

# JW\_Loops

James Waterford

2023-04-06

## Loops

Loops, not dissimilar to those in Python, help you iterate a task for a given set of intervals. For loops are another way that we can tell a computer to repeat tasks for us. They are versatile and very explicit, so that means that we are controlling everything that is run on each iteration of the loop (mostly).

The general structure: `for ([value] in [list]) {        do_something_with(values)        }`

```
lengths <- c(13.3,15,100)

for (value in lengths) {
  mass <- 0.73 * value^2
  print(mass)
  ## `return()` does not work in loops ##
}
```

```
## [1] 129.1297
## [1] 164.25
## [1] 7300
```

```
mass_lbs <- c(2.2,3.5,9.6,1.2)
for (mass_lb in mass_lbs) {
  mass_kg = mass_lb* 2.2
  print(mass_kg)
}
```

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

What is an index in R? It is the numeric position of values inside any data structure in R. We can use indexes to sort through the full length of a

```
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"
```

How do we save the results from a For Loop? When using `vector()`, we don't need insert the mode or length

```
flowers <- c("daisies","lilacs","lillies")
my_results <- vector(mode = "character", length = length(flowers))
for (i in 1:length(flowers)) {
  upper <- toupper(flowers[i])
  my_results[i] <-(upper)
}
my_results
```

```
## [1] "DAISES" "LILACS" "LILLIES"
```

```
lengths <- c(1.1, 2.2, 1.6)
widths <- c(3.5, 2.4, 2.8)
areas <- vector(mode = "numeric", length = length(lengths))
for (i in 1:length(lengths)) {
  areas[i] <- lengths[i] * widths[i]
}
areas
```

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

```
#1#

dino_dat <- read.csv("../197-raw_storage/dinosaur_lengths.csv")

get_mass_from_length_by_name <-function(length,name){
  if(name == "Stegosauria") {
    a = 10.95
    b = 2.64
  } else if (name == "Theropoda") {
    a = 0.73
    b = 3.63
  } else if (name == "Sauropoda") {
    a = 214.44
    b = 1.46
  } else {
    a = 25.37
    b = 2.49
  }
  mass <- a * length^b
  return(mass)
}

#2#
# Come back to this
```

```
#results <- get_mass_from_length_by_name(length = dino_dat$lengths, dino_dat$species)
#3#
```

```
head(dino_dat)
```

```
##      species  lengths
## 1 Stegosauria 18.52588
## 2 Ankylosauria 16.43598
## 3 Ankylosauria 23.73421
## 4  Sauropoda 23.93411
## 5 Ankylosauria 21.68718
## 6 Ankylosauria 21.38363
```

```
#dino_dat$mass <- results
```

```
#4#
```

```
dino_dat %>%
  group_by(species) %>%
  summarize(mean_mass = (sum(mass)/length(lengths)))
```

```
## # A tibble: 4 x 2
##   species      mean_mass
##   <chr>         <dbl>
## 1 Ankylosauria     52.9
## 2 Sauropoda       61.3
## 3 Stegosauria     62.9
## 4 Theropoda       57.5
```

```
library(ggplot2)
```

```
#download.file(url = "http://www.datacarpentry.org/semester-biology/data/individual_collar_data.zip.",
```

```
file <- unzip("../197-raw_storage/individual_collar_data.zip")
```

```
list.files(pattern = "collar-data-*.txt") -> cfiles
```

```
reader <- function(filelist) {
```

```
  for (i in 1:length(filelist)) {
    read.csv(filelist[i]) -> data_grab
    ggplot(data = data_grab, mapping = aes(x=long, y=lat)) +
      geom_path() +
      ggtitle(filelist[i])
    ggsave(paste0("plot_",i,".pdf"), device = "pdf")
  }
```

```
}
```

```
reader(filelist = cfiles)
```

```
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
```

```
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
```

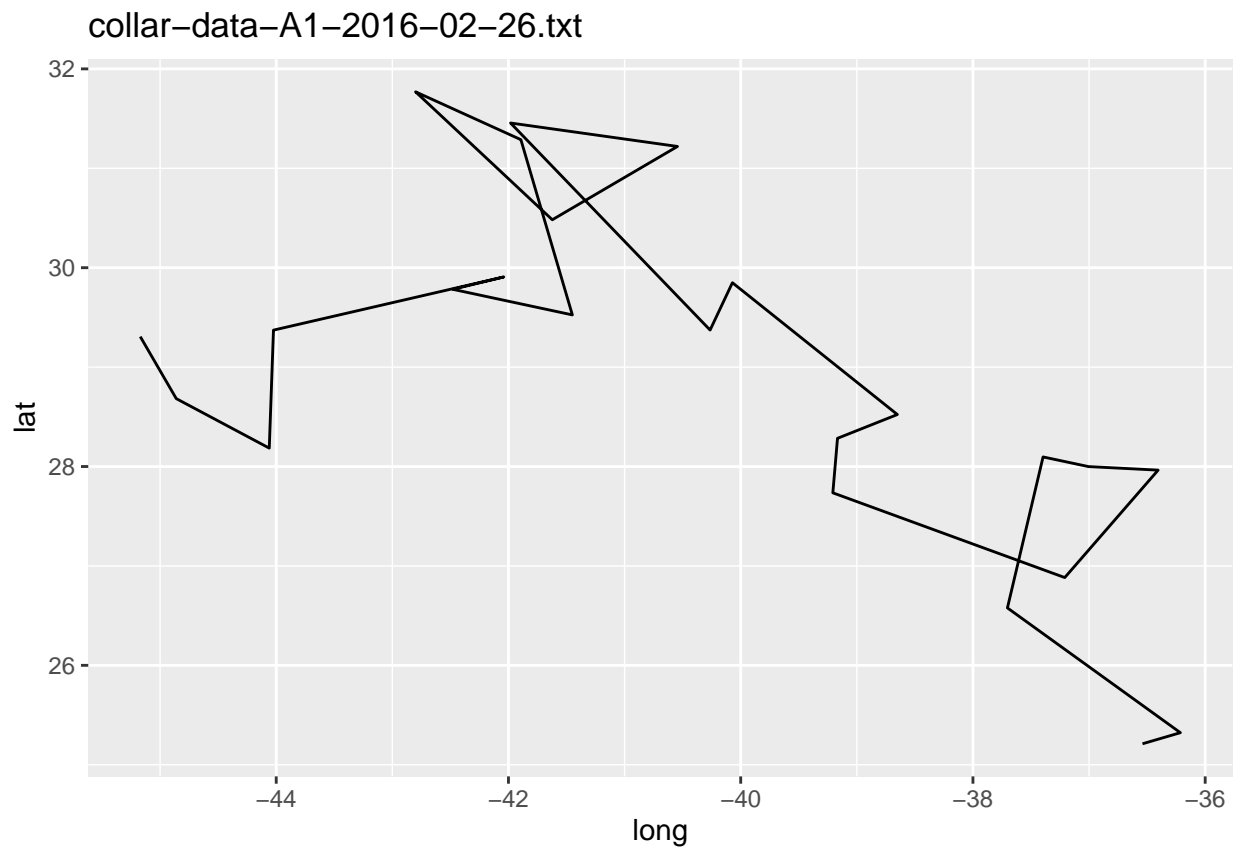
```
reader <- function(filelist) {

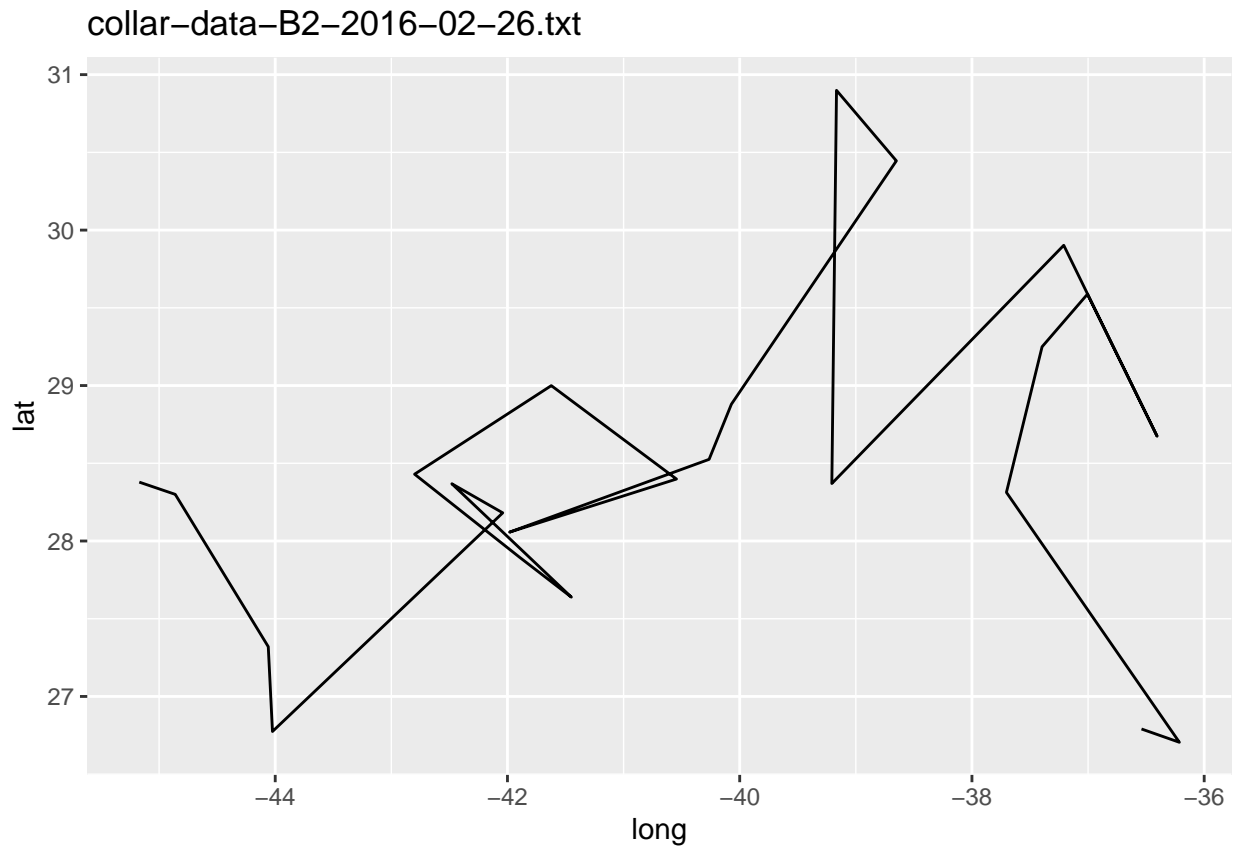
  for (i in 1:length(filelist)) {
    read.csv(filelist[i]) -> data_grab
    ggplot(data = data_grab, mapping = aes(x=long, y=lat)) +
      geom_path() +
      ggtitle(filelist[i]) -> test

    print(test)
  }

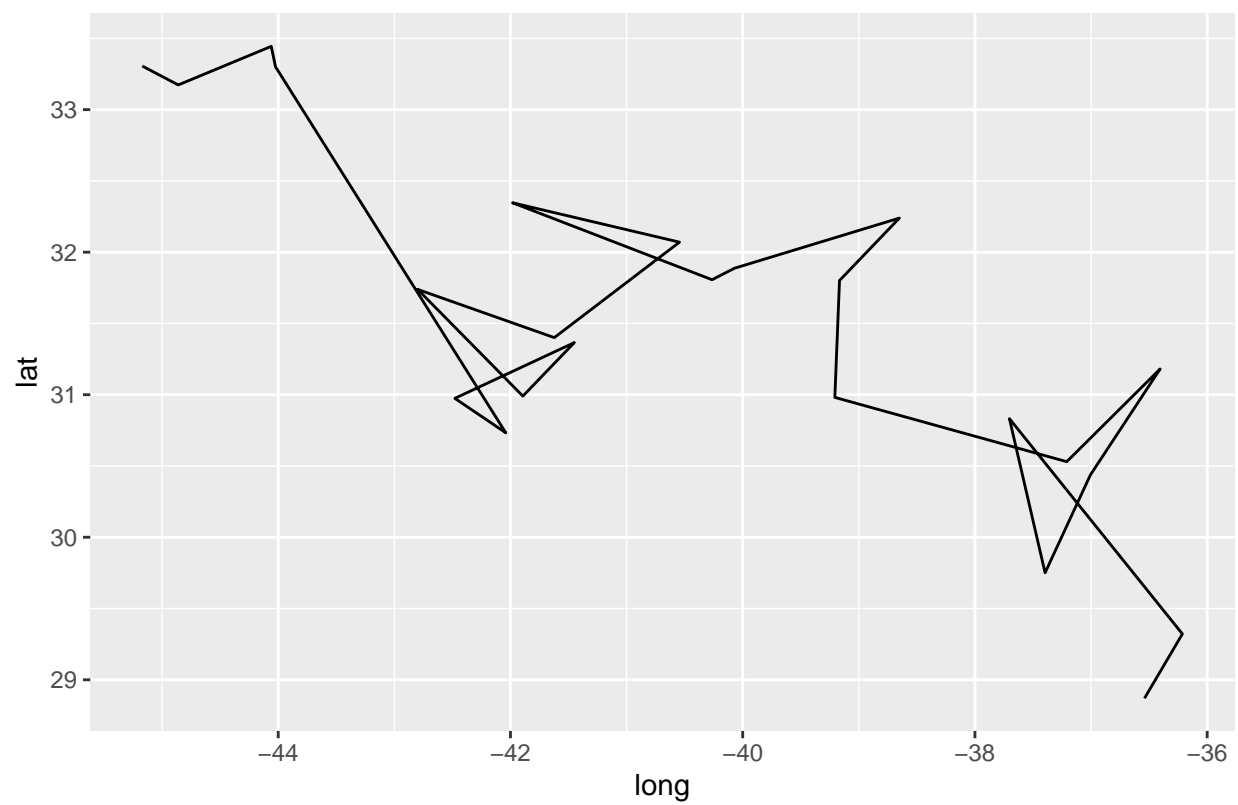
}

reader(filelist = cfiles)
```

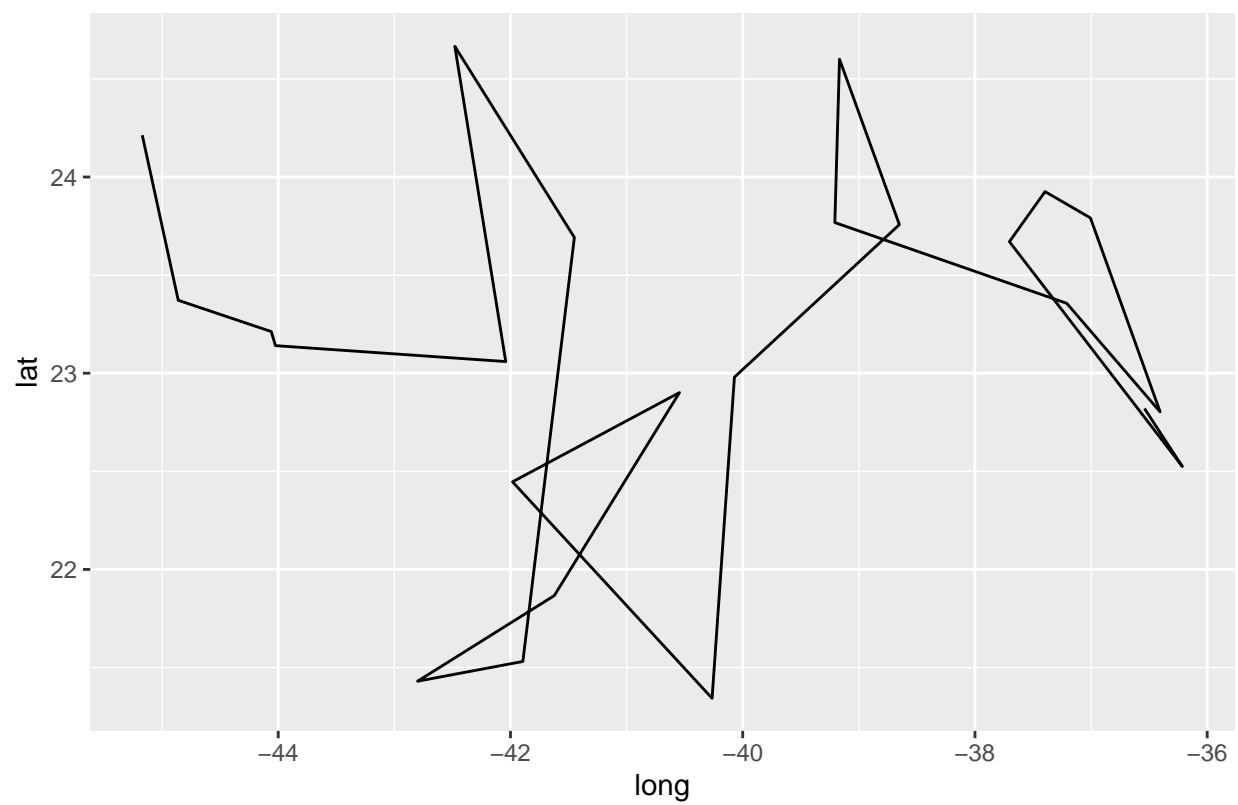




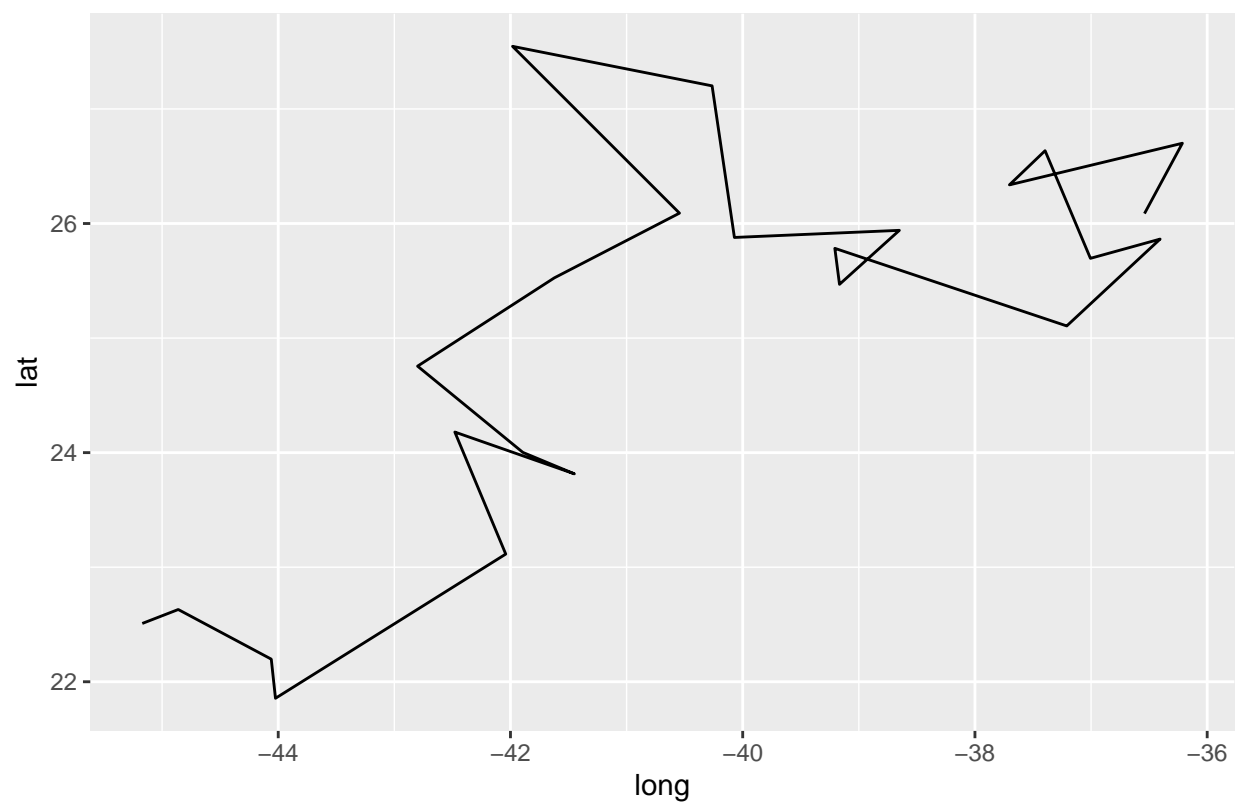
collar-data-C3-2016-02-26.txt



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

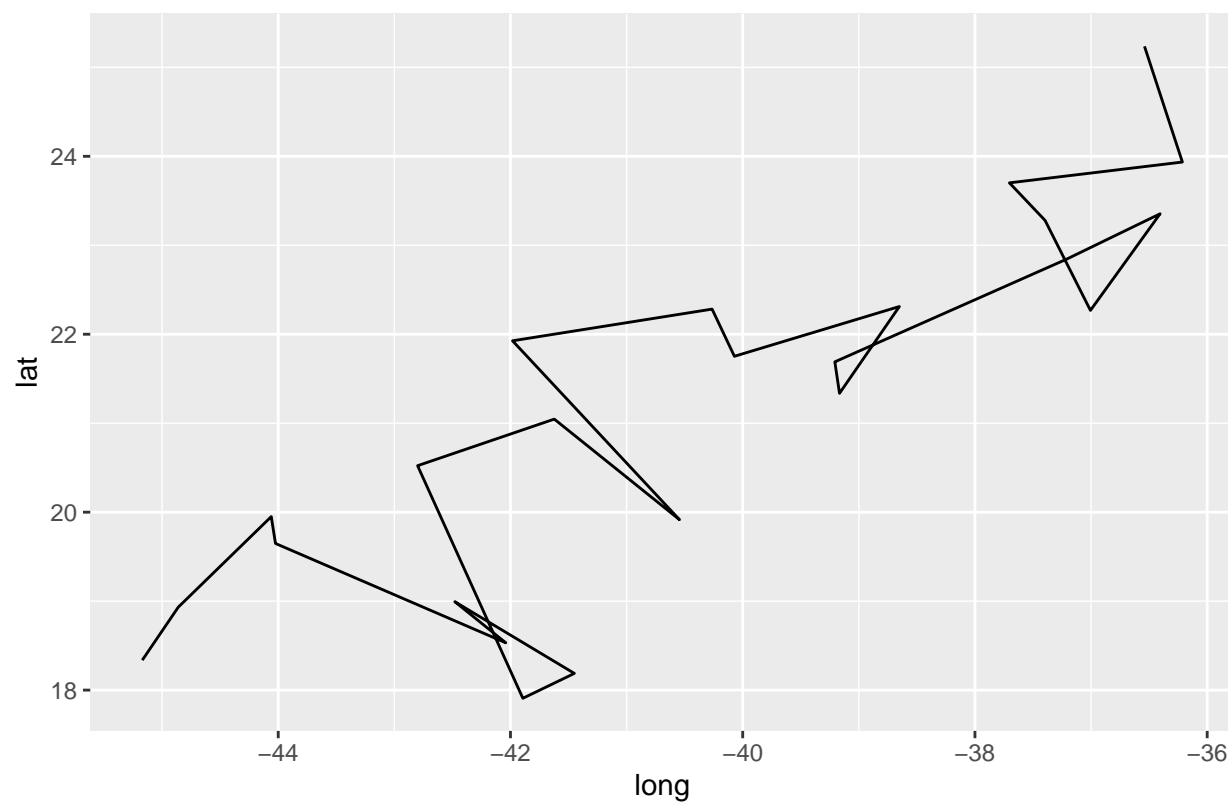


collar-data-E5-2016-02-26.txt

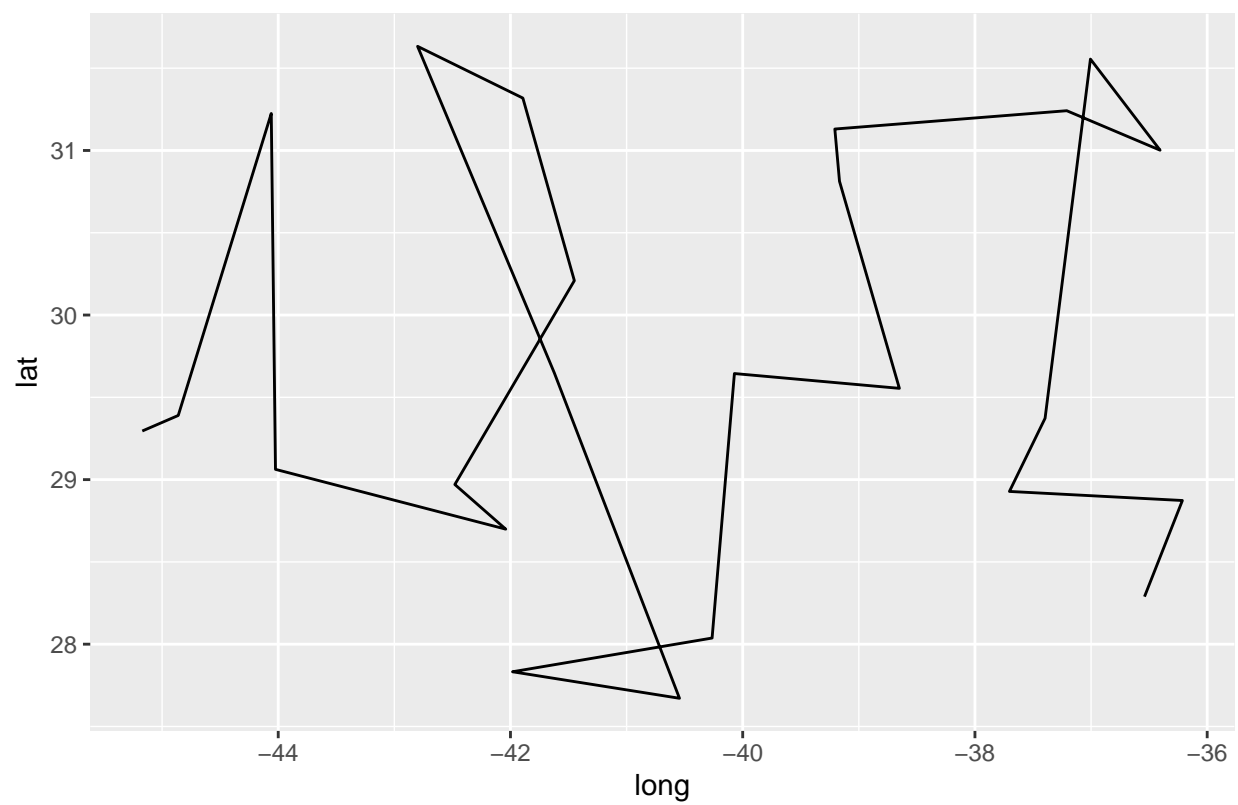




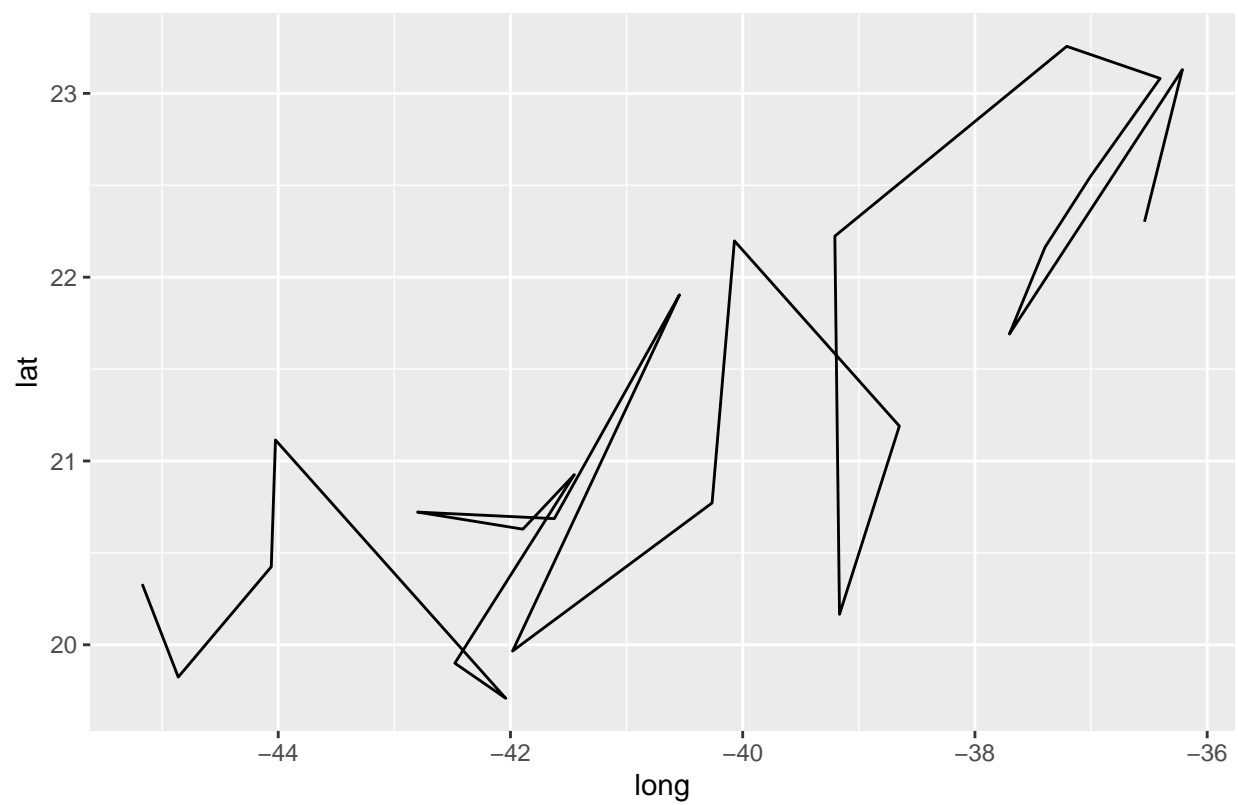
collar-data-F6-2016-02-26.txt

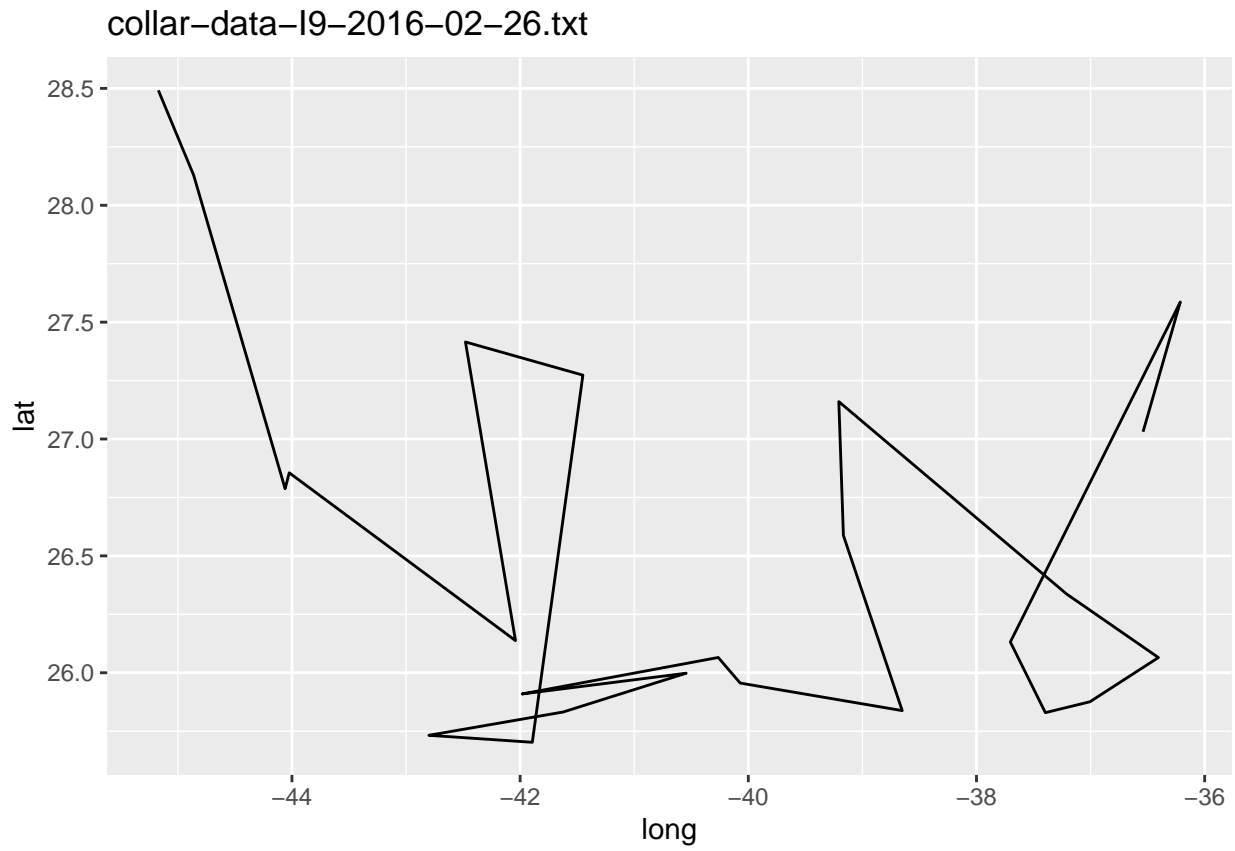


collar-data-G7-2016-02-26.txt

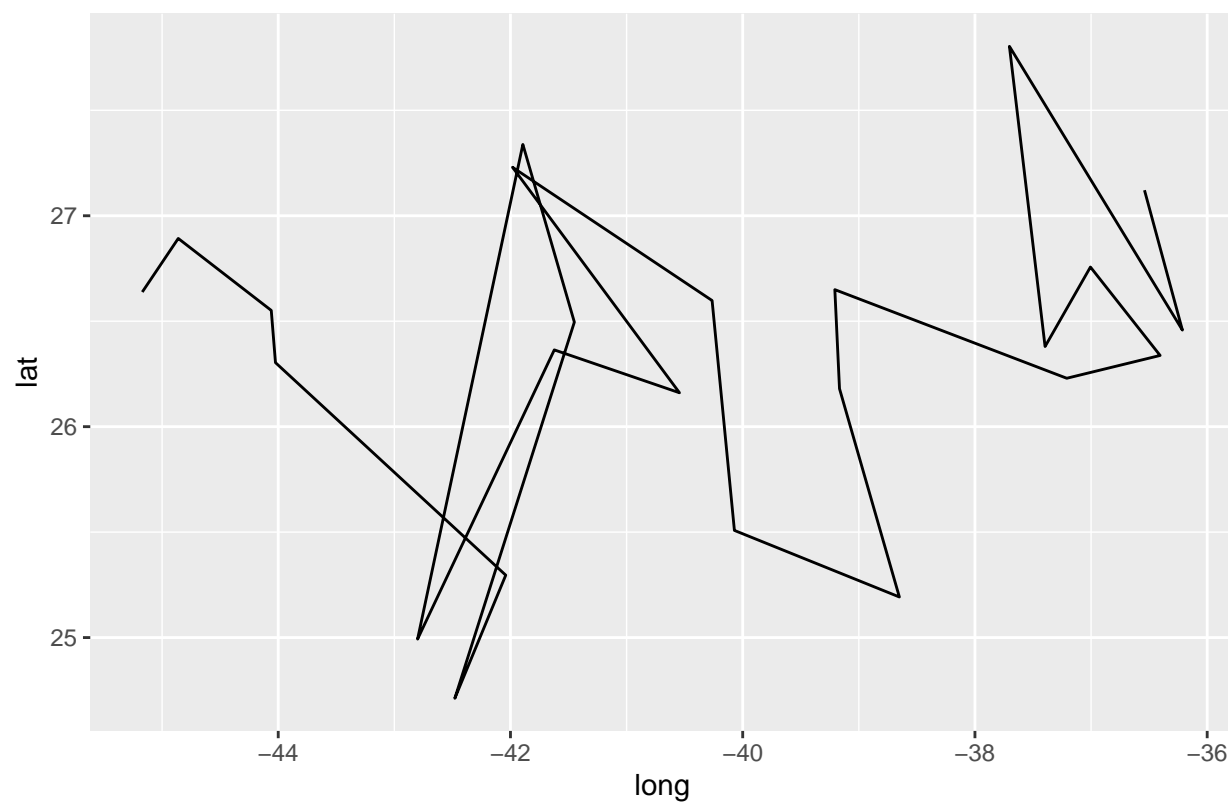


collar-data-H8-2016-02-26.txt





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



Use '!' to embed a plot ![title](address) Use knitr package to add fig\_chunk