PS3: Scrape and plot data

Pooja Sadarangani

2022-10-28

1.1 Ethical issues

1. Consult wikipedia terms of usage. Does it put restrictions on web scraping?

There are no explicit restrictions on web scraping in wikipedia's terms of usage. However the scrapers must keep the follwing points in mind while scraping wiki pages:

"Disrupting the services by placing an undue burden on a Project website or the networks or servers connected with a Project website;"

"Disrupting the services by inundating any of the Project websites with communications or other traffic that suggests no serious intent to use the Project website for its stated purpose;"

2. Consult robots.txt. Is it permitted to scrape wiki-pages?

```
##  en.wikipedia.org
```

```
## [1] TRUE
```

According to the robots.txt, we have the permission to scrape wiki-pages but we are expected to to do it responsibly.

3. Describe what do you do in order to reduce the burden to wikipedia website.

I can do the followings things to reduce the burden to wikipedia website:

1. **Download the webpages once first, and download more only if required**
2. **Use cached version for developing and deploying**
3. **Limit query requests to what the server can handle**
4. **Do not download the pages that we are not allowed to scrape after consulting to robots.txt**
5. **Donate money to wikipedia**

1.2 Parse the list of mountains

1. Load the wikipedia list of mountains by height

2. Find all the tables there in the html.

```
## [1] 9
```

3. Find the table headers, and determine which columns are mountain names, heights, and where are the links to the individual mountain pages.

```
##
```

```
## 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

## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v stringr 1.4.1
## v tidyr   1.2.1     v forcats 0.5.2
## v readr   2.1.3
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()         masks stats::filter()
## x readr::guess_encoding() masks rvest::guess_encoding()
## x dplyr::lag()            masks stats::lag()

##  [1] "Mountain"             "Metres"               "Feet"
##  [4] "Range"                "Location and Notes\n" "Mountain"
##  [7] "Metres"               "Feet"                 "Range"
## [10] "Location and Notes\n" "Mountain"             "Metres"
## [13] "Feet"                 "Location and Notes\n" "Mountain"
## [16] "Metres"               "Feet"                 "Range"
## [19] "Location and Notes\n" "Mountain"             "Metres"
## [22] "Feet"                 "Location and Notes\n" "Mountain"
## [25] "Metres"               "Feet"                 "Location and Notes\n"
## [28] "Mountain"             "Metres"               "Feet"
## [31] "Location and Notes\n" "Mountain"             "Metres"
## [34] "Feet"                 "Location and Notes\n" "Mountain"
## [37] "Metres"               "Feet"                 "Range"
## [40] "Location and Notes\n"

## [[1]]
## # A tibble: 14 x 5
##    Mountain                       Metres Feet   Range     `Location and Notes`
##    <chr>                          <chr>  <chr>  <chr>     <chr>
##  1 Mount Everest                  8,848  29,029 Himalayas "Nepal/China"
##  2 K2                             8,612  28,255 Karakoram "Pakistan/China"
##  3 Kangchenjunga                  8,586  28,169 Himalayas "Nepal/India"
##  4 Lhotse                         8,516  27,940 Himalayas "Nepal - Climbers asc~
##  5 Makalu                         8,485  27,838 Himalayas "Nepal"
##  6 Cho Oyu                        8,188  26,864 Himalayas "Nepal - Considered \~
##  7 Dhaulagiri                     8,167  26,795 Himalayas "Nepal - Presumed wor~
##  8 Manaslu                        8,163  26,781 Himalayas "Nepal"
##  9 Nanga Parbat                   8,126  26,660 Himalayas "Pakistan"
## 10 Annapurna                      8,091  26,545 Himalayas "Nepal - First eight-~
## 11 Gasherbrum I (Hidden peak; K5) 8,080  26,509 Karakoram "Pakistan/China - Ori~
## 12 Broad Peak                     8,051  26,414 Karakoram "Pakistan/China"
## 13 Gasherbrum II (K4)             8,035  26,362 Karakoram "Pakistan/China - Ori~
## 14 Shishapangma                   8,027  26,335 Himalayas "China"
```

From the above table structure, we can tell that montain names are in column 1 and height is in column 2. By looking at the table structure, we are unable to tell where the links are stored. However,from the html_code we can tell that the links are present in the td element -> a element -> attribute href

4. Create a data frame that contains names and heights of the mountains above 6800m, and the links to the corresponding wikipedia pages. You'll add longitude and latitude for each mountain in this data frame later.

```
## The number of rows after filtering mountains having height greater than 6800m 175
```

5. Print a small sample of your data frame to see that it looks reasonable.

```
##   Mountain_name Heights                                     Links
## 1 Mount Everest    8848 https://en.wikipedia.org/wiki/Mount_Everest
## 2            K2    8612            https://en.wikipedia.org/wiki/K2
## 3 Kangchenjunga    8586 https://en.wikipedia.org/wiki/Kangchenjunga
## 4        Lhotse    8516        https://en.wikipedia.org/wiki/Lhotse
## 5        Makalu    8485        https://en.wikipedia.org/wiki/Makalu
## 6       Cho Oyu    8188       https://en.wikipedia.org/wiki/Cho_Oyu
```

1.3 Scrape the individual mountain data

1. Write a function that converts the longitude/latitude string to degrees (positive and negative)

2. Write another function that takes link as an argument and loads the mountain's html page and extracts latitude and longitude.

3. loop over the table of mountains you did above, download the mountain data, and extract the coordinates. Store these into the same data frame.

```
## [1] "Mountain_name" "Heights"       "Links"         "latitude"
## [5] "longitude"
```

Print a sample of the dataframe and check that it looks good. How many mountains did you get?

```
##   Mountain_name Heights                                     Links latitude
## 1 Mount Everest    8848 https://en.wikipedia.org/wiki/Mount_Everest 27.98806
## 2            K2    8612            https://en.wikipedia.org/wiki/K2 35.88250
## 3 Kangchenjunga    8586 https://en.wikipedia.org/wiki/Kangchenjunga 27.70250
## 4        Lhotse    8516        https://en.wikipedia.org/wiki/Lhotse 27.96167
## 5        Makalu    8485        https://en.wikipedia.org/wiki/Makalu 27.88972
## 6       Cho Oyu    8188       https://en.wikipedia.org/wiki/Cho_Oyu 28.09417
##   longitude
## 1  86.92528
## 2  76.51333
## 3  88.14667
## 4  86.93333
## 5  87.08889
## 6  86.66083
```

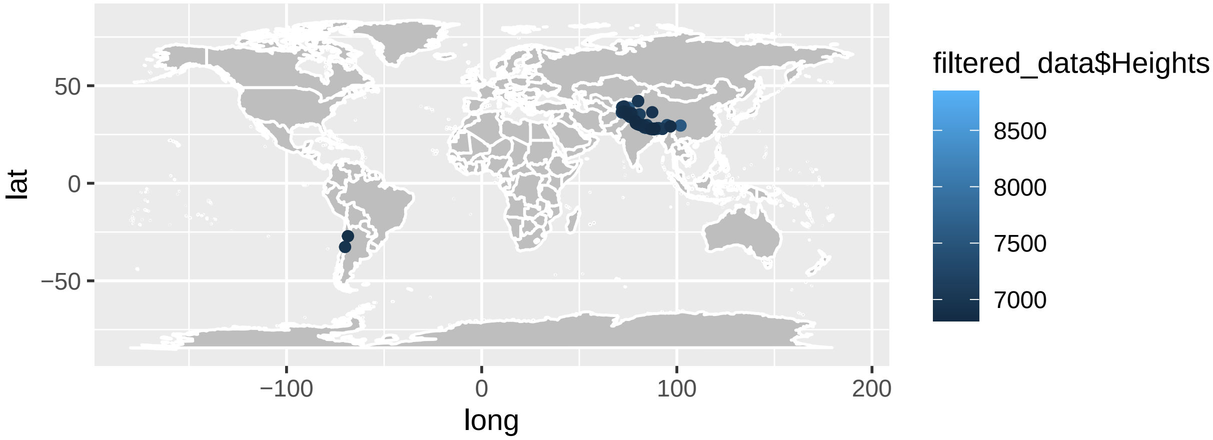
```
## The number of mountains that I got are after the removal of mountains having invalid links:  165
```

1.4 Plot the mountains

1. Plot all the mountains on a world map. Color those according to their height.

```
##
## Attaching package: 'maps'

## The following object is masked from 'package:purrr':
##
##     map
```



2. Describe what did you get. Where are the tall mountains located? Do all the locations make sense (i.e. you do not have mountains in the middle of sea and such)?

From looking at the graph, I can tell that there are 2 mountains situated in South Amaerica, rest all tall mountains are in Asia. All locations make sense as they all seem to be on land and not in sea.