

Lecture 8: Web Scraping

World Heritage Sites in Danger

Isabel Casas
icasas@deusto.es

Before everything

- Log into your computer and open R Studio
- Open an R script where you can type the commands we are seeing in class
- This will help you to do the activities suggested in class

Web scraping

- **Web scraping** is the process of automatically extracting data from websites using software tools.
- This data can then be saved and analyzed for various purposes such as data mining, research, or business intelligence.
- The process involves sending a request to a website, extracting data from the website HTML code, and parsing that data into a structured format.
- The example in this class is based on @Munzertetal2015.

Examples of Web Scraping

- Price comparison: extract prices for a particular product from different websites.
- Job search: extract job postings from various job boards and aggregators.
- Lead generation: extract contact information from websites. This information can be used to generate leads for sales and marketing purposes.
- News aggregation: extract news articles from various websites and consolidate them into a single source. This information can be used to stay up-to-date on current events.
- Social media monitoring: extract data from social media platforms such as Twitter or Facebook.

Example: Web Scraping the Wikipedia

- The *UNESCO* is an organisation within the UN which, among other things, fights for the preservation of the world natural and cultural heritage.
- They provide a [list](#) of sites that have been declared of importance to the world for their cultural, natural or mixed value.
- Sites with a red bullet are considered in danger. The definition of danger in this context appears in this [link](#):
- The List of [World Heritage in Danger](#) is designed to inform the international community of conditions which threaten the very characteristics and to encourage corrective action.

Analysing World Heritage in Danger

Important questions to answer are:

- ❶ Which sites are threatened and where are located?
- ❷ Are some regions in the world where sites are more endangered than others?
- ❸ What are the reasons that put a site at risk?
 - Wikipedia dedicates a page to [currently and previously endangered sites](#).
 - Our aim is to use the data from Wikipedia to answer the questions above. We will also plot a world map with the endangered sites.

Activity 1

- 1 Go to the Wikipedia website and check the table with the endangered sites.
- 2 Mouse right click “source code”. That is how an HTML website looks like. The table starts with the HTML command:

```
<table class="wikitable plainrowheaders sortable">
```

Find it. What commands does finish that table?

Some Web Scraping Packages

We will need the following packages, so install them in your computer if you have not done before. The upload them to memory.

```
#packages = c("RCurl", "XML", "maps", "stringr")  
#install.packages(packages, dependencies = TRUE)  
library(RCurl)  
library(XML)  
library(maps)  
library(stringr)
```


Read Table from URL

- The R chunk below, reads all the lines from the Wikipedia website we want.
- Functions in package *XML* know how to interpret HTML elements, for example tables or lists from a website. This works via a so called parser, in this case in function *htmlParse()*.
- R extracts all HTML tables in *parsed_wiki* and copy then to a variable called *tables*.

```
url.name <- "https://en.wikipedia.org/wiki/List_of_World_Heritage_in_Danger"
wiki_read <- readLines(url.name, encoding = "UTF-8")
parsed_wiki <- htmlParse(wiki_read, encoding = "UTF-8")
tables <- XML::readHTMLTable(parsed_wiki, stringsAsFactors= FALSE)
names(tables)
```

```
## [1] "NULL" "NULL" "NULL"
```

- R has converted HTML code into a list with names GeoGroup, NULL, GeoGroup, NULL and NULL.

Activity 2

As you have seen there are a few tables in this website. Check what is inside of each element in the list using `tables[[number]]`?

Which table are we interested in ?

write you code here

Read Info from Table

- We are interested in the second table, which we copy to *danger_table*
- We see that the first row of *danger_table* has the names of the table columns, so we use them to name the columns of our table and then remove that first row.

```
danger_table <- tables[[2]]  
danger_table[1,]
```

```
##      V1      V2      V3      V4      V5      V6  
## 1 Name Image Location Criteria Areaaha (acre) Year (WHS) End
```

```
names(danger_table) <- danger_table[1,]  
danger_table <- danger_table[-1,]  
names(danger_table)
```

```
## [1] "Name"          "Image"          "Location"       "Criteria"  
## [5] "Areaaha (acre)" "Year (WHS)"     "Endangered"     "Reason"  
## [9] "Refs"
```

```
names(danger_table)[c(5, 6)] <- c("Acre", "Year")
```

Activity 3

- How many columns have the Wikipedia table when we see it in our browser?
- Do they correspond to the columns in *danger_table*?

Read Info from Table

We are interested in variables in columns 3, 4, 6 and 7 of the table:

- **Location** which has the coordinates of the sites
- **Criteria** defining if the site has a cultural or natural value
- **Year**, year in which the site was included in the UNESCO heritage list
- **Endangered** which is the year in which the site was declared endangered in the hope that something is done to protect it and sending the signal that it could be remove from the UNESCO heritage list if not.

```
danger_table$Name[1:3]
```

```
## [1] "Angkor"  
## [2] "Bagrati Cathedral and Gelati Monastery"  
## [3] "Bahla Fort"
```

Activity 4

```
# Get info from the 10th endangered site from danger_table  
# (code here)  
  
# Get info from last endangered site from danger_table  
# (code here)
```

Let us see what is inside *danger_table*:

```
str(danger_table)
```

We need to recode variable *Criteria* to avoid problems with strange characters.

```
danger_table$Criteria[1:3]
```

```
## [1] "Cultural:(i), (ii), (iii), (iv)" "Cultural:(iv)"  
## [3] "Cultural:(iv)"
```

```
danger_table$Criteria<- ifelse(str_detect(danger_table$Criteria, "Natural") ==  
TRUE, "nat", "cult")  
danger_table$Criteria[1:3]
```

```
## [1] "cult" "cult" "cult"
```


Also, we want to convert columns *Year* and *Endangered* into numbers.

```
danger_table$Year[1:3]
```

```
## [1] "1992" "1994" "1987"
```

```
danger_table$Year <- as.numeric(danger_table$Year)  
danger_table$Year[1:3]
```

```
## [1] 1992 1994 1987
```

Data Cleansing

- Column “Endangered” is ambiguous: it has a dash at the end of each number which we want to remove.
- To do so, we specify a so-called regular expression “[^[:alnum:]]”. Some entries contain several years, we want to keep the last 4 digits. We use functions *str_sub* and *str_replace_all*.

```
library("stringr")
danger_table$Endangered[1:3]
```

```
## [1] "1992-2004" "2010-2017" "1988-2004"
```

```
#replace non-alphanumeric characters (the dash) for a blank
```

```
danger_table$Endangered<- as.numeric(str_replace_all(danger_table$Endangered,"[^[:alnum:]]", ""))
```

```
#pick the last four digits of each number
```

```
danger_table$Endangered <- as.numeric(str_sub(danger_table$Endangered,-4,-1))
```

```
danger_table$Endangered[1:3]
```

```
## [1] 2004 2017 2004
```

Variable *Location* is quite complicated. It contains coordinates in different formats, as well as the name of the place.

```
danger_table$Location[1:3]
```

Data Cleansing

- For a map, we need the latitude (30.84167N) and longitude (29.66389E).
- To extract this information, we use regular expressions again. We are not going to go in detail about them in this class.

```
reg_y <- "[/][ -]*[[:digit:]]*[.]*[[:digit:]]*[*;]"
reg_x <- "[;][ -]*[[:digit:]]*[.]*[[:digit:]]*"
y_coords <- str_extract(danger_table$Location, reg_y)
y_coords <- as.numeric(str_sub(y_coords, 3, -2))
danger_table$y_coords <- y_coords
x_coords <- str_extract(danger_table$Location, reg_x)
x_coords <- as.numeric(str_sub(x_coords, 3, -1))
danger_table$x_coords <- x_coords
round(danger_table$y_coords, 2)[1:3]
```

```
## [1] 13.43 42.26 22.97
```

```
dim(danger_table)
```

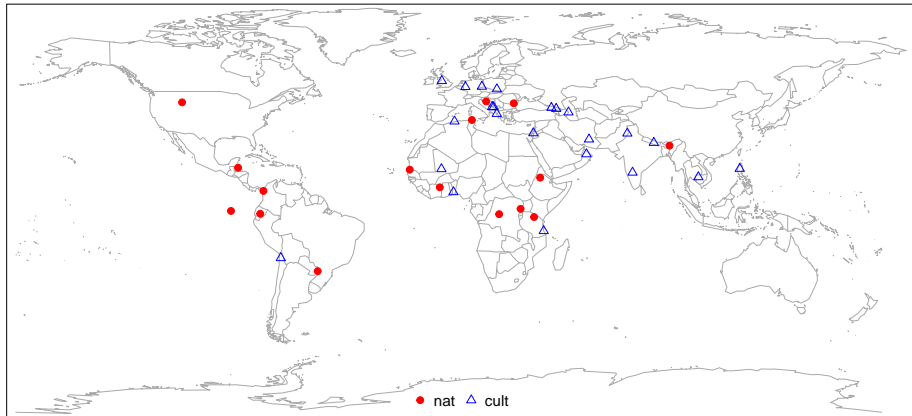
```
## [1] 39 11
```

Mapping Data

- We have cleaned the data and we have now a table with information on 57 sites.
- Let us plot the sites in a world map using package *maps*.
- The first line in the chunk below assigns a circle (19) to endangered sites due to natural causes and a triangle (2) to cultural causes.

```
pch <- ifelse(danger_table$Criteria == "nat", 19, 2)
col <- ifelse(danger_table$Criteria == "nat", "red", "blue")
map("world", col = "darkgrey", lwd = 0.5,
    mar = c(0.1, 0.1, 0.1, 0.1))
points(danger_table$x_coords, danger_table$y_coords,
       pch = pch, col = col)
legend("bottom", c("nat", "cult"), pch=c(19, 2),
       col=c("red", "blue"), bty="n", ncol=2)
box()
```

Mapping Data



- Endangered cultural heritage sites (triangle) are clustered in the Middle East and Southwest Asia.
- Natural heritage sites in danger (circles) are more prominent in Africa

Activity 5

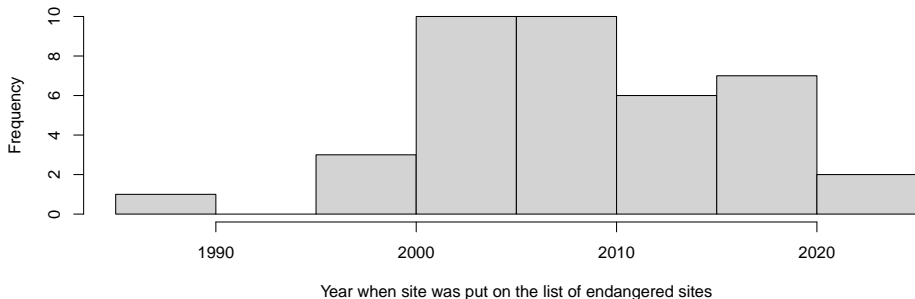
Plot the map again but now assign different symbols and colours to the sites.

- We do not have a lot of information in the table to take conclusions on the reasons for these endangered sites.
- There is a variable “Reason” on the table and we could decide to find certain words or do a cloud map for it (we will learn about it next week).
- Risk of cultural sites could be due to some political or economic unrest.
- Risk of natural sites could be due to some environmental conditions.

Text mining

- In the figure below, we can see that the frequency of sites put on the “red list” has increased over the years until year 2015, but it has decreased afterwards. What are the reasons for this?

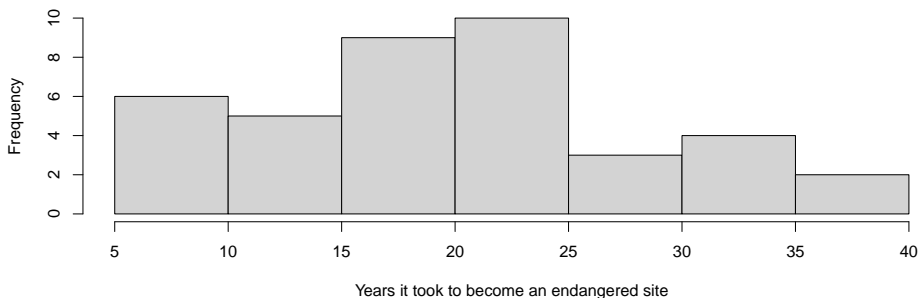
```
hist(danger_table$Endangered, freq = TRUE,  
     xlab = "Year when site was put on the list of endangered sites", main
```



- How long it took for sites to be declared endangered from the time they were enlisted in the UNESCO list?
- Many of the sites were put on the red list only shortly after their designation as world heritage?
- According to the official selection criteria for becoming a cultural or natural heritage, it is not a necessary condition to be endangered.
- In contrast, endangered sites run the risk of losing their status as world heritage. So why do they become part of the List of World Heritage Sites when it is likely that the site may soon run the risk of losing it again?
- One could speculate that the committee may be well aware of these facts and might use the list as a political means to enforce protection of the sites.

Text mining

```
duration <- danger_table$Endangered - danger_table$Year  
hist(duration, freq = TRUE,  
xlab = "Years it took to become an endangered site", main = "")
```



Using only few lines of code, we have enriched the data and gathered new insights, which might not have been obvious from examining the table alone.

Is web scraping legal?

- Things have changed a lot in the last years. Web scraping used to be very common a decade ago, but now there are many companies like Amazon or Ryanair who are protecting against web scraping.
- Instead, they are providing APIs which are functions to get data from their databases, but functions that control what data is being downloaded.

Is web scraping legal?

- There have been many cases in the tribunals about web scraping. Among them, the [US Supreme Court case Feist Publications vs Rural Telephone Service](#) established that scraping and republishing facts like telephone listings is allowed.
- A similar case in [Australia Telstra vs Phone Directories](#) concluded that data can not be copyrighted if there is no identifiable author. And in the European Union the case [ofir.dk vs home.dk](#) decided that regularly crawling and deep linking is permissible.

Is web scraping legal?

- Web scraping can be legal if it is done with permission or if the data being scraped is publicly available and not confidential or protected.

Interesting article about web scraping legality

Activity 6

- Plot a map with the world capitals.
- Information can be retrieve from this [website](#).
- Use R to gather the data from the table and plot their coordinates in a map.

Homework

Run and understand the R code on your own and do the suggested activities.

References