WEEK_2_uploading_data

Kevin Healy

2023-08-28

Uploading data and packages

In this script you will upload some data that is in the form of a .csv file, download and load in some packages. USe this file to help you with the weekly workbook.

Downloading packages.

Its a good idea to load all of the packages you plan to use at the start to keep eveything nice a tidy.

For this session we are going to download the package 'pageviews' which will download the number of wiki views for a given page input (note you will need the internet to use this), ...

```
#This line will download the package from CRAN. You only need to do this
#once so next time you open R you only need to use the library() line.

#Remove the # when running this line
# install.packages("pageviews")

#This line will download the package from CRAN.
library(pageviews)
```

We will now use this package to capture some pageview data for wikipedia. Lets do it for the little blue macaw (Cyanopsitta spixii) which you might know better as the parrot from the Disney movie Rio. Lets track daily page views between 2016 and 2023 using the code below

```
#This code will download the page views for Cyanopsitta spixii
#and save it as a dataframe object.
wiki_data <- article_pageviews(project = "en.wikipedia",
                               article = "Cyanopsitta spixii",
                               start = as.Date('2018-01-01'),
                               end = as.Date("2023-01-01"),
                               user_type = c("user"),
                               platform = "all",
                               granularity = "daily"
                              )
```

The data is saved as a data.frame so we can use the row column formate of wiki_data[,] like in last weeks datasets. To see the first 10 lines we can also use head() were we will see the column names such as data and views.

```
head(wiki_data)
```

```
##     project language            article     access agent granularity       date
## 1 wikipedia       en Cyanopsitta_spixii all-access  user       daily 2018-01-01
## 2 wikipedia       en Cyanopsitta_spixii all-access  user       daily 2018-01-02
## 3 wikipedia       en Cyanopsitta_spixii all-access  user       daily 2018-01-03
## 4 wikipedia       en Cyanopsitta_spixii all-access  user       daily 2018-01-04
## 5 wikipedia       en Cyanopsitta_spixii all-access  user       daily 2018-01-05
## 6 wikipedia       en Cyanopsitta_spixii all-access  user       daily 2018-01-06
##   views
## 1     6
## 2     8
## 3    11
## 4     6
## 5     4
## 6     5
```

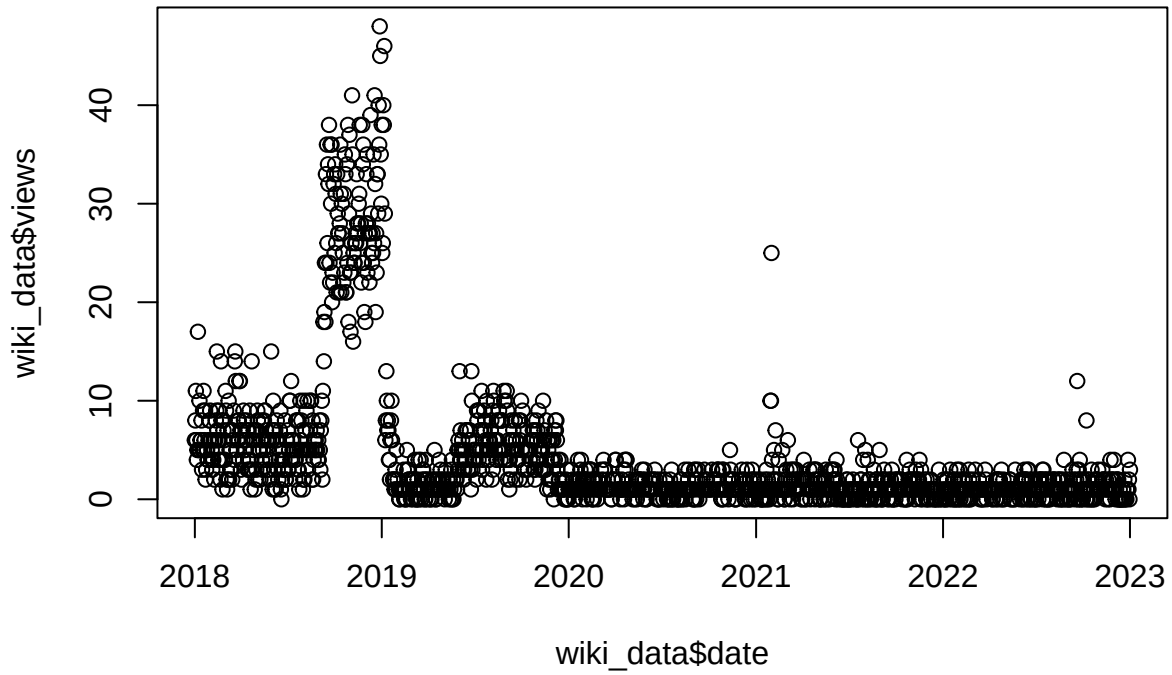
We can now ask things like what is the mean number of daily views, etc.

```
mean(wiki_data$views)
```

```
## [1] 4.623215
```

We can also plot the data (more on this next week) using the below code. What do you think happened towards the end of 2018?

```
plot(wiki_data$views ~ wiki_data$date)
```



Uploading data

From CANVAS download the file names lifespan_data_28_8_2023.xls and save it as a .csv in a folder on your computer. Make sure the name of the file matches the name below and that you have set your working directory to the folder with the file in it.

If it works you should see it appear in the environment in the top right panel but nothing will happen in the console.

```
lifespan_data <- read.csv("lifespan_data_28_8_2023.csv",
                          header = T,
                          sep = ",")
```

This dataset is from a paper testing how a species ecology might drive thier lifespan (See paper here https://royalsocietypublishing.org/doi/full/10.1098/rspb.2014.0298).

```
head(lifespan_data)
```

```
##                     species    class           order maximum_lifespan_yr mass_g
## 1 Dicrostonyx_groenlandicus Mammalia        Rodentia                 3.3     66
## 2          Didelphis_aurita Mammalia Didelphimorphia                 4.0    850
## 3     Didelphis_marsupialis Mammalia Didelphimorphia                 4.2   1530
## 4      Didelphis_virginiana Mammalia Didelphimorphia                 6.6   3000
## 5         Dinomys_branickii Mammalia        Rodentia                12.8  12250
## 6         Diphylla_ecaudata Mammalia      Chiroptera                 8.0     28
##     volancy fossoriallity foraging_environment daily_activity
## 1 nonvolant semifossorial          terrestrial     cathemeral
## 2 nonvolant  nonfossorial         semiarboreal      nocturnal
## 3 nonvolant  nonfossorial         semiarboreal      nocturnal
## 4 nonvolant  nonfossorial         semiarboreal      nocturnal
## 5 nonvolant semifossorial          terrestrial      nocturnal
## 6    volant  nonfossorial          terrestrial      nocturnal
```

Like before you can subset the dataset using either square brackets lifespan_data[,] or the dollar sign lifespan_data$

Lets subset the data to just flying (volant) species and calculate the mean

```
#Do it using brackets
mean(lifespan_data[lifespan_data$volancy == "volant", "maximum_lifespan_yr"])
```

```
## [1] 21.29683
```

```
#We could also do it this way using the dolar sign
mean(lifespan_data[lifespan_data$volancy == "volant", ]$maximum_lifespan_yr)
```

```
## [1] 21.29683
```

We can also use arguments like the greater than sign to ask for animals larger that a certain size. In this case we will do it for animals greater than 100g

```r
#save an object with just the volant species
mean(lifespan_data[lifespan_data$mass_g > 100, "maximum_lifespan_yr"])
```

```
## [1] 22.80689
```

Remember you can save objects such as subsets of a data.frame. For example, we can save two subset of the main dataset, one for mammals and one for Aves in order to make it a little easier to do further subsetting.

```r
#subset to just mammals and save it as Mammal_data
Mammal_data <- lifespan_data[lifespan_data$class == "Mammalia",]

#Lets calculate mean body mass of nocturnal mammals body mass
mean(Mammal_data[Mammal_data$daily_activity == "nocturnal", "mass_g"])
```

```
## [1] 18159.93
```

```r
#subset to just aves and save it as Aves_data
Aves_data <- lifespan_data[lifespan_data$class == "Aves",]

#Lets calculate mean body mass of nocturnal aves body mass
mean(Aves_data[Aves_data$daily_activity == "nocturnal", "mass_g"])
```

```
## [1] 1062.121
```