Visualizing French snow research stations

A RStudio Demonstration and Statistical Analysis Draft

Jean-Marc Vincent, Lucas Leandro Nesi

1/11/2022

This document presents the fundamental aspects of RStudio markdown with data from French snow research stations.

## Useful shortcuts

This section keeps useful RStudio shortcuts

* Control + Alt + i : Open R chunk
* Control + Alt + c : Execute R chunk
* Control + Enter : Execute line

## Data acquisition

The data is available [here](https://donneespubliques.meteofrance.fr/?fond=produit&id_produit=94&id_rubrique=32). First, we want a simple dataset explaining the stations. It is available under the Documentation section “Liste des stations du réseau” Download the CSV file (This [link](https://donneespubliques.meteofrance.fr/donnees_libres/Txt/Nivo/postesNivo.csv)).

## Reading the data

We can use the function read\_csv from tidyverse:

library(tidyverse)  
data <- read\_csv("data/postesNivo.csv")

## Checking the data

We can print the data to see/understand all the columns:

data

## # A tibble: 129 × 5  
## Latitude Longitude ID Altitude Nom   
## <dbl> <dbl> <chr> <dbl> <chr>   
## 1 46.8 6.36 07392 1036 METABIEF\_STATION   
## 2 46.0 6.97 07393 2196 LE TOUR BALME   
## 3 46.3 6.71 07454 1535 Bernex   
## 4 46.3 6.67 07457 790 VACHERESSE AUXI   
## 5 45.2 6.20 07466 2090 ST SORLIN D'ARVES   
## 6 45.9 6.89 07467 2310 PLAN AIGUILLE TMB   
## 7 46.0 6.89 07470 1850 LA FLEGERE   
## 8 45.4 6.98 07498 1854 Val d Isere Joseray  
## 9 45.9 6.87 07499 1025 CHAMONIX-OBS   
## 10 45.4 7.06 07566 1830 Bonneval sur Arc   
## # ℹ 119 more rows

So the data has 5 columns:

* Latitude : The latitude of the Station
* Longitude : The longitude of the Station
* ID : A identifier for cross-reference between other datasets (This is important for the real snow data set!)
* Altitude : The altitude of the Station
* Nom : The name of the Station

We can select a column (for example, Altitude) using the following syntax:

## Basic statistics on this dataset

We can compute basic statistics on this dataset, for example, we can get a summary of the altitude column:

summary(data$Altitude)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 790 1472 1713 1729 1920 3200

Summary computes the same thing as:

min(data$Altitude)

## [1] 790

quantile(data$Altitude, 0.25)

## 25%   
## 1472

median(data$Altitude)

## [1] 1713

mean(data$Altitude)

## [1] 1728.574

quantile(data$Altitude, 0.75)

## 75%   
## 1920

max(data$Altitude)

## [1] 3200

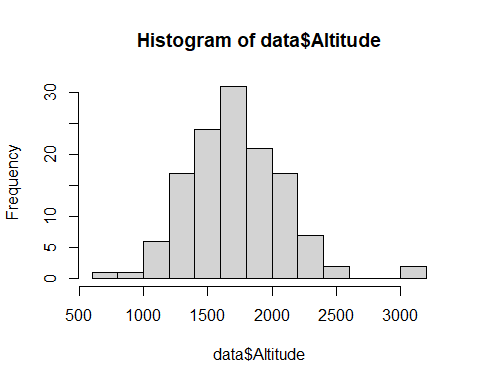
## Simple visualizations

We can create simple visualizations with this data.

### Histogram

We can generate a simple histogram of the altitude with function hist:

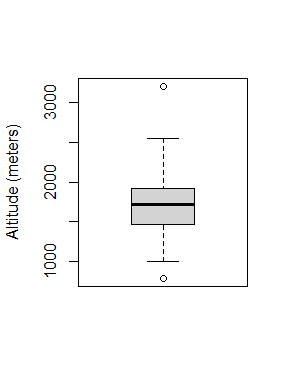
hist(data$Altitude)



### Boxplot

Another possibility is the boxplot, we can use the function boxplot:

boxplot(data$Altitude, ylab="Altitude (meters)")



## A Map plot of the data

We can use more sophisticated libraries to plot the locations of the stations on a map. We will use the packages ggplot and maps. *Do not worry* about ggplot functions; they will be covered in the following classes. I want to show how useful Literate Programming is and some tricks in RStudio.

First we need to install a extra library for generating a map:

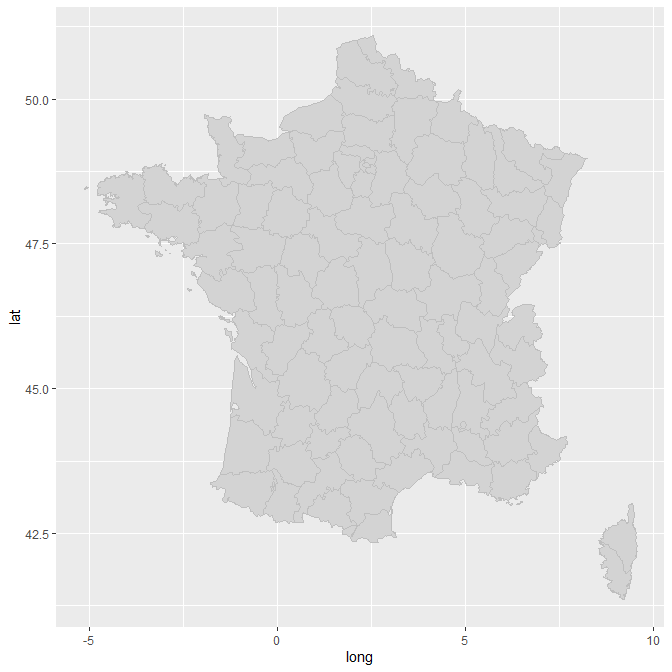
if("maps" %in% rownames(installed.packages()) == FALSE){  
install.packages("maps", repos = "https://cloud.r-project.org/")  
}

It is then necessary to select the map of France using the function map\_data of maps:

france\_map <- map\_data("france")

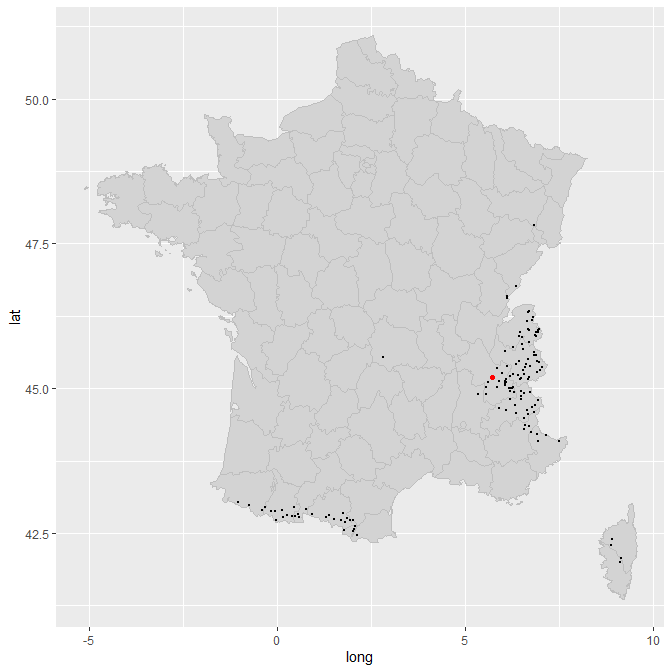
And we can plot the map of France with all regions:

ggplot(france\_map, aes(x = long, y = lat), height=10) +  
 geom\_polygon(aes(group = group), fill="lightgray", color="gray")



We can then add the stations of our other dataset, plotting them as black points. Also, we can add Grenoble Latitude (45.1885) and Longitude (5.7245) for a reference and plot it as a red point.

grenoble\_data <- data.frame(Longitude=c(5.7245), Latitude=c(45.1885), Name=c("Grenoble"))  
ggplot(france\_map, aes(x = long, y = lat), height=10) +  
 geom\_polygon(aes(group = group), fill="lightgray", color="gray") +  
 geom\_point(data=data, aes(x = Longitude, y=Latitude), size=0.5) +  
 geom\_point(data=grenoble\_data, aes(x=Longitude, y=Latitude), color="red")



Because there are some many stations, we can zoom a part of the map, the region near Grenoble, and add the names of the stations:

grenoble\_data <- data.frame(Longitude=c(5.7245), Latitude=c(45.1885), Name=c("Grenoble"))  
ggplot(france\_map, aes(x = long, y = lat), height=10) +  
 geom\_polygon(aes(group = group), fill="lightgray", color="gray") +  
 geom\_point(data=data, aes(x = Longitude, y=Latitude), size=1) +  
 geom\_point(data=grenoble\_data, aes(x=Longitude, y=Latitude), color="red") +  
 coord\_cartesian(xlim = c(5.2, 6.6), ylim=c(44.8, 45.5)) +  
 geom\_text(data=data, aes(x = Longitude, y=Latitude+0.01, label=Nom), size=3) +  
 geom\_text(data=grenoble\_data, aes(x = Longitude, y=Latitude+0.01, label=Name), color="red", size=3)

