

Data Visualization - Napoleon's March in 1812

Shakir Ahmed

2023-09-13 (YYYY-MM-DD)

Setup

Initialization of packages, and downloading additional packages needed for this homework assignment.

Theme set has also been set to the perfect standard.

Napoleon's march in 1812

This exercise is to recreate the Napoleon's march plot by Charles John Minard in ggplot2. The data is provided in package 'HistData', and this is the package from where we will be importing the data.

- Importing/reading the code using the rds() function after setting file path to 'data' folder

Now, we can recreate the plot of Napoleon's march in 1812. Parameters included are Latitude, Longitude, troop strength (size) and temperature.

```
#Troop data
data(Minard.troops, package="HistData")
str(Minard.troops)
```

```
## 'data.frame': 51 obs. of 5 variables:
## $ long      : num 24 24.5 25.5 26 27 28 28.5 29 30 30.3 ...
## $ lat       : num 54.9 55 54.5 54.7 54.8 54.9 55 55.1 55.2 55.3 ...
## $ survivors: int 340000 340000 340000 320000 300000 280000 240000 210000 180000 175000 ...
## $ direction: Factor w/ 2 levels "A","R": 1 1 1 1 1 1 1 1 1 1 ...
## $ group     : int 1 1 1 1 1 1 1 1 1 1 ...
```

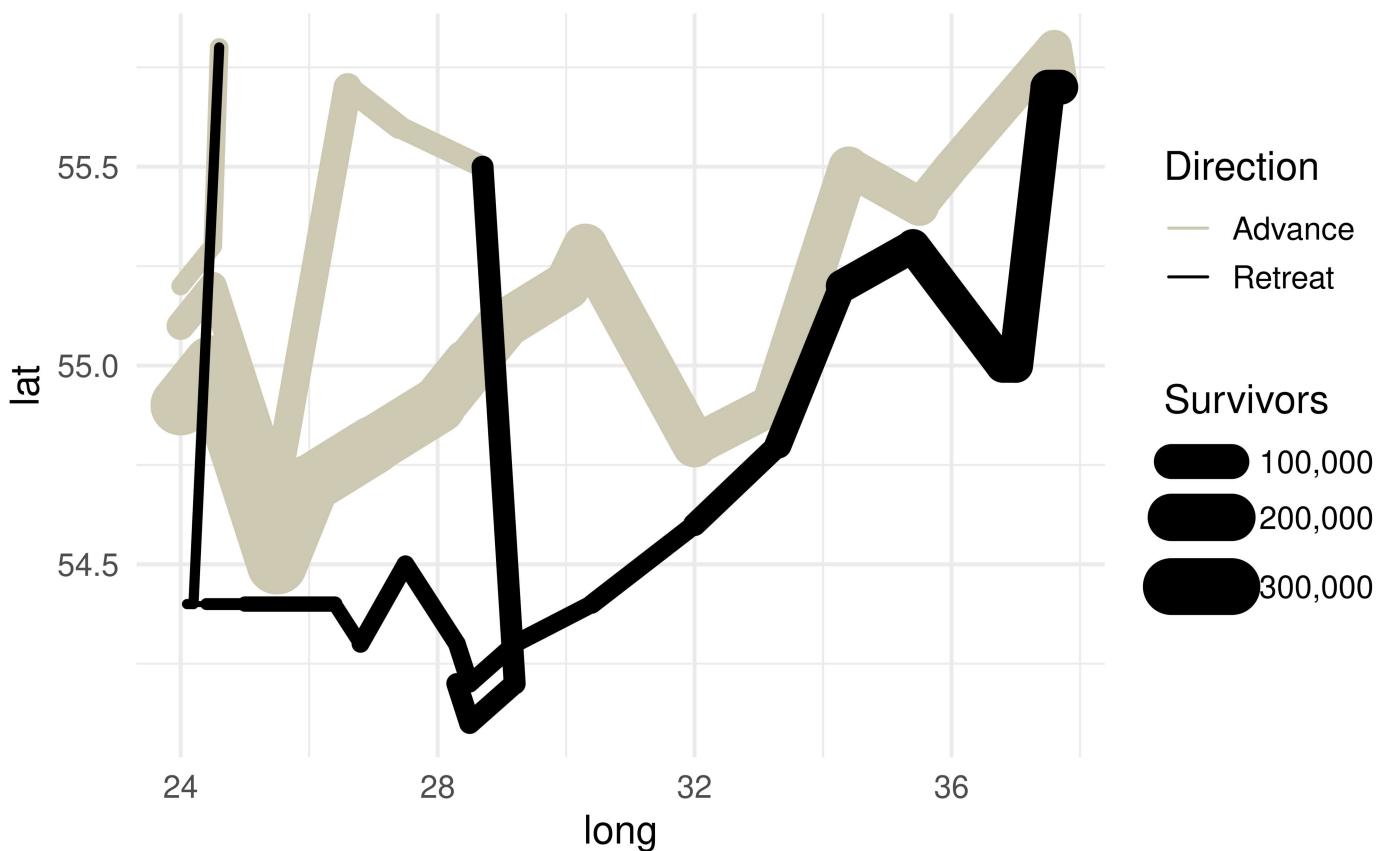
```
#Cities Data
data(Minard.cities, package="HistData")
str(Minard.cities)
```

```
## 'data.frame': 20 obs. of 3 variables:
## $ long: num 24 25.3 26.4 26.8 27.7 27.6 28.5 28.7 29.2 30.2 ...
## $ lat : num 55 54.7 54.4 54.3 55.2 53.9 54.3 55.5 54.4 55.3 ...
## $ city: Factor w/ 20 levels "Bobr","Chjat",...: 5 18 15 9 4 7 16 13 1 19 ...
```

```
#Temperature data
data(Minard.temp, package="HistData")
str(Minard.temp)
```

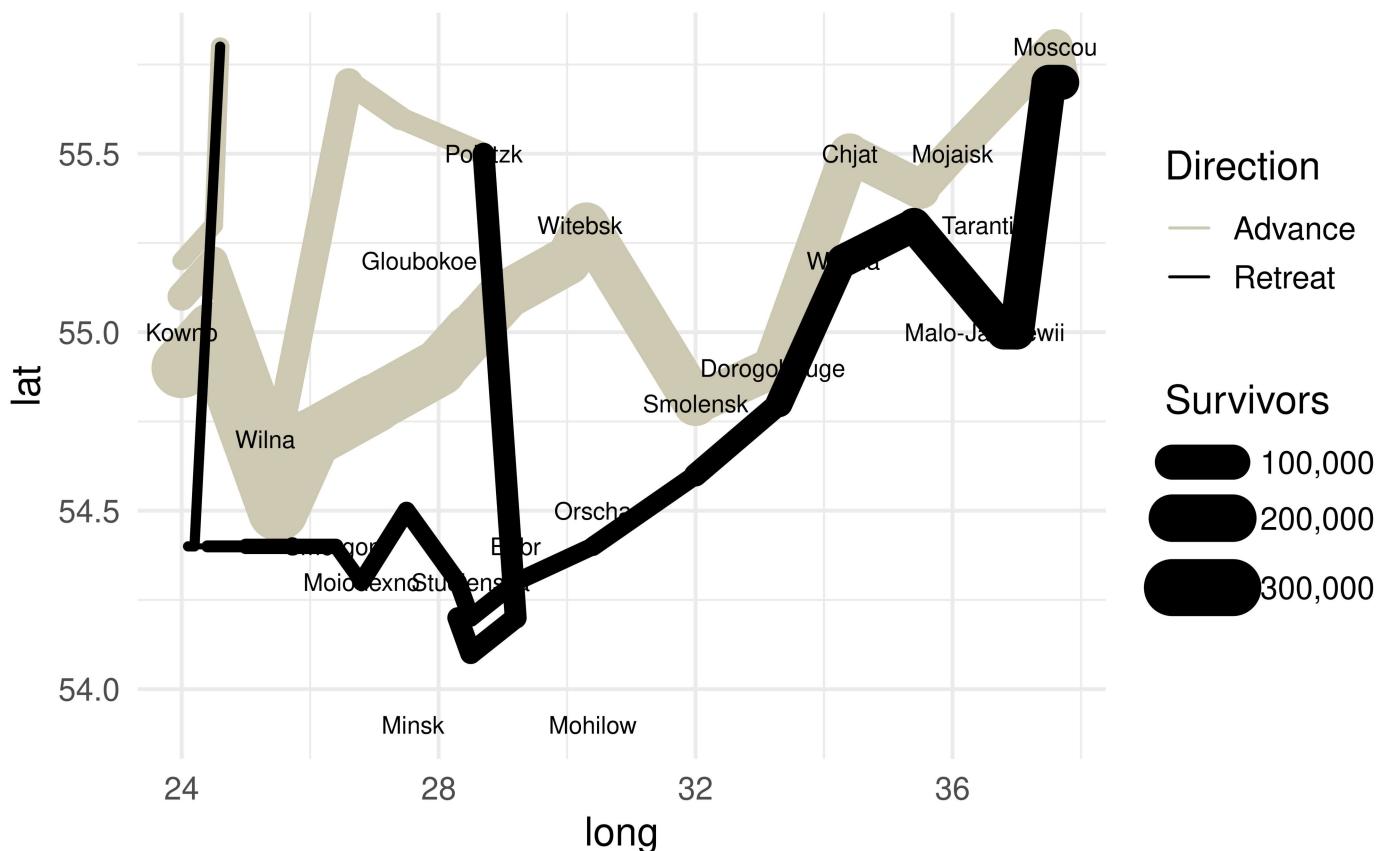
```
## 'data.frame': 9 obs. of 4 variables:
## $ long: num 37.6 36 33.2 32 29.2 28.5 27.2 26.7 25.3
## $ temp: int 0 0 -9 -21 -11 -20 -24 -30 -26
## $ days: int 6 6 16 5 10 4 3 5 1
## $ date: Factor w/ 8 levels "Dec01","Dec06",...: 7 8 4 5 NA 6 1 2 3
```

```
#Below plot is for plotting the Latitude and Longitude the troops travelled
#Width is set as the number of survivors to show the strength of the army
#Knitr function is used to scale the graph
breaks <- c(1, 2, 3) * 10^5 #New vector set is defined
ggplot(Minard.troops, aes(long, lat)) +
  geom_path(aes(size = survivors, colour = direction, group = group),
            lineend="round") +
  scale_size("Survivors", range = c(1,10), #c(0.5, 15),
             breaks=breaks, labels=scales::comma(breaks)) +
  scale_color_manual("Direction",
                     values = c("cornsilk3", "black"),
                     labels=c("Advance", "Retreat"))
```



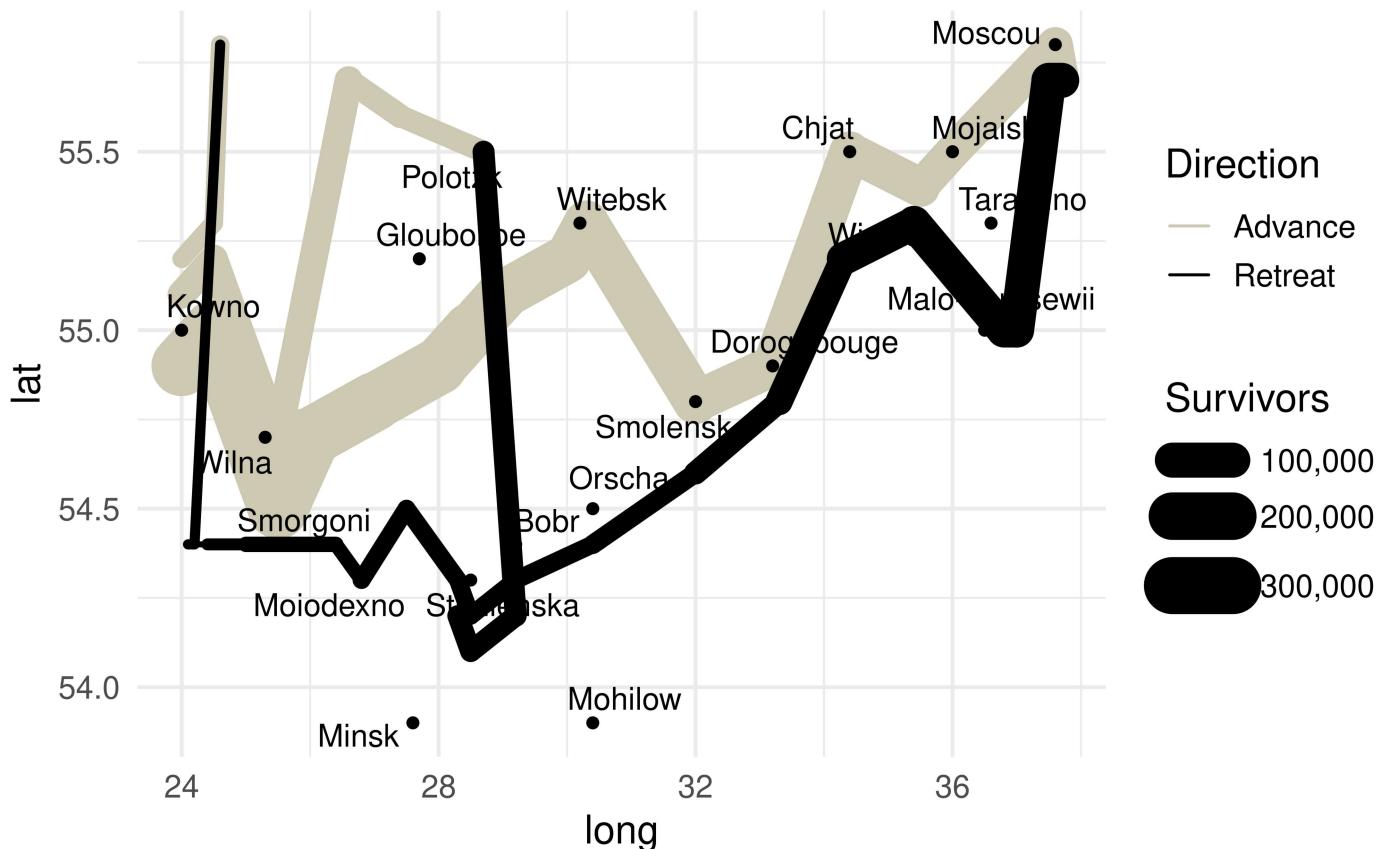
```
#ggplot trick to assign it to a variable to use later
plot_troops <- last_plot()

#Cities are now appended on top of the above plot
plot_troops + geom_text(data = Minard.cities, aes(label = city), size = 3)
```



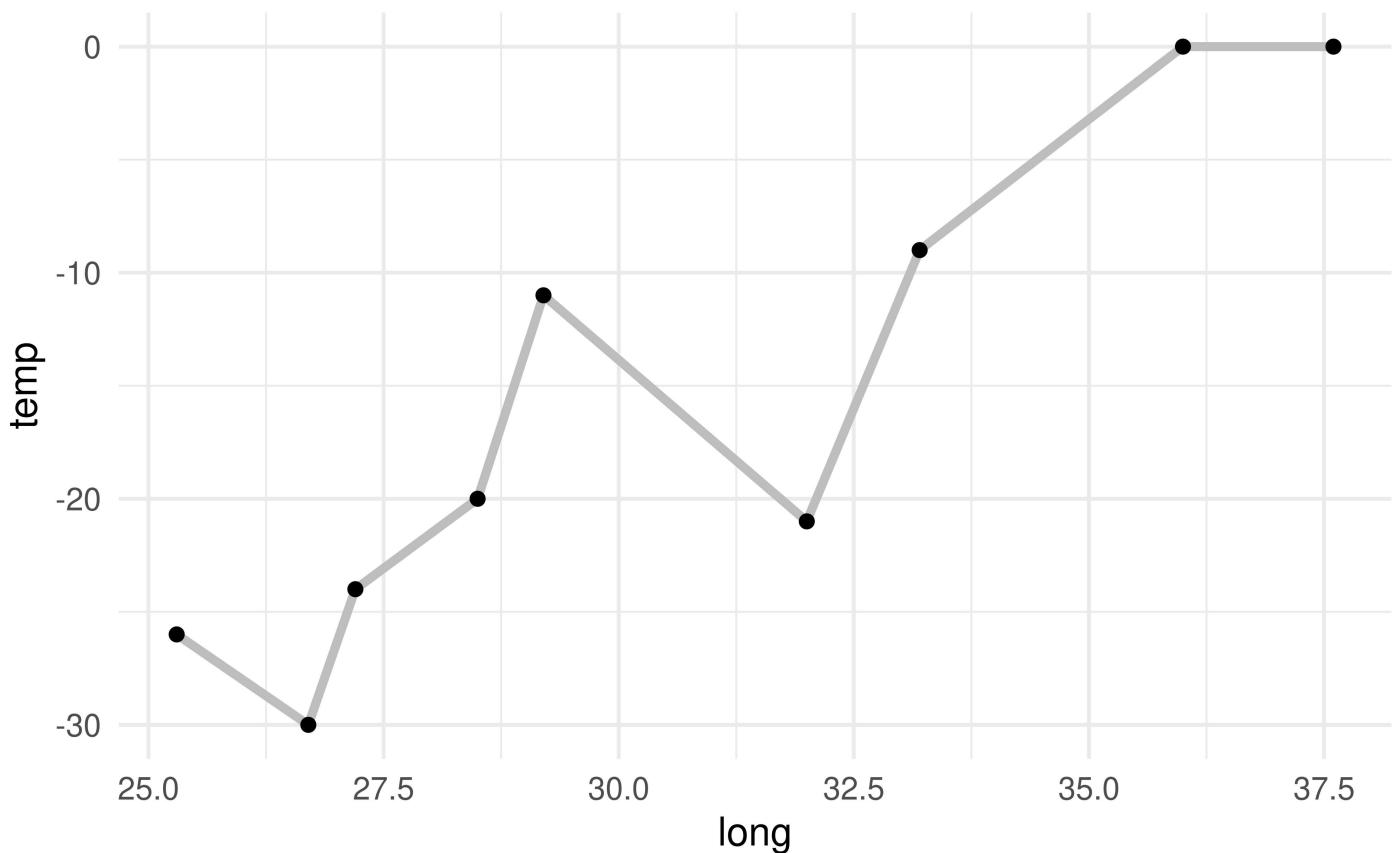
```
#Package ggrepel is used to automatically move the labels away from points and to ensure none of the labels overlap
```

```
if (!require(ggrepel)) {install.packages("ggrepel"); require(ggrepel)}
library(ggrepel)
plot_troops +
  geom_point(data = Minard.cities) +
  geom_text_repel(data = Minard.cities, aes(label = city))
```



```
#ggplot trick to assign it to a variable to use later
plot_troops_cities <- last_plot()

#Plotting the temperature with custom size
ggplot(Minard.temp, aes(long, temp)) +
  geom_path(color="grey", size=1.5) +
  geom_point(size=2)
```

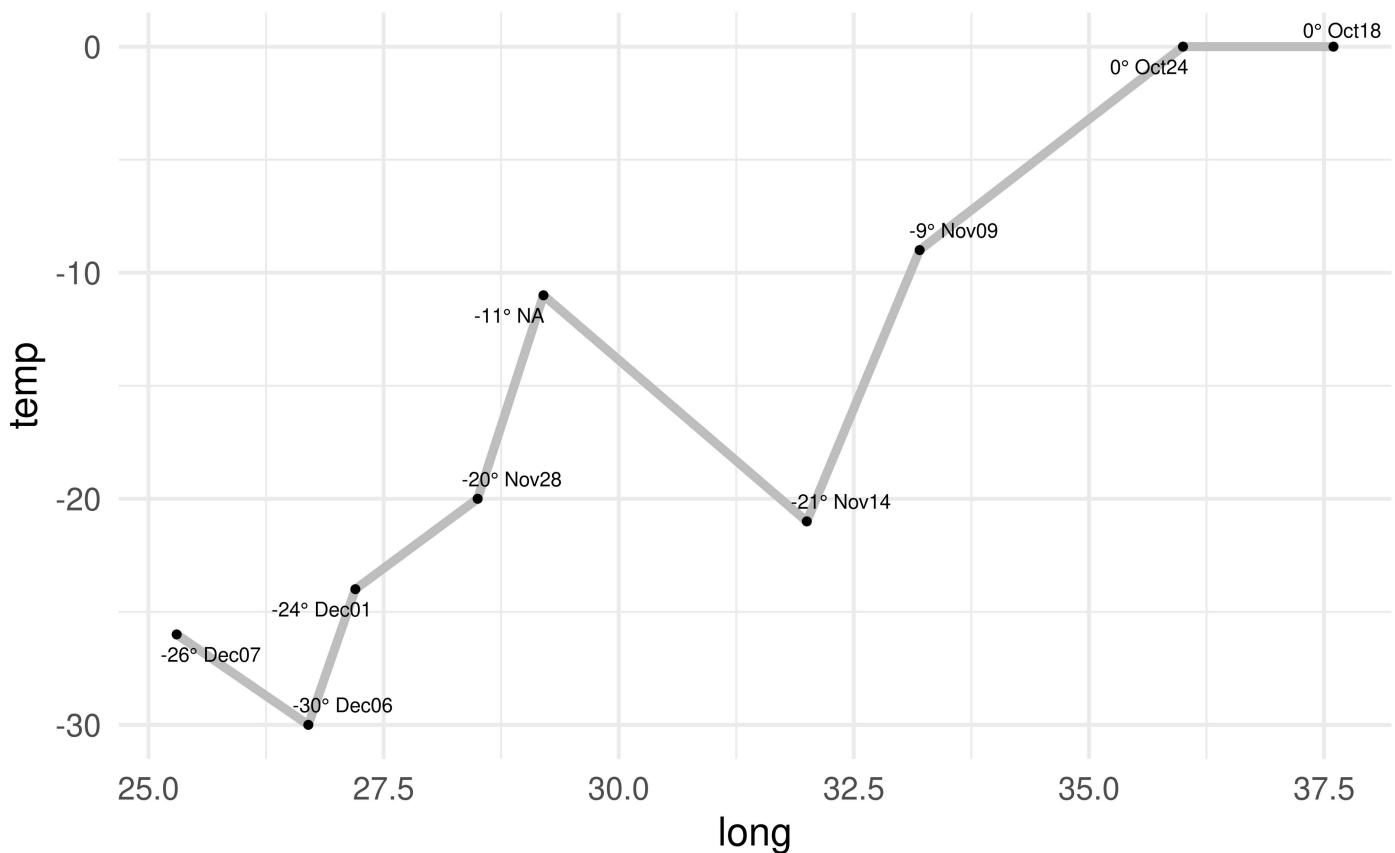


```
#Below code is to standardize the temperature format by combining temperature and date
Minard.temp <- Minard.temp %>%
  mutate(label = paste0(temp, "° ", date))
head(Minard.temp$label)
```

```
## [1] "0° Oct18"    "0° Oct24"    "-9° Nov09"   "-21° Nov14"
## [5] "-11° NA"     "-20° Nov28"
```

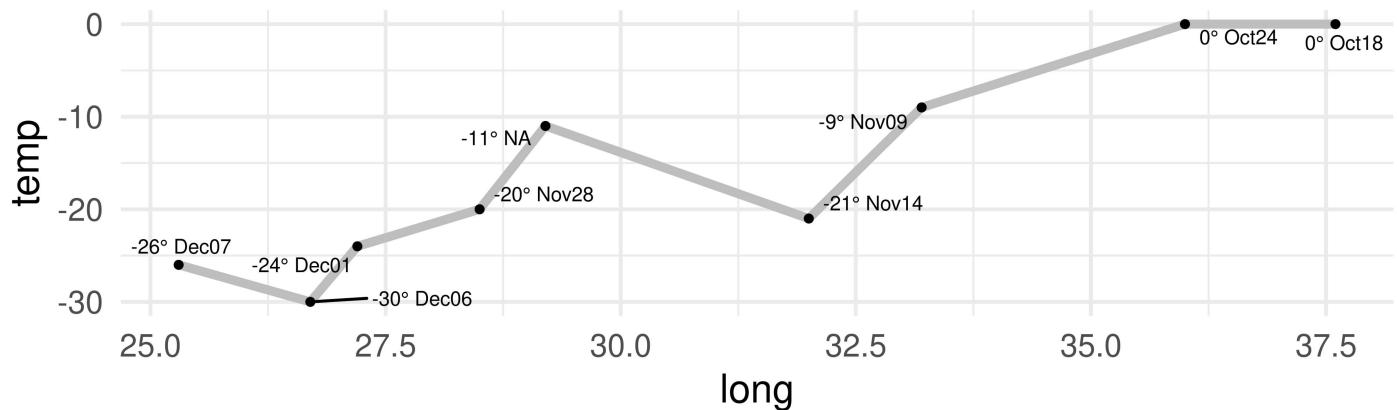
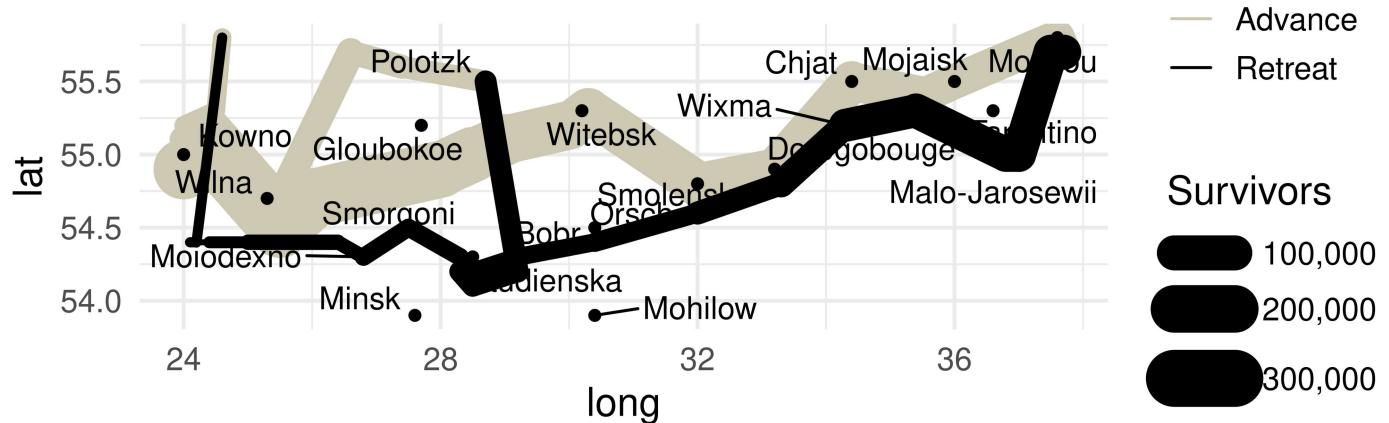
#These plots are pushing the right most two points outside the graph. Using geom_text_repel to bring back the data

```
ggplot(Minard.temp, aes(long, temp)) +
  geom_path(color="grey", size=1.5) +
  geom_point(size=1) +
  geom_text_repel(aes(label=label), size=2.5)
```



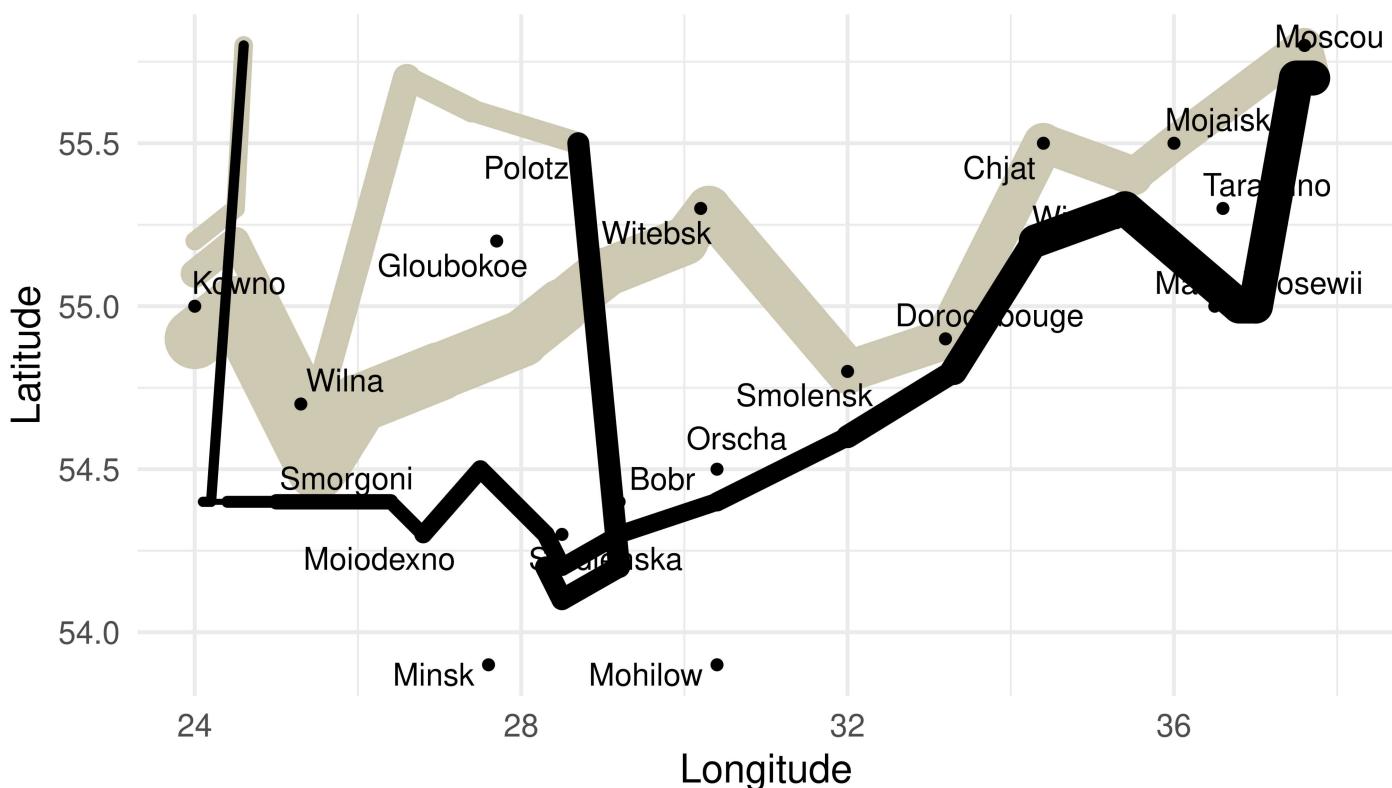
```
#ggplot trick to assign it to a variable to use later  
plot_temp <- last_plot()
```

```
#Now, we have to assemble the two graphs one below the other  
grid.arrange(plot_troops_cities, plot_temp)
```



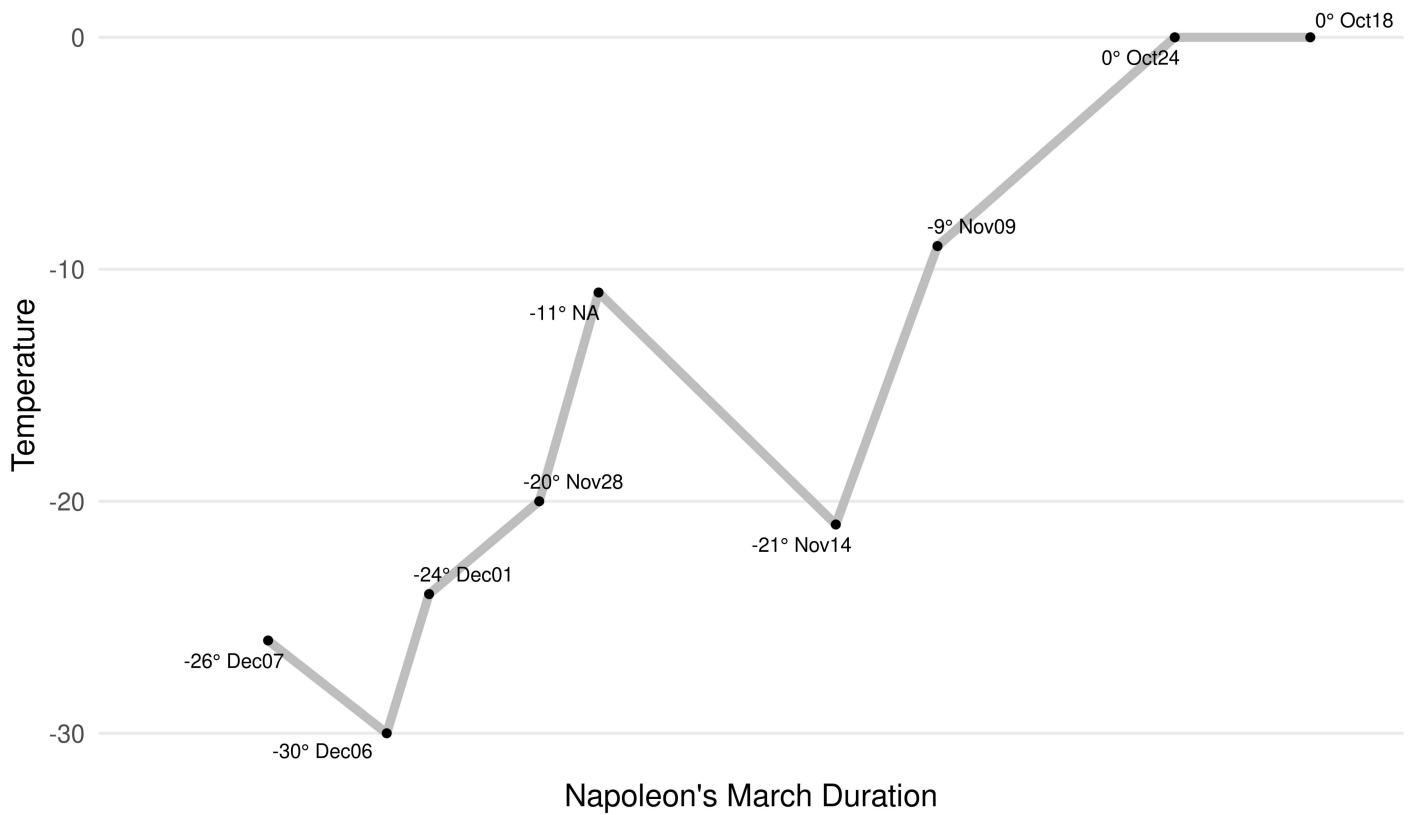
```
#In troops and cities plot, we have to set X axis limits to a range that will coincide with those for the plot of temperature
#Remove X and Y axis labels scales
#Remove Legends for survivors and directions
plot_troops_cities +
  coord_cartesian(xlim = c(24, 38)) +
  labs(x = "Longitude", y = "Latitude", title="Minard's version of Napoleon's March") +
  guides(color = FALSE, size = FALSE)
```

Minard's version of Napoleon's March



```
#ggplot trick to assign it to a variable to use later
plot_troops_cities_fixed <- last_plot()

#Setting theme for temperature graph
plot_temp +
  coord_cartesian(xlim = c(24, 38)) +
  labs(x="Napoleon's March Duration", y="Temperature",caption = "Michael Friendly (2018), Minard meets ggplot2") +
  theme_bw() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.grid.minor.y = element_blank(),
        axis.text.x = element_blank(), axis.ticks = element_blank(),
        panel.border = element_blank())
```



Michael Friendly (2018), Minard meets ggplot2

```
#ggplot trick to assign it to a variable to use later
plot_temp_fixed <- last_plot()

#Function to combine both the plots into one single plot and add a border
grid.arrange(plot_troops_cities_fixed, plot_temp_fixed, nrow=2, heights=c(3.5, 1.2))
grid.rect(width = .99, height = .99, gp = gpar(lwd = 2, col = "gray", fill = NA))
```

