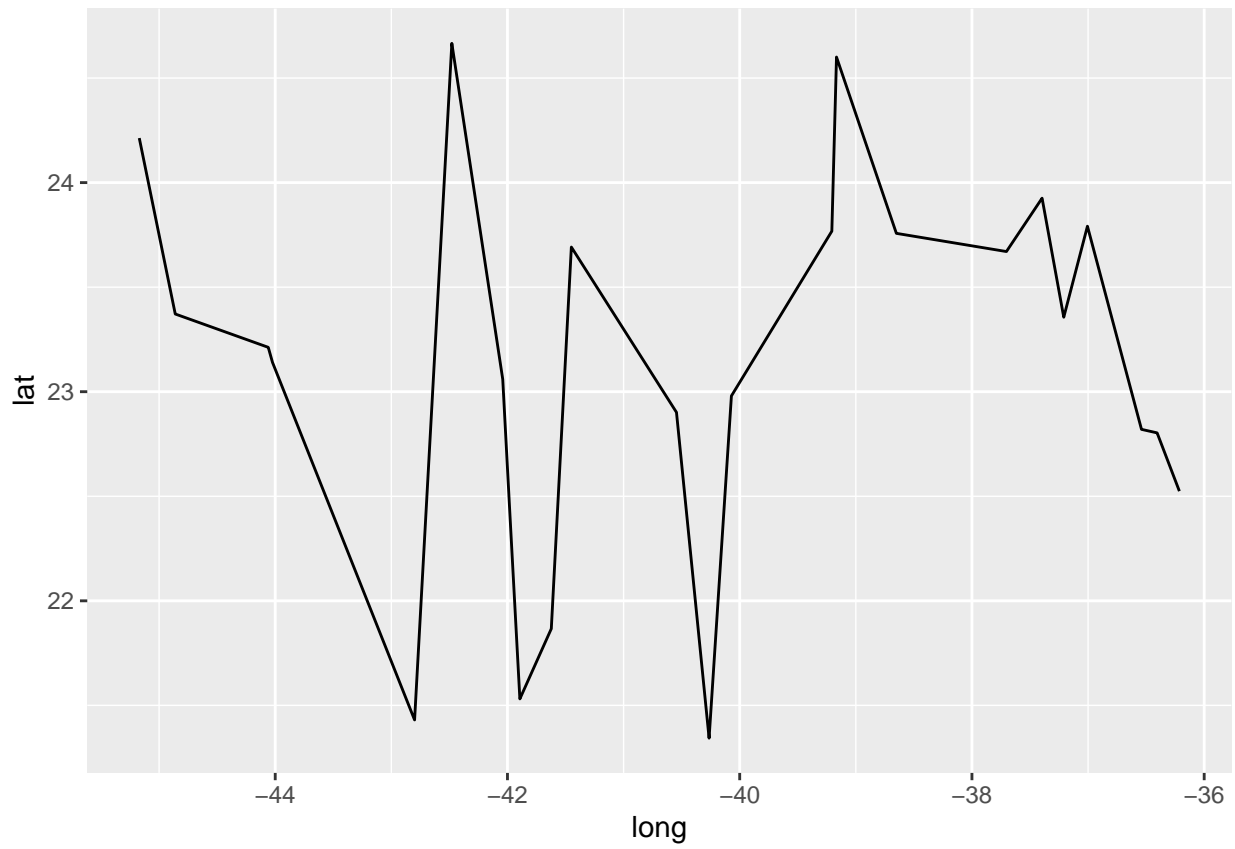
```
for (i in 1:num_files){  
  # specify the file  
  filename <- data_files[i]  
  # read in the text file  
  data <- read.csv(paste0("../Assignments/data_raw/", filename))  
  # make a plot  
  plot <- print(ggplot(data, aes(x= long, y = lat))+  
    geom_line())  
  plot  
}
```

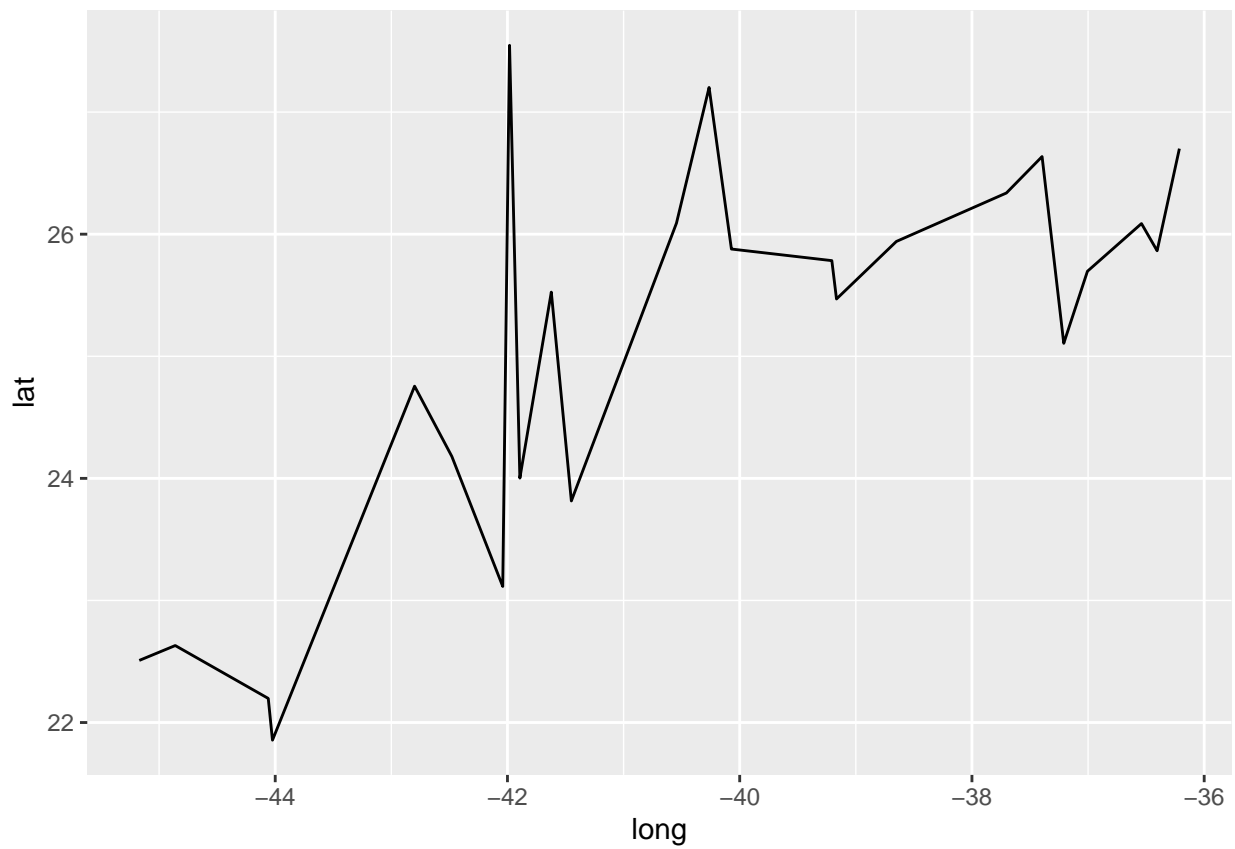


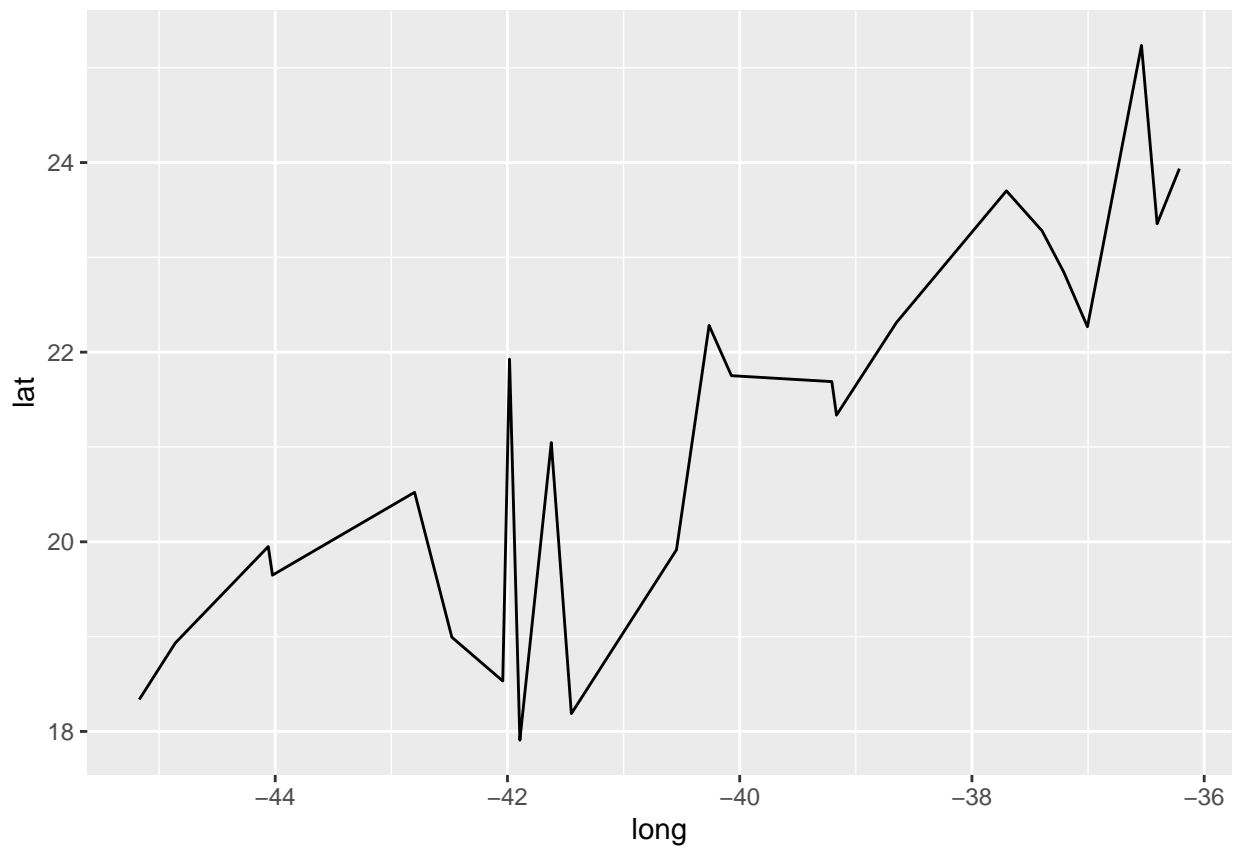


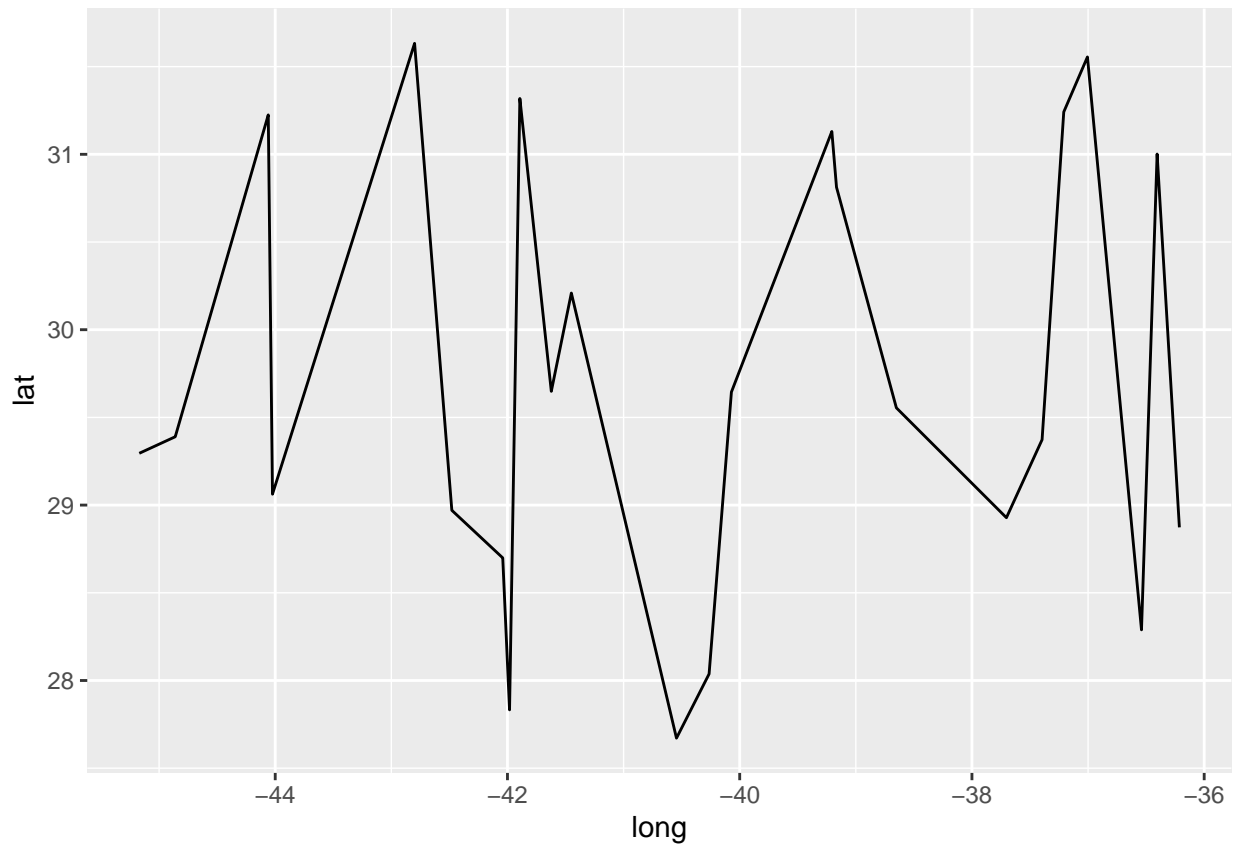


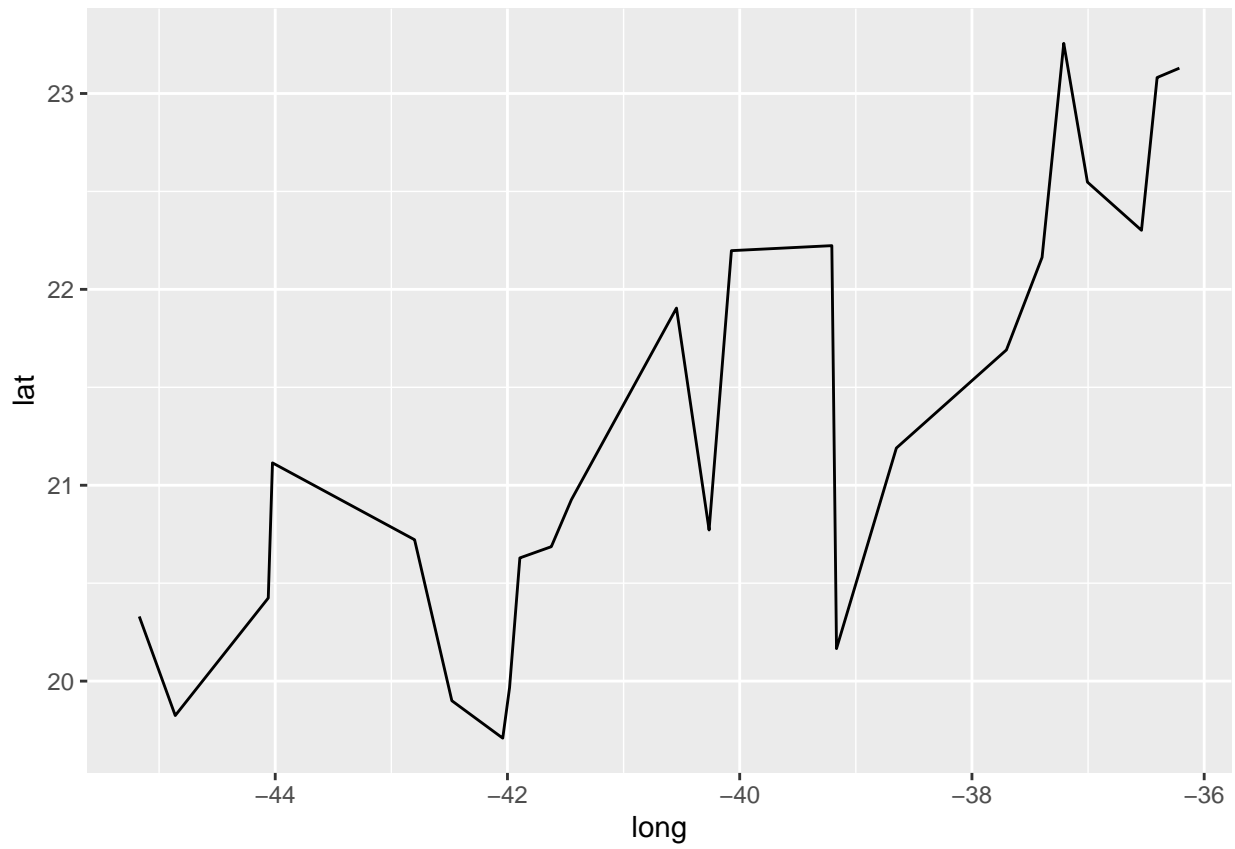


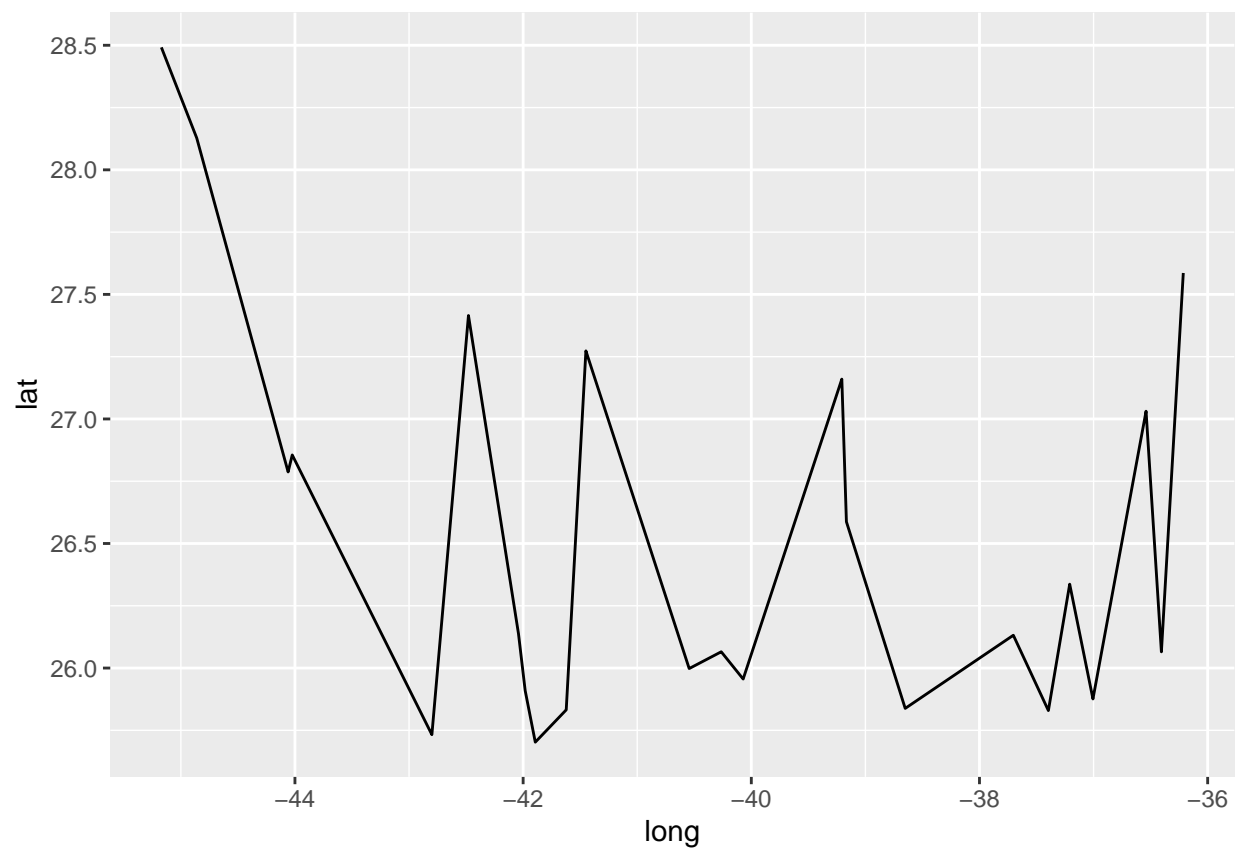


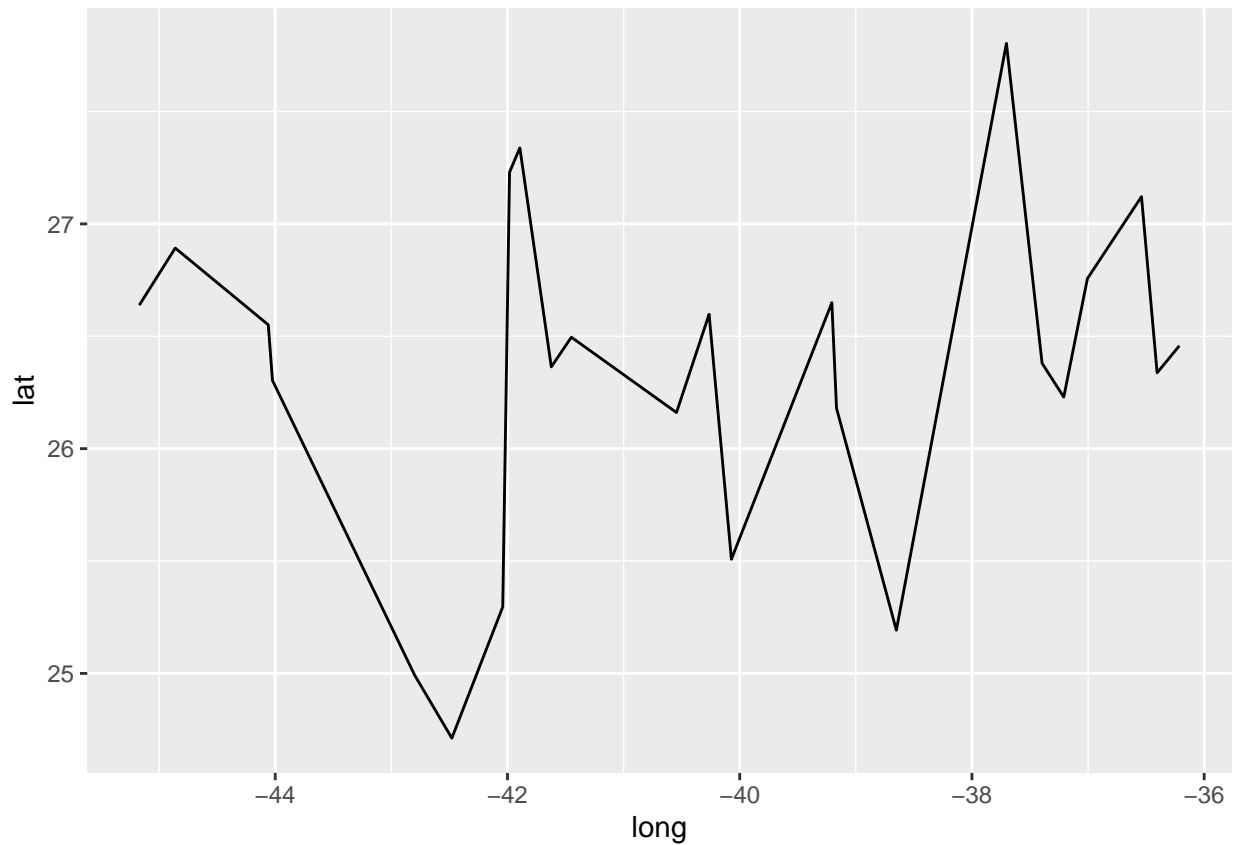






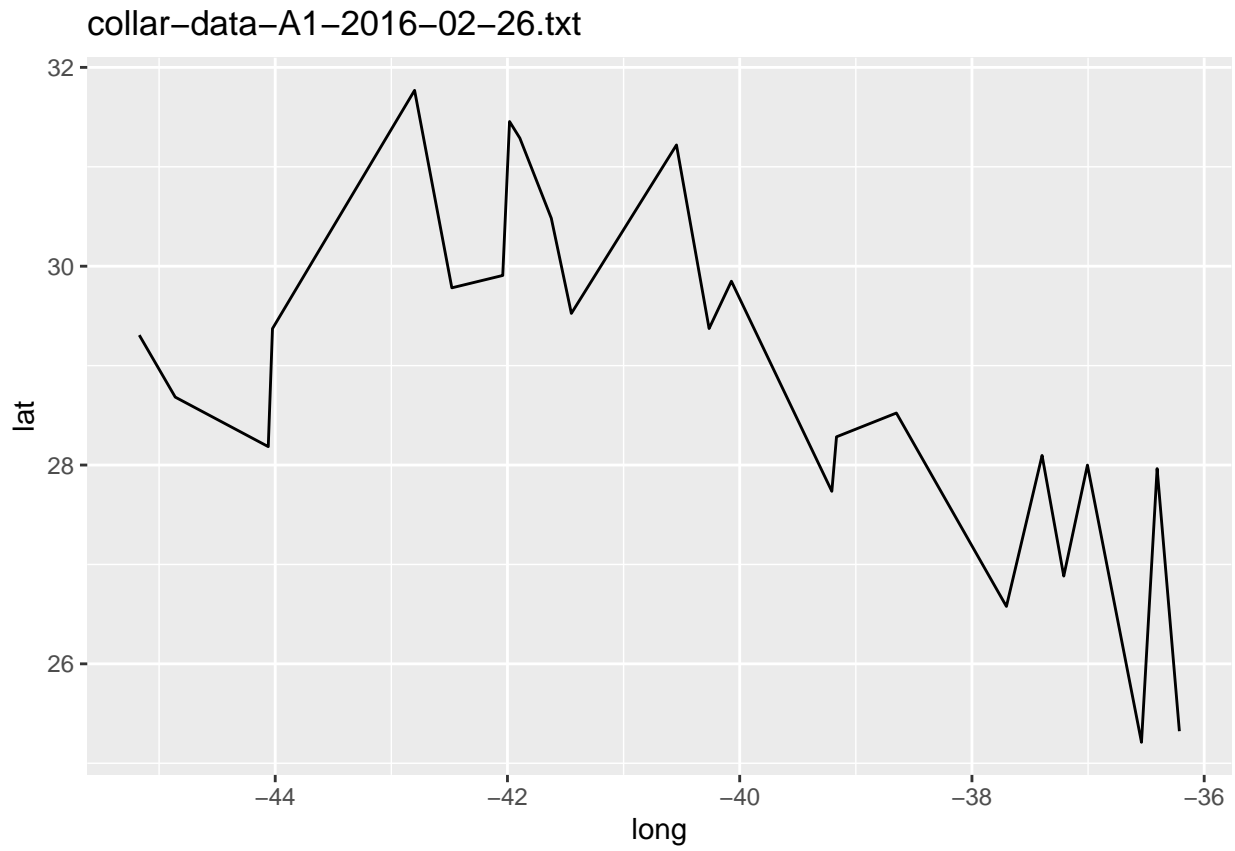




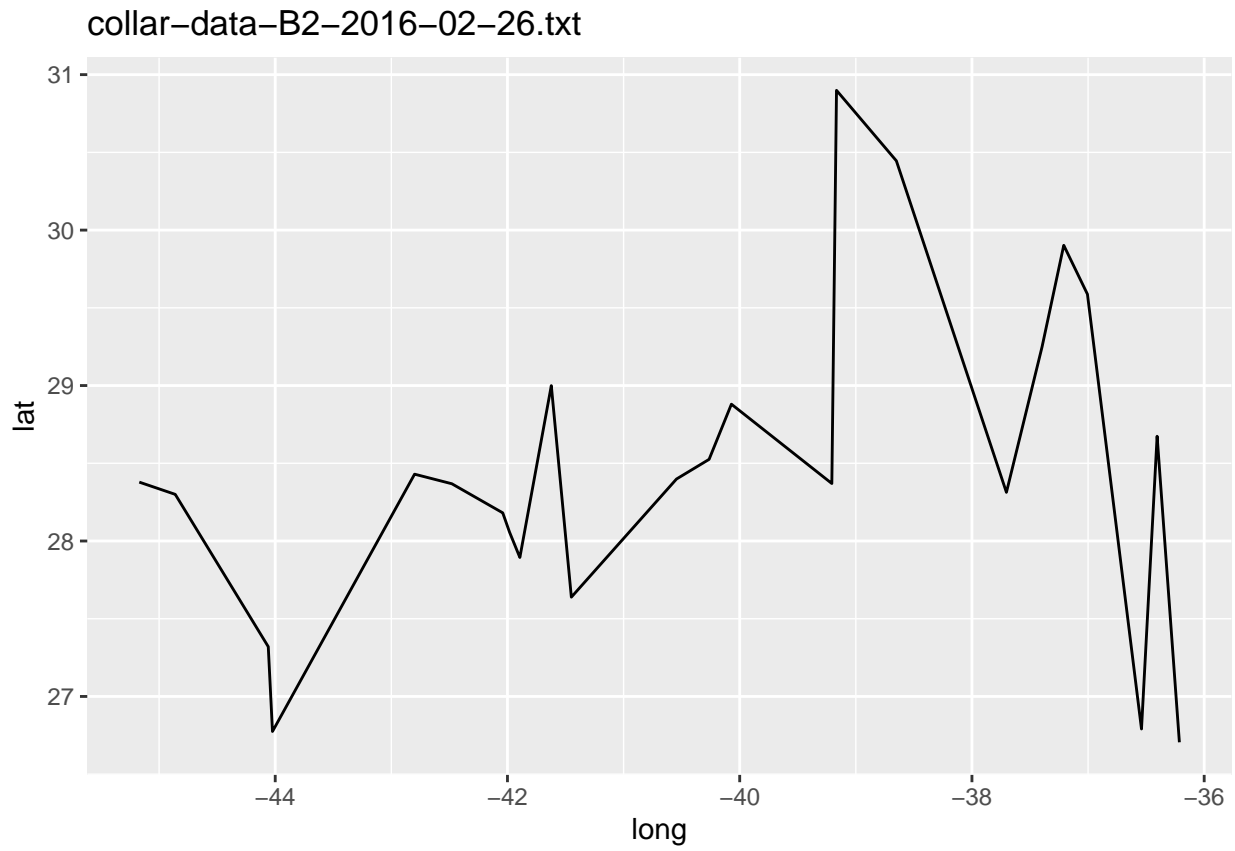


Include the name of the file in the graph as the graph title using the `ggtitle()`.

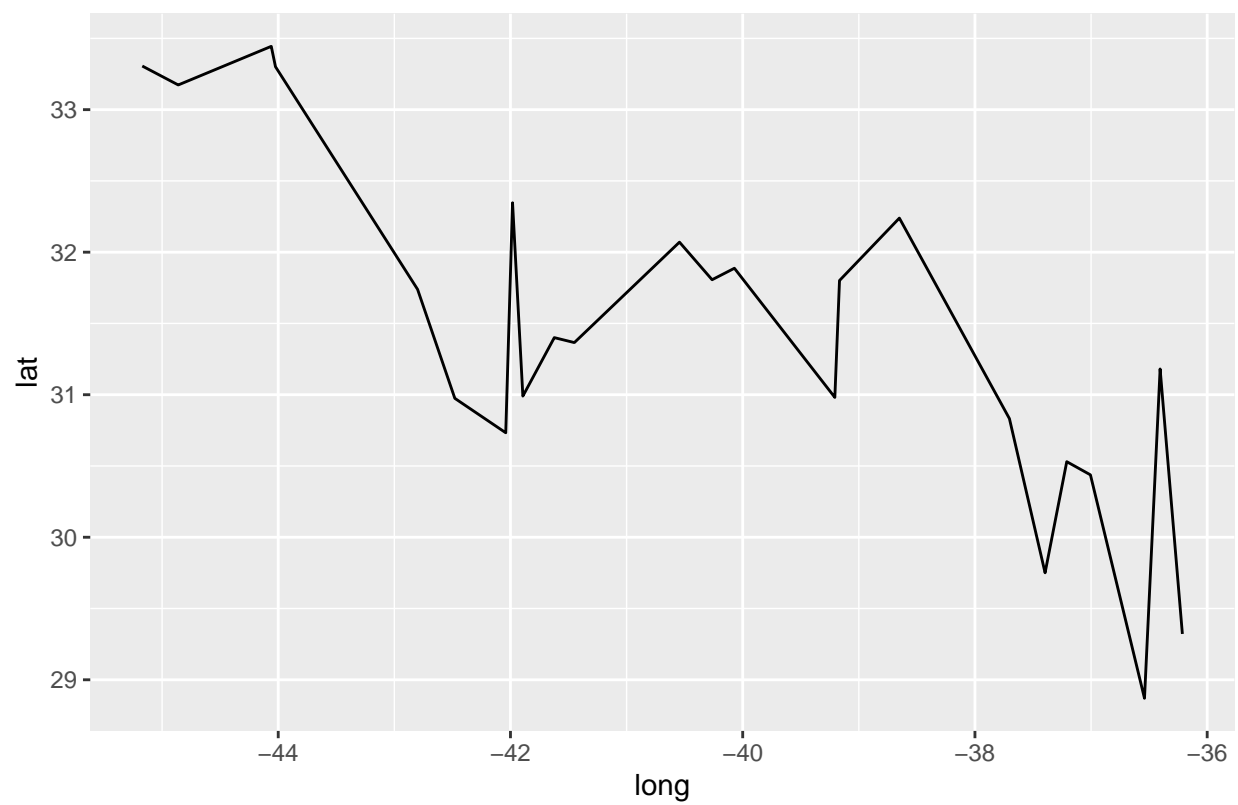
```
for (i in 1:num_files){  
  # specify the file  
  filename <- data_files[i]  
  # read in the text file  
  data <- read.csv(paste0("../Assignments/data_raw/", filename))  
  # make a plot  
  plot <- print(ggplot(data, aes(x= long, y = lat))+  
    geom_line()+  
    ggtitle(filename))  
  plot  
}
```



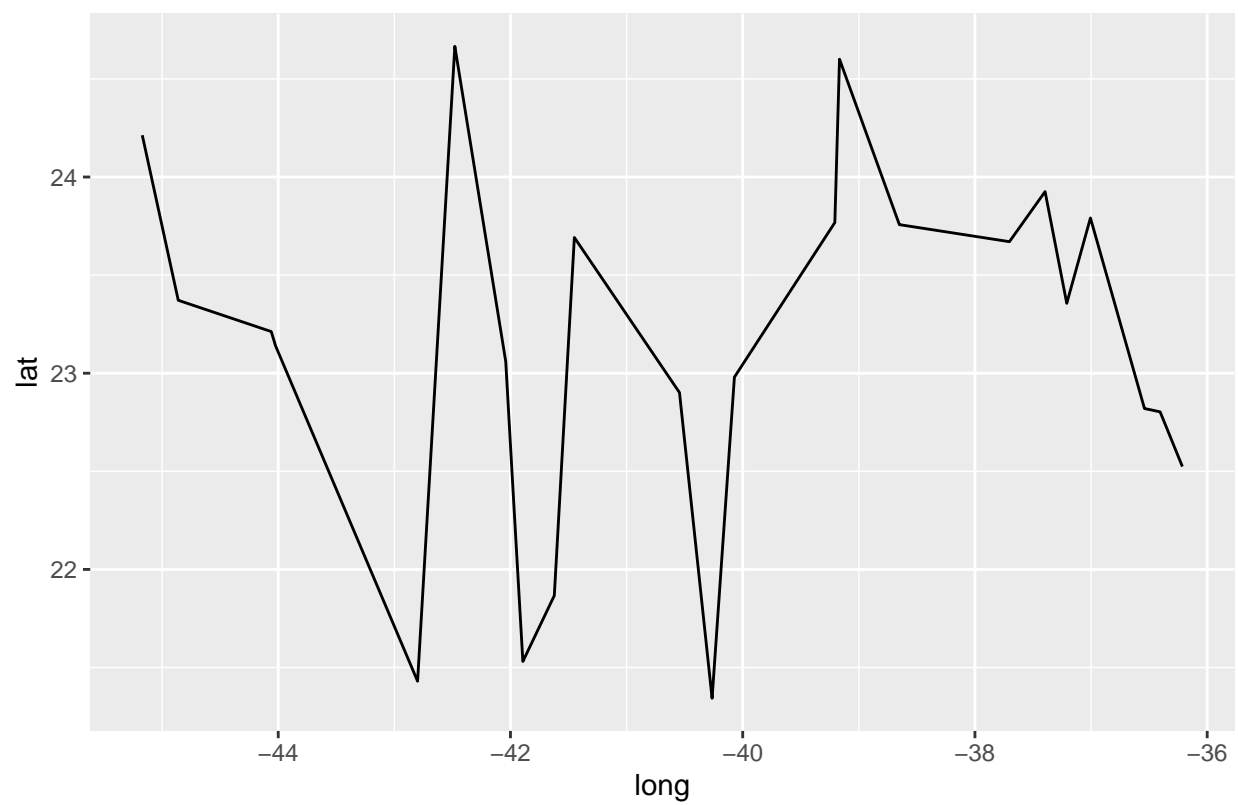




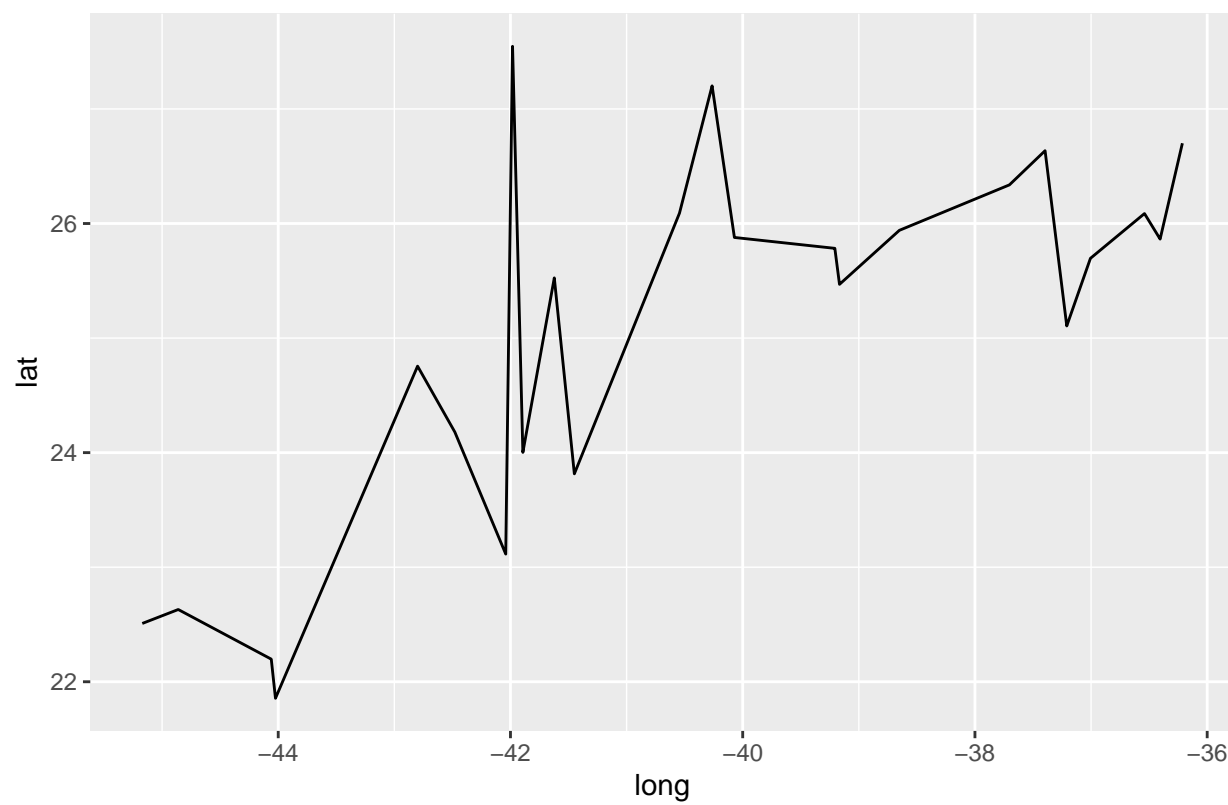
collar-data-C3-2016-02-26.txt



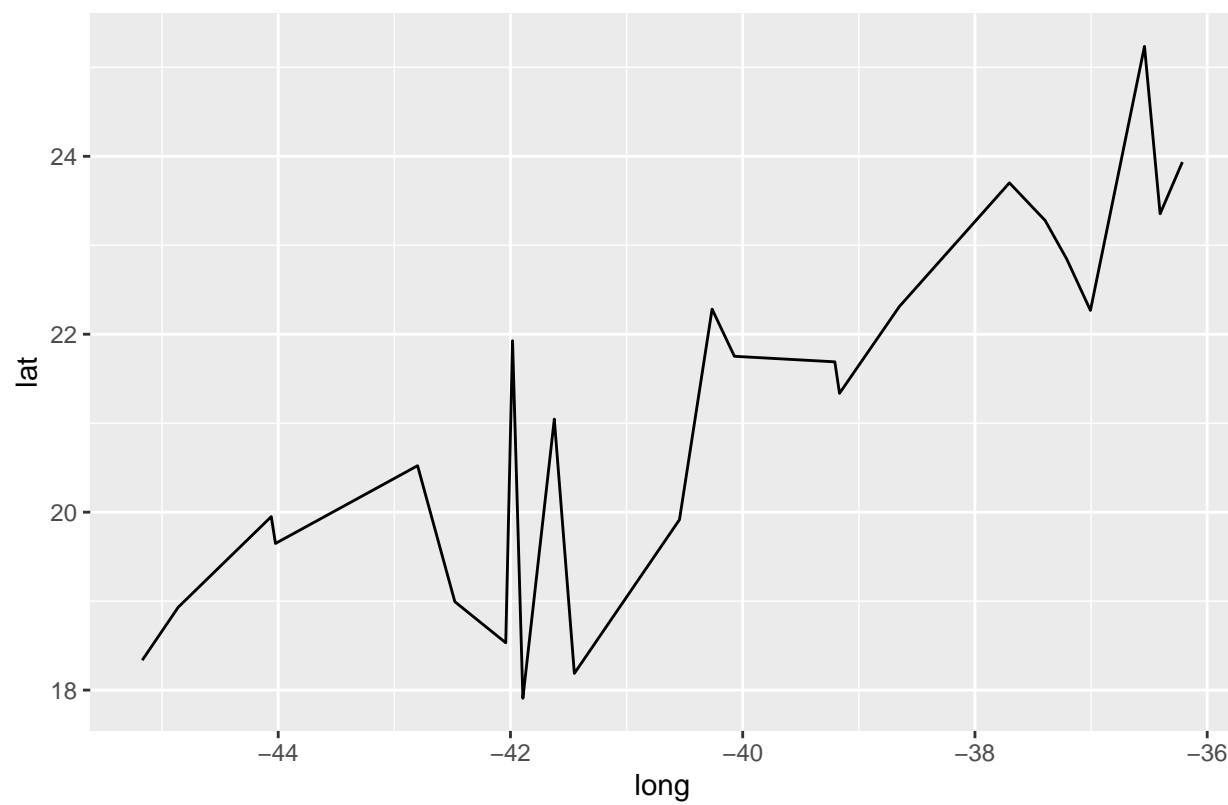
collar-data-D4-2016-02-26.txt



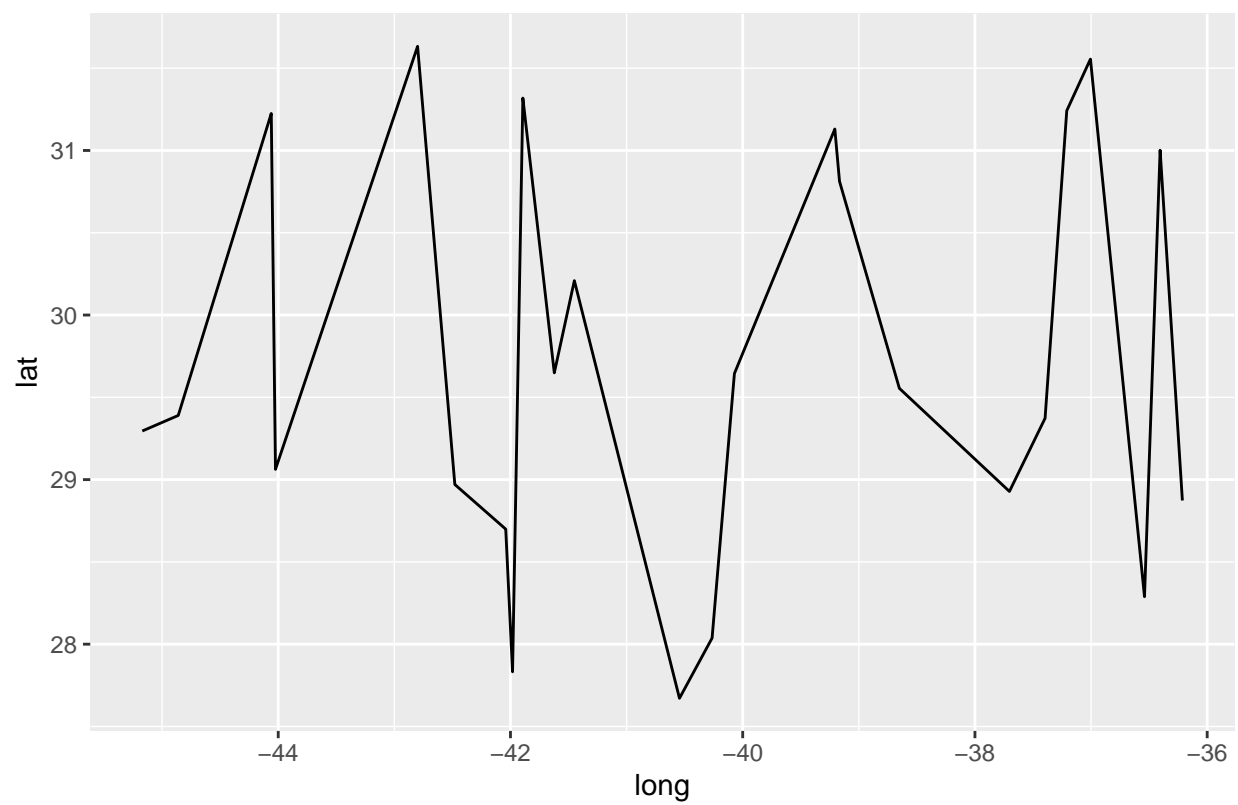
collar-data-E5-2016-02-26.txt



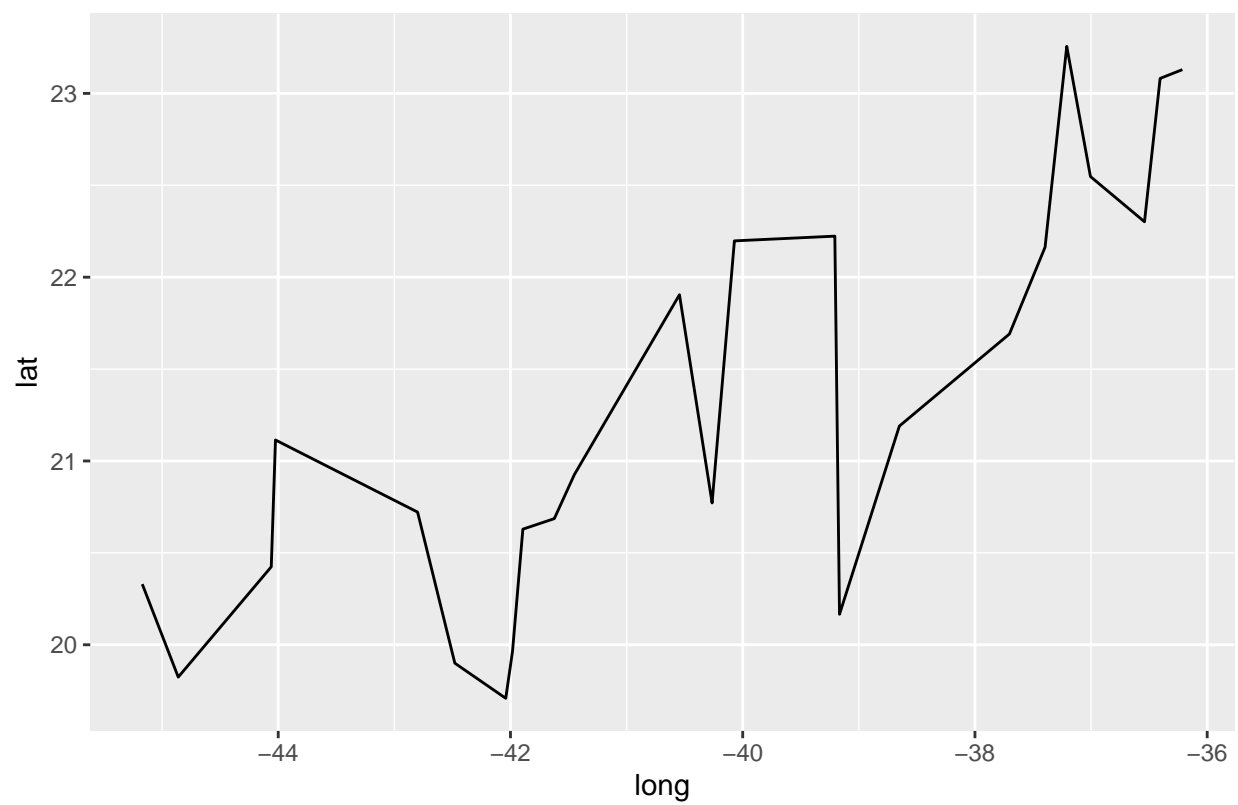
collar-data-F6-2016-02-26.txt

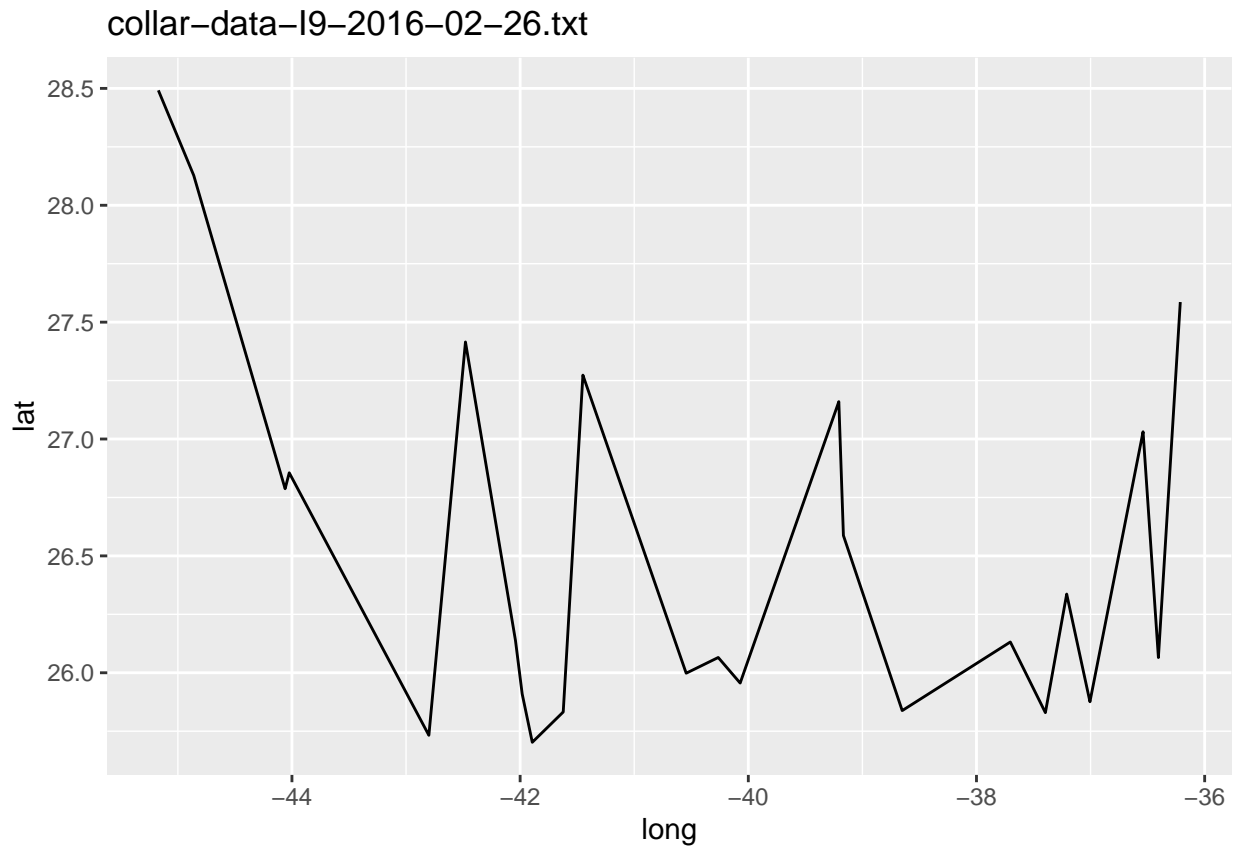


collar-data-G7-2016-02-26.txt



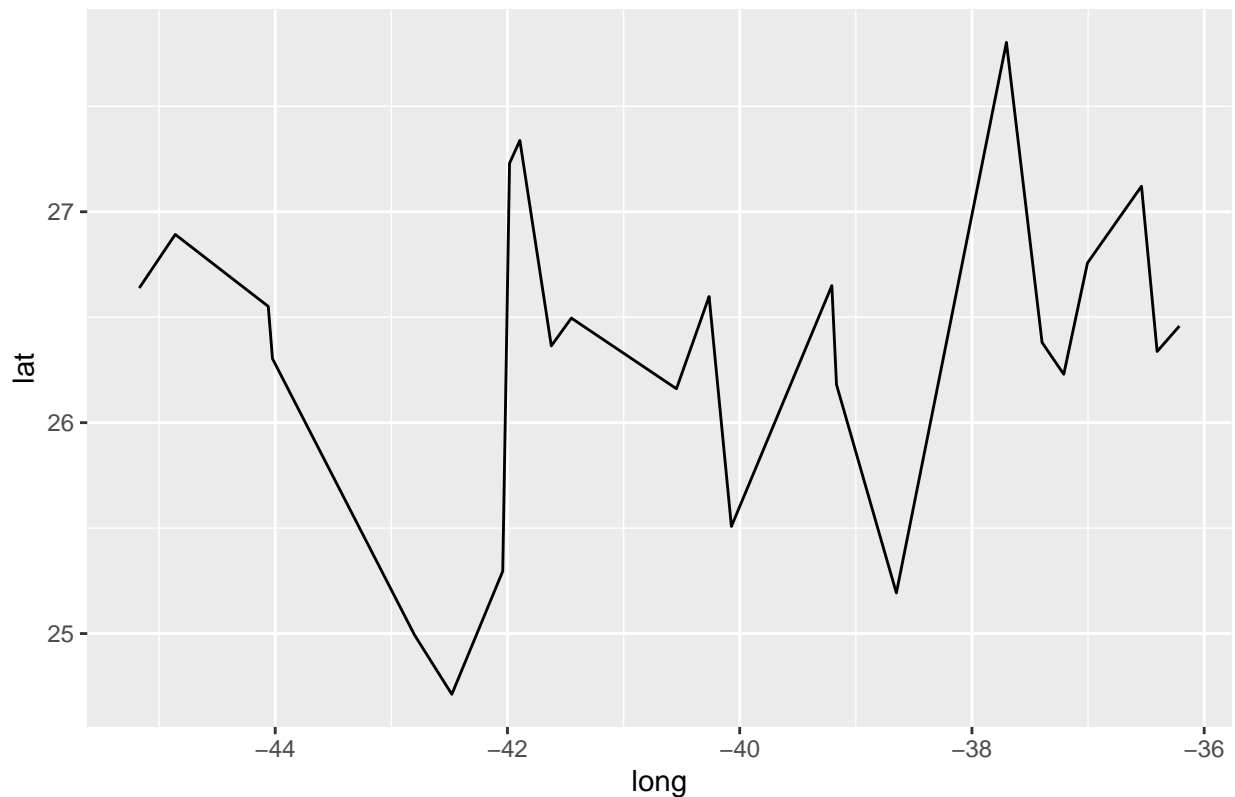
collar-data-H8-2016-02-26.txt







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



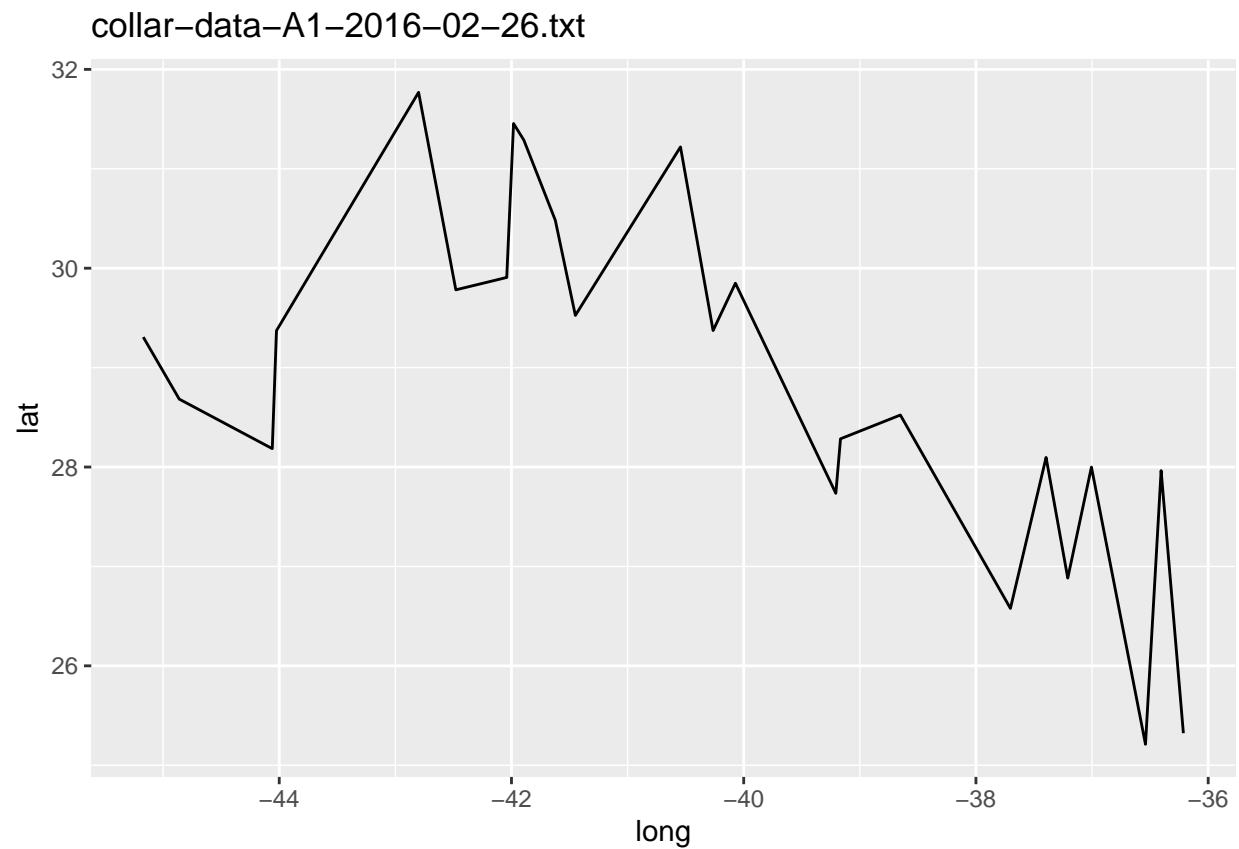
- b. Add code to the loop to calculate the minimum and maximum latitude in the file, and store these values, along with the name of the file, in a data frame.

```
results_3b <- data.frame(file_name = vector(mode = "character", length = num_files),
                          max_lat = vector(mode = "numeric", length = num_files),
                          min_lat = vector(mode = "numeric", length = num_files))

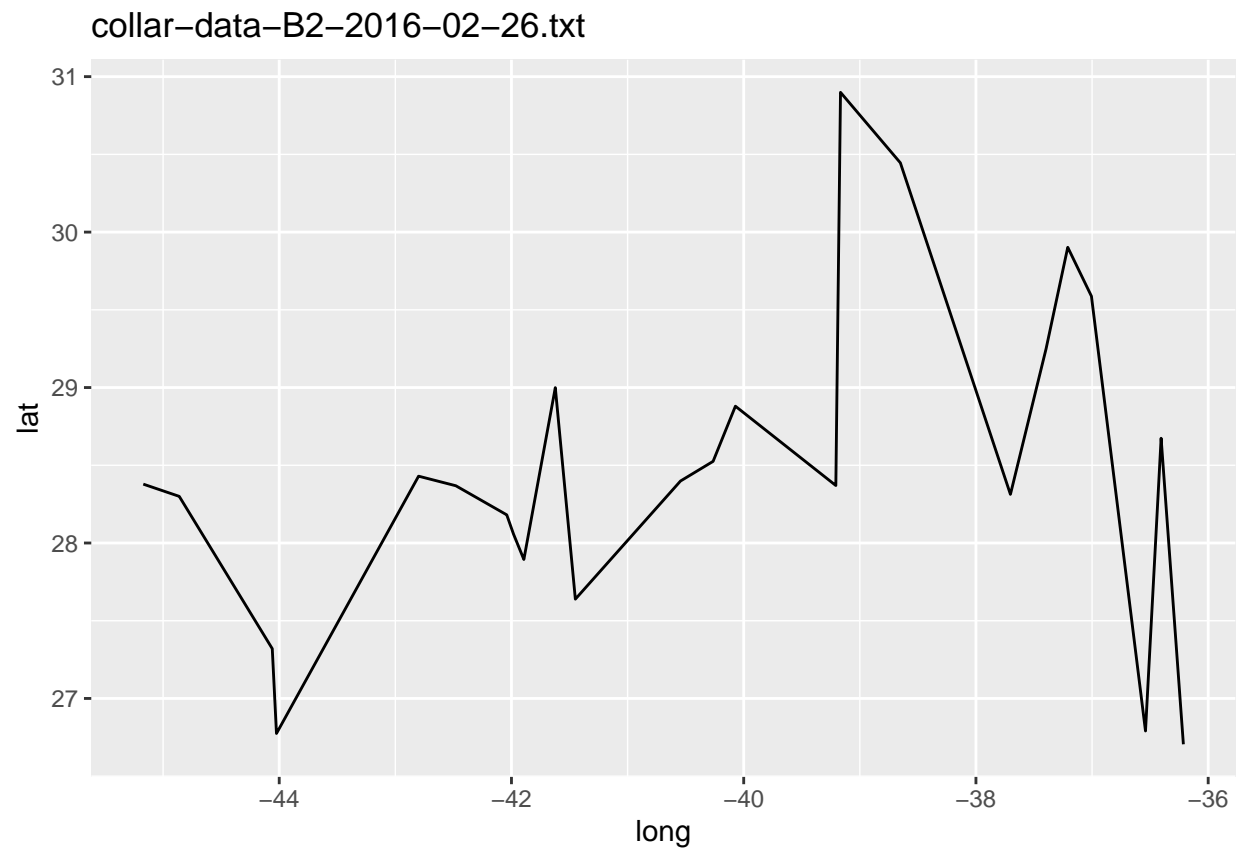
for (i in 1:num_files){
  # specify the file
  filename <- data_files[i]
  # read in the text file
  data <- read.csv(paste0("../Assignments/data_raw/", filename))
  # make a plot
  print(ggplot(data, aes(x= long, y = lat))+
        geom_line()+
        ggtitle(filename))

  results_3b$file_name[i] <- filename
  results_3b$max_lat[i] <- max(data$lat)
  results_3b$min_lat[i] <- min(data$lat)

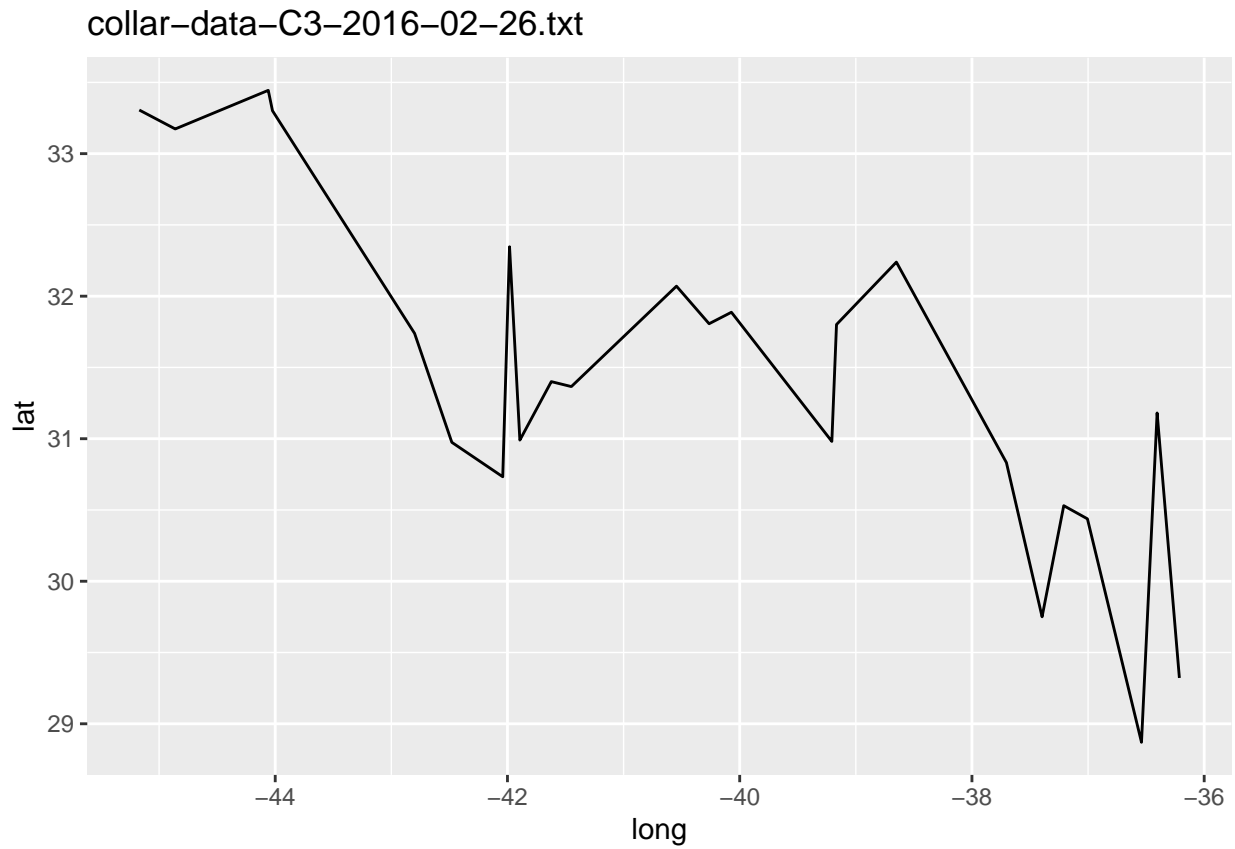
  print(results_3b)
}
```



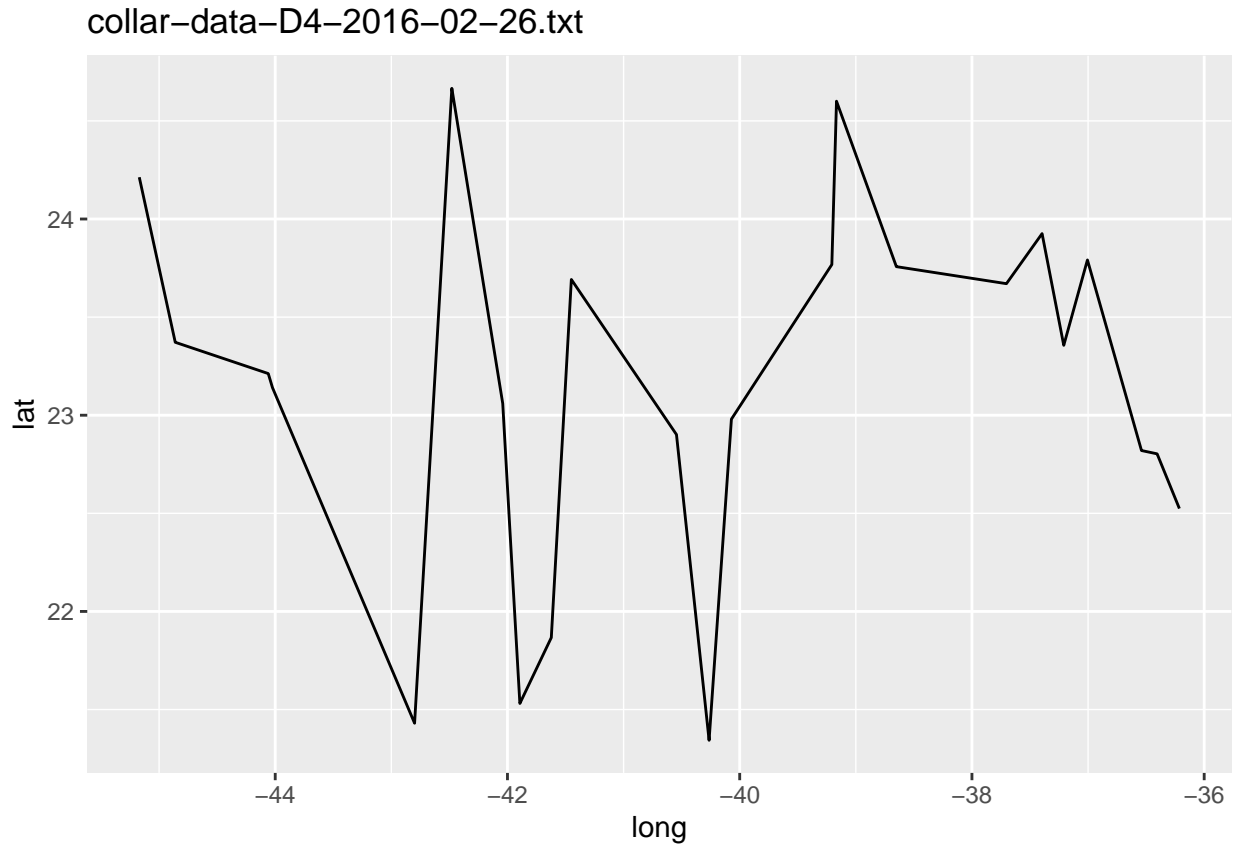
```
##          file_name  max_lat min_lat
## 1 collar-data-A1-2016-02-26.txt 31.76912 25.2108
## 2                                0.00000 0.0000
## 3                                0.00000 0.0000
## 4                                0.00000 0.0000
## 5                                0.00000 0.0000
## 6                                0.00000 0.0000
## 7                                0.00000 0.0000
## 8                                0.00000 0.0000
## 9                                0.00000 0.0000
## 10                               0.00000 0.0000
```



##	file_name	max_lat	min_lat
## 1	collar-data-A1-2016-02-26.txt	31.76912	25.21080
## 2	collar-data-B2-2016-02-26.txt	30.89907	26.70509
## 3		0.00000	0.00000
## 4		0.00000	0.00000
## 5		0.00000	0.00000
## 6		0.00000	0.00000
## 7		0.00000	0.00000
## 8		0.00000	0.00000
## 9		0.00000	0.00000
## 10		0.00000	0.00000

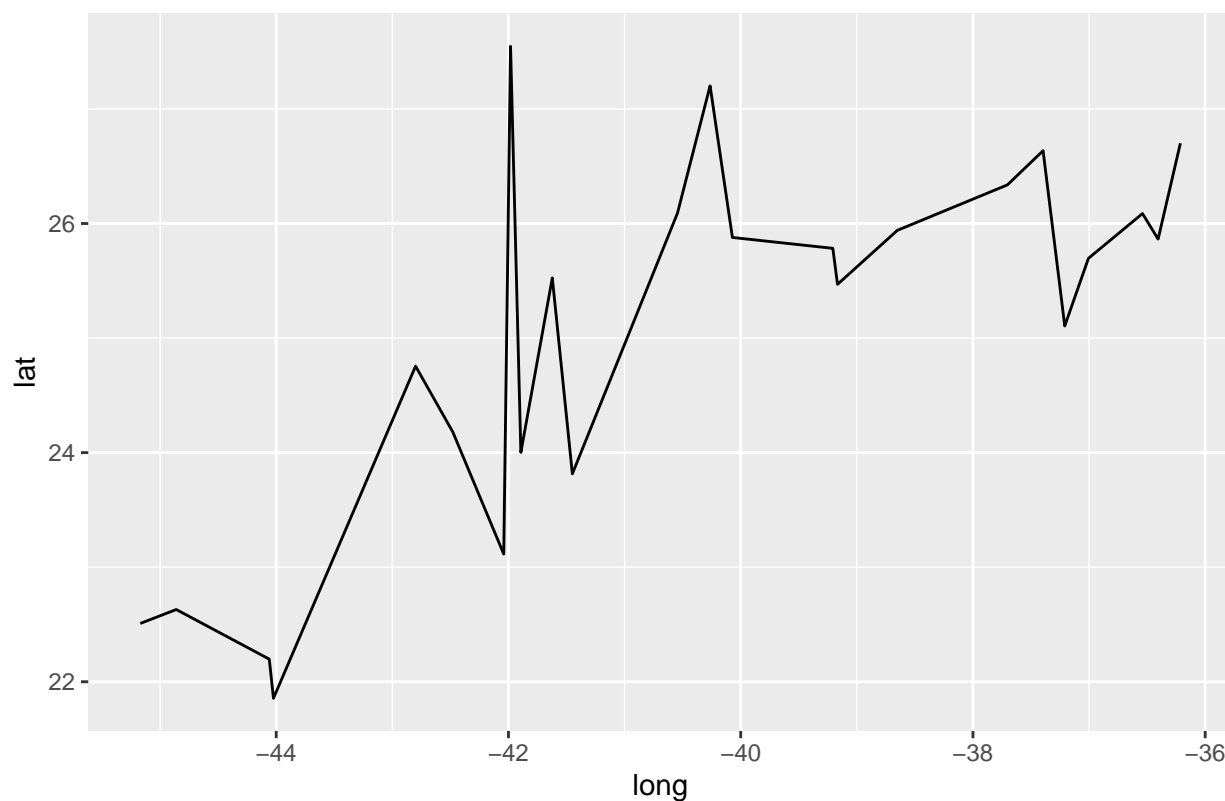


##	file_name	max_lat	min_lat
## 1	collar-data-A1-2016-02-26.txt	31.76912	25.21080
## 2	collar-data-B2-2016-02-26.txt	30.89907	26.70509
## 3	collar-data-C3-2016-02-26.txt	33.44421	28.86998
## 4		0.00000	0.00000
## 5		0.00000	0.00000
## 6		0.00000	0.00000
## 7		0.00000	0.00000
## 8		0.00000	0.00000
## 9		0.00000	0.00000
## 10		0.00000	0.00000



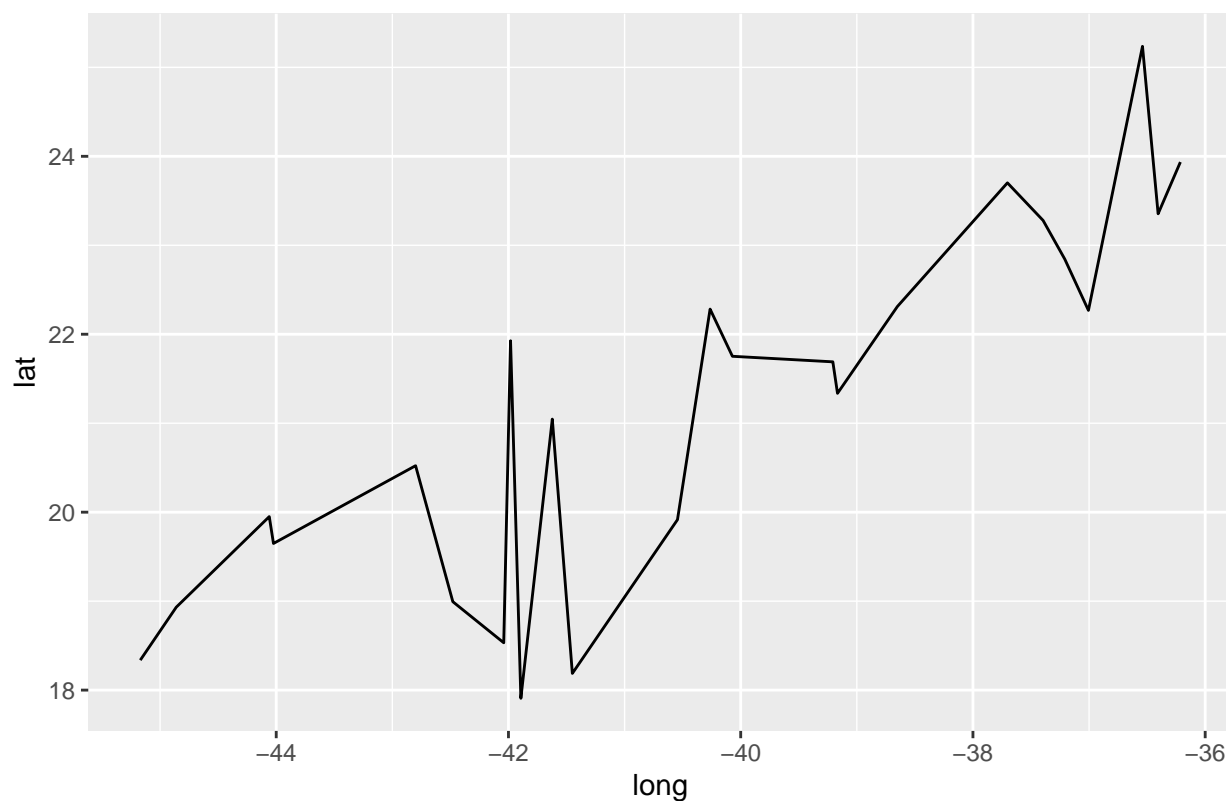
```
##           file_name max_lat min_lat
## 1 collar-data-A1-2016-02-26.txt 31.76912 25.21080
## 2 collar-data-B2-2016-02-26.txt 30.89907 26.70509
## 3 collar-data-C3-2016-02-26.txt 33.44421 28.86998
## 4 collar-data-D4-2016-02-26.txt 24.66598 21.34315
## 5           0.00000 0.00000
## 6           0.00000 0.00000
## 7           0.00000 0.00000
## 8           0.00000 0.00000
## 9           0.00000 0.00000
## 10          0.00000 0.00000
```

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



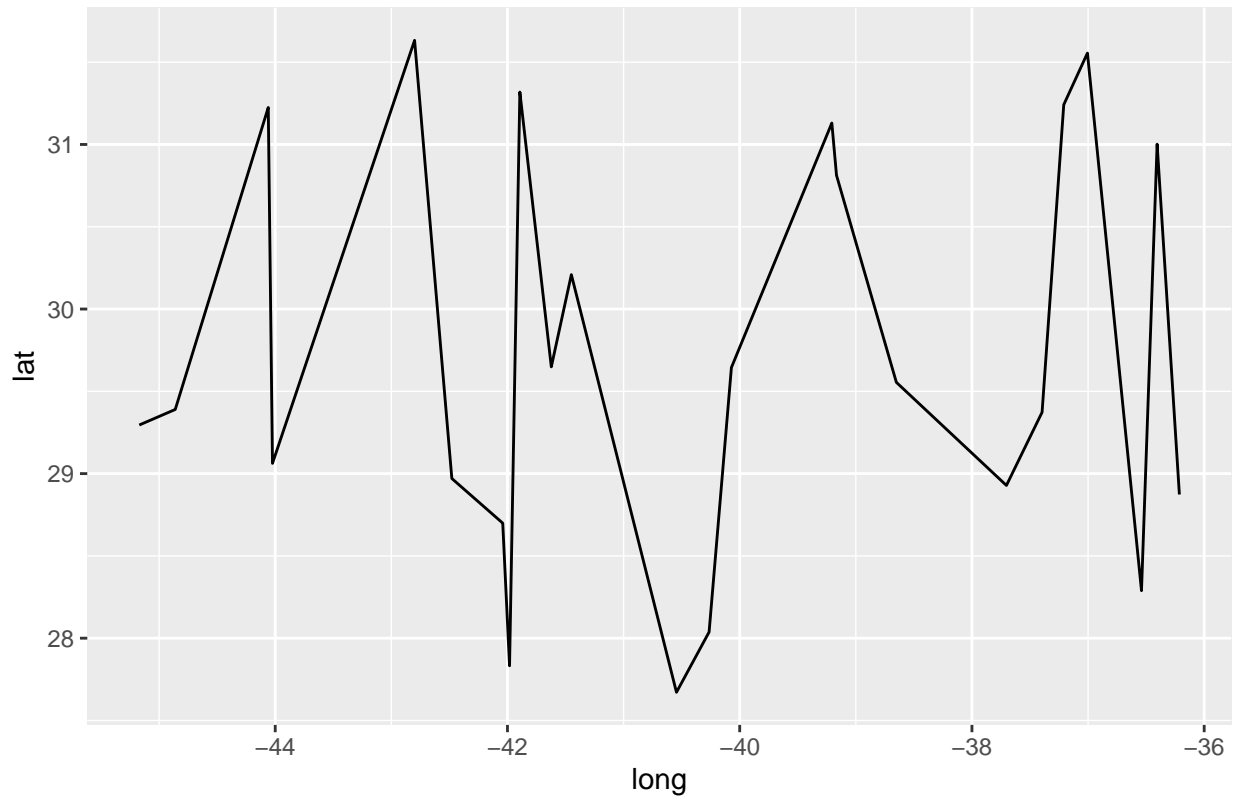
##		file_name	max_lat	min_lat
## 1	collar-data-A1-2016-02-26.txt	31.76912	25.21080	
## 2	collar-data-B2-2016-02-26.txt	30.89907	26.70509	
## 3	collar-data-C3-2016-02-26.txt	33.44421	28.86998	
## 4	collar-data-D4-2016-02-26.txt	24.66598	21.34315	
## 5	collar-data-E5-2016-02-26.txt	27.54663	21.85565	
## 6		0.00000	0.00000	
## 7		0.00000	0.00000	
## 8		0.00000	0.00000	
## 9		0.00000	0.00000	
## 10		0.00000	0.00000	

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



##		file_name	max_lat	min_lat
## 1	collar-data-A1-2016-02-26.txt	31.76912	25.21080	
## 2	collar-data-B2-2016-02-26.txt	30.89907	26.70509	
## 3	collar-data-C3-2016-02-26.txt	33.44421	28.86998	
## 4	collar-data-D4-2016-02-26.txt	24.66598	21.34315	
## 5	collar-data-E5-2016-02-26.txt	27.54663	21.85565	
## 6	collar-data-F6-2016-02-26.txt	25.23623	17.90788	
## 7		0.00000	0.00000	
## 8		0.00000	0.00000	
## 9		0.00000	0.00000	
## 10		0.00000	0.00000	

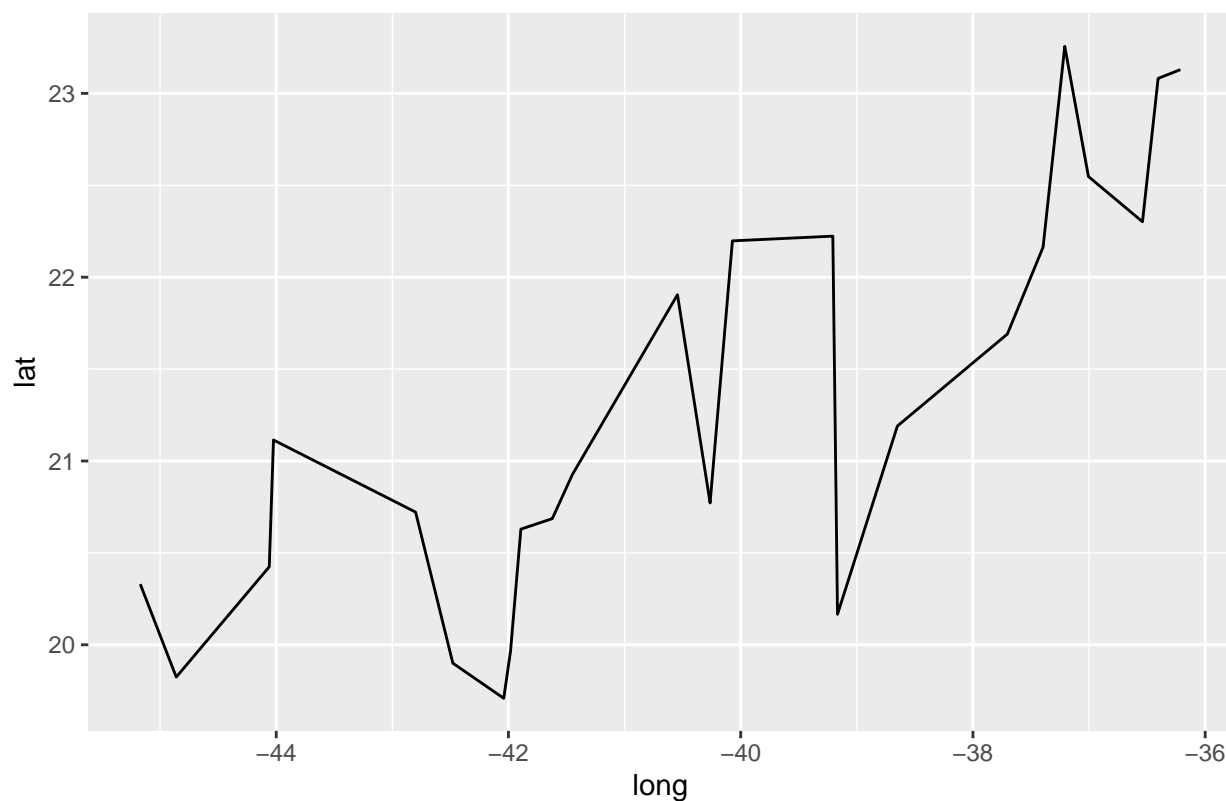
collar-data-G7-2016-02-26.txt



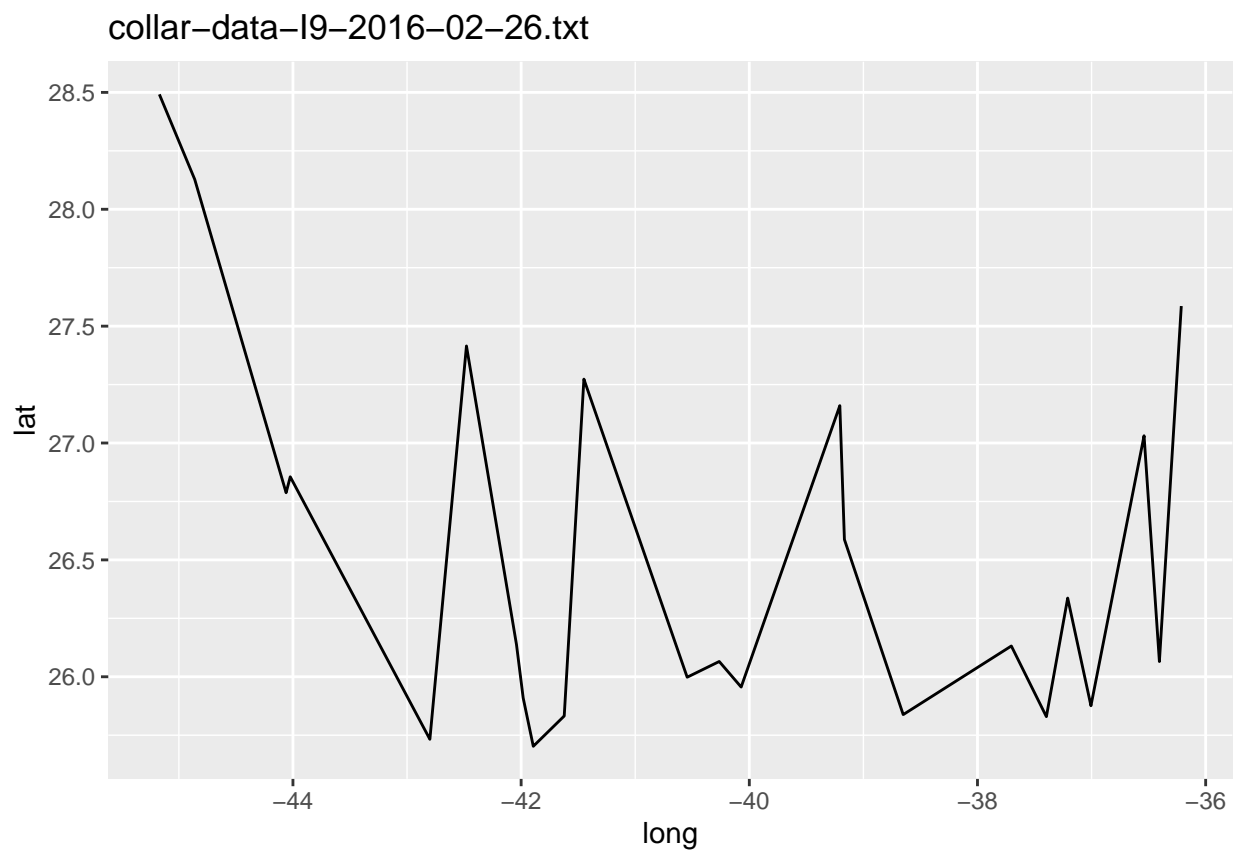
##		file_name	max_lat	min_lat
## 1		collar-data-A1-2016-02-26.txt	31.76912	25.21080
## 2		collar-data-B2-2016-02-26.txt	30.89907	26.70509
## 3		collar-data-C3-2016-02-26.txt	33.44421	28.86998
## 4		collar-data-D4-2016-02-26.txt	24.66598	21.34315
## 5		collar-data-E5-2016-02-26.txt	27.54663	21.85565
## 6		collar-data-F6-2016-02-26.txt	25.23623	17.90788
## 7		collar-data-G7-2016-02-26.txt	31.63272	27.67120
## 8			0.00000	0.00000
## 9			0.00000	0.00000
## 10			0.00000	0.00000



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

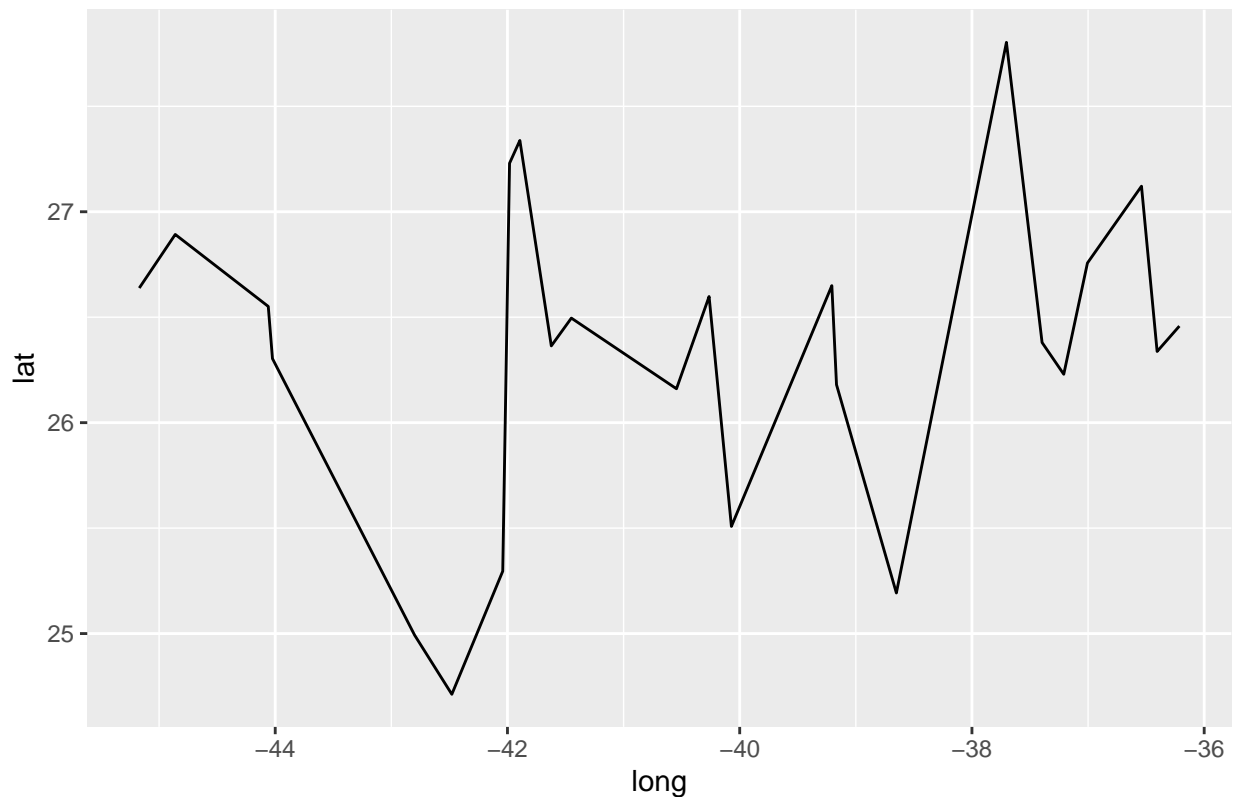


##		file_name	max_lat	min_lat
## 1	collar-data-A1-2016-02-26.txt	31.76912	25.21080	
## 2	collar-data-B2-2016-02-26.txt	30.89907	26.70509	
## 3	collar-data-C3-2016-02-26.txt	33.44421	28.86998	
## 4	collar-data-D4-2016-02-26.txt	24.66598	21.34315	
## 5	collar-data-E5-2016-02-26.txt	27.54663	21.85565	
## 6	collar-data-F6-2016-02-26.txt	25.23623	17.90788	
## 7	collar-data-G7-2016-02-26.txt	31.63272	27.67120	
## 8	collar-data-H8-2016-02-26.txt	23.25601	19.70875	
## 9		0.00000	0.00000	
## 10		0.00000	0.00000	



##		file_name	max_lat	min_lat
## 1		collar-data-A1-2016-02-26.txt	31.76912	25.21080
## 2		collar-data-B2-2016-02-26.txt	30.89907	26.70509
## 3		collar-data-C3-2016-02-26.txt	33.44421	28.86998
## 4		collar-data-D4-2016-02-26.txt	24.66598	21.34315
## 5		collar-data-E5-2016-02-26.txt	27.54663	21.85565
## 6		collar-data-F6-2016-02-26.txt	25.23623	17.90788
## 7		collar-data-G7-2016-02-26.txt	31.63272	27.67120
## 8		collar-data-H8-2016-02-26.txt	23.25601	19.70875
## 9		collar-data-I9-2016-02-26.txt	28.49172	25.70252
## 10			0.00000	0.00000

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



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

Show the data frame as output.

```
for (i in 1:num_files){
  # specify the file
  filename <- data_files[i]
  # read in the text file
  data <- read.csv(paste0("../Assignments/data_raw/", filename))
  # make a plot
  #print(ggplot(data, aes(x= long, y = lat))+
  #  geom_line()+
  #  ggtitle(filename))

  results_3b$file_name[i] <- filename
}
```

```

results_3b$max_lat[i] <- max(data$lat)
results_3b$min_lat[i] <- min(data$lat)

print(results_3b)
}

```

```

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

```

```

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

```

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

## Reflection

This reflection is worth 25 points, separate from your Week 12 assignment. Reflections are graded for completion only.

Write about 5 sentences addressing *at least* one of the following questions.

- What has worked well for you in this course for you so far?
- What has been particularly challenging for you so far?
- Is there anything that I can do to help your learning in the course?
- Other reflections about the content of the course that you would like to share.

*Answer:*