Data mining and wrangling - Eid Occurance

Innocenter Amima

2019/06/05

In the spirit of Eid I present this and for my curiosity in wrangling and mining data in R.

A brief history : Id-Ul-Fitr commonly known as Eid marks the end of fasting (*Ramadhan*) and is the first day of the Islamic month *Shawwal*.

The following link contains information about Eid, its occurence of per Day, Month, year. For today, I intend to only mine the table and explore.

Objectives

1. Mine data - table - from a URL using the package rvest (this was my first time and its really simple)
2. Explore basic data cleaning using dplyr (this is included in tidyverse library)
3. Explore the occurence of Eid i.e per month, per day
4. Perform visualization using ggplot :-).

This is a learning curve and feel free to drop in your comments and/or suggestions. I will show you a bit of my thought process when analysing and wrangling data.

Let's go 😄 □

Side note check this link out to learn more about including emojis in a markdown.

Loading packages required

```
library(tidyverse) # data wrangling
library(rvest) #used for web scraping
```

Data Mining

Loading the data - but first we have to mine it from the URL provided

```
url.page <- read_html('https://www.timeanddate.com/holidays/kenya/eid-al-fitr')

Eid.table <- html_nodes(url.page, 'table')

head(Eid.table)
```

```
## {xml_nodeset (3)}
## [1] <table class="tb-quick-facts">\n<tr>\n<th>This year:</th>\n<td>J5, 5 ...
## [2] <table id="tb-hol_obs" class="tb-theme fw sep">\n<thead><tr>\n<th>Ye ...
## [3] <table class="tb-quick-facts"><tbody>\n<tr>\n<th>English</th>\n<td>E ...
```

There are 3 tables - no idea which one contains the Eid data. I will extract all the tables - out of curiosity and they're only 3. If we had several tables - we could explicity use the table names e.g *html_nodes('#table2')*

```
Eid.tables <- url.page %>%
  html_nodes('table') %>% #to select <table> nodes
  .[1:3] %>%
  html_table(fill = TRUE)

str(Eid.tables) # from 2015 t0 2025
```

```
## List of 3
##  $ :'data.frame':    4 obs. of  2 variables:
##   ..$ X1: chr [1:4] "This year:" "Next year:" "Last year:" "Type:"
##   ..$ X2: chr [1:4] "J5, 5 Jun 2019" "J3, 25 Mei 2020" "J1, 16 Jun 2018" "Public holiday"
##  $ :'data.frame':    12 obs. of  5 variables:
##   ..$ Year        : int [1:12] 2015 2015 2016 2017 2018 2019 2020 2021 2022 2023 ...
##   ..$ Weekday     : chr [1:12] "J2" "J3" "Alh" "J3" ...
##   ..$ Date        : chr [1:12] "19 Jul" "20 Jul" "7 Jul" "26 Jun" ...
##   ..$ Name        : chr [1:12] "Eid al-Fitr" "Eid al-Fitr observed" "Eid al-Fitr" "Eid al-Fitr" ...
##   ..$ Holiday Type: chr [1:12] "Public holiday" "Public holiday" "Public holiday" "Public holiday" ...
##  $ :'data.frame':    4 obs. of  2 variables:
##   ..$ X1: chr [1:4] "English" "German" "Norwegian" "Swahili"
##   ..$ X2: chr [1:4] "Eid al-Fitr, End of Ramadan" "Eid al-Fitr (Fest des Fastenbrechens)" "Eid al-Fitr, S
lutt på ramadan" "Idd el Fitr, Mwisho wa Ramadhani"
```

The second table contains information we are interested in - it has Eid occurance data from the year 2015 - 2025 (some are predictions).

```
Eid <- Eid.tables[[2]]
```

Another method to extact the table is by creating an empty list and populating it with data

```
Eid2 <- list() #creating an empty list

Eid2 <- url.page %>%
  html_nodes('table') %>%
  html_table(fill = TRUE) %>%
  .[[2]] #populating it with table 2
```

EDA and cleaning

```
str(Eid)
```

```
## 'data.frame':    12 obs. of  5 variables:
##  $ Year        : int  2015 2015 2016 2017 2018 2019 2020 2021 2022 2023 ...
##  $ Weekday     : chr  "J2" "J3" "Alh" "J3" ...
##  $ Date        : chr  "19 Jul" "20 Jul" "7 Jul" "26 Jun" ...
##  $ Name        : chr  "Eid al-Fitr" "Eid al-Fitr observed" "Eid al-Fitr" "Eid al-Fitr" ...
##  $ Holiday Type: chr  "Public holiday" "Public holiday" "Public holiday" "Public holiday" ...
```

The data contains (12, 5) - that is 12 observations and 5 variables

In the year 2015 - there exists two entries the second one is the observed and thus will delete the first entry

```
Eid = Eid[-1,]
```

From the structure above, we can see that the names of weekday is written in some language for example Sunday is J2 - I checked the English equivalence in the website and replaced them.

The column names and data types are

```
colnames(Eid)
```

```
## [1] "Year"         "Weekday"      "Date"         "Name"
## [5] "Holiday Type"
```

```
c(typeof(Eid$Year), typeof(Eid$Weekday), typeof(Eid$Date))
```

```
## [1] "integer"   "character" "character"
```

The analysis will be based on the month and hence I separate the day from the month in column Date

```
Eid <- Eid %>%
  separate(Date, c('Day', 'Month'))
```

The month of May is written as Mei - I replaced that and the weekday as shown below.

> From the website: Sun-J2, Mon-J3, Tue-J4, Wed-J5, Thu-Alh, Fri-Ij, Sat-J1

```
Eid$Month <- with(Eid, replace(Month, Month == "Mei", "May"))
Eid$Weekday <- with(Eid, replace(Weekday, Weekday == "J1",  "Sat" ))
Eid$Weekday <- with (Eid, gsub("J2", "Sun", Weekday ))
```

I had to do these one item after another - I will figure out a way next time maybe a loop

The *replace* and *gsub* functions worked for just one item - I tried concatinating the other and got an error while compiling or the matching was not exactly correct. I got this error with gsub *argument 'replacement' has length > 1 and only the first element will be used*

I also replaced the rest

Here is our cleaned data

```
Eid
```

```
##    Year Weekday Day Month                 Name   Holiday Type
## 2  2015     Mon  20   Jul Eid al-Fitr observed Public holiday
## 3  2016     Thu   7   Jul          Eid al-Fitr Public holiday
## 4  2017     Mon  26   Jun          Eid al-Fitr Public holiday
## 5  2018     Sat  16   Jun          Eid al-Fitr Public holiday
## 6  2019     Wed   5   Jun          Eid al-Fitr Public holiday
## 7  2020     Mon  25   May          Eid al-Fitr Public holiday
## 8  2021     Fri  14   May          Eid al-Fitr Public holiday
## 9  2022     Wed   4   May          Eid al-Fitr Public holiday
## 10 2023     Sun  23   Apr          Eid al-Fitr Public holiday
## 11 2024     Thu  11   Apr          Eid al-Fitr Public holiday
## 12 2025     Tue   1   Apr          Eid al-Fitr Public holiday
```

```
Eid.month <- Eid %>% count(Month, name="Month_occurence")
Eid.month
```

```
## # A tibble: 4 x 2
##   Month Month_occurence
##   <chr>           <int>
## 1 Apr                 3
## 2 Jul                 2
## 3 Jun                 3
## 4 May                 3
```

For the past 12 years each month has been represented three times except for July.

```
Eid.day <- Eid %>% count(Weekday, name='Weekday_occurence')
Eid.day
```

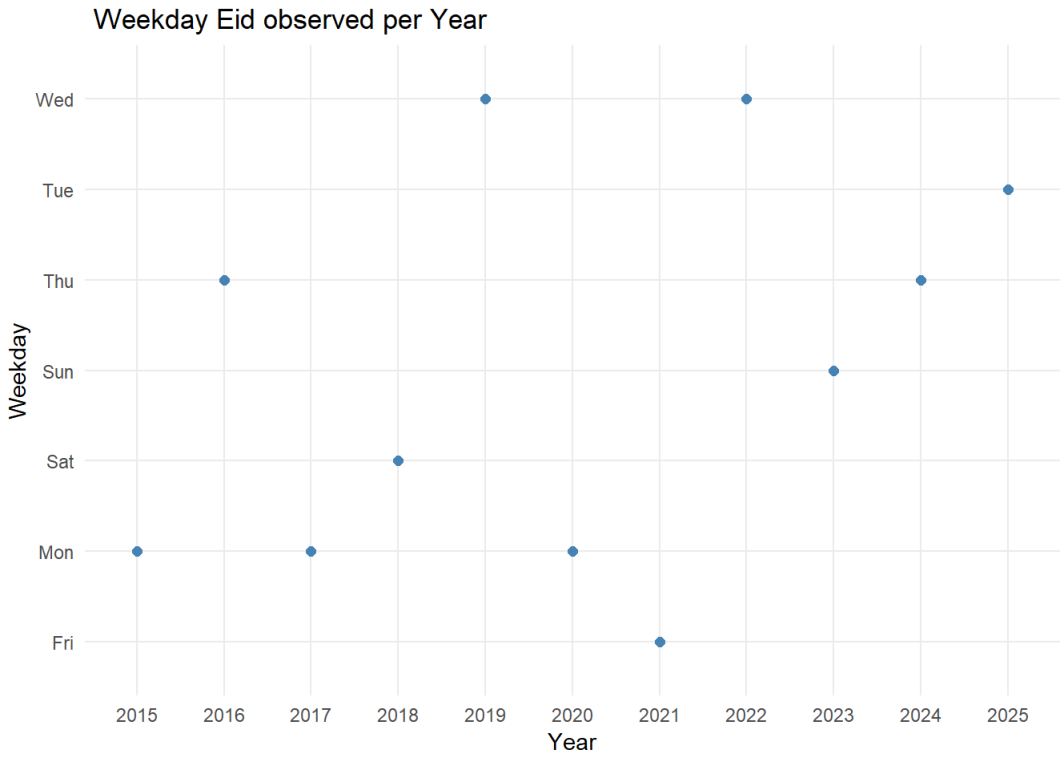
```
## # A tibble: 7 x 2
##   Weekday Weekday_occurence
##   <chr>               <int>
## 1 Fri                     1
## 2 Mon                     3
## 3 Sat                     1
## 4 Sun                     1
## 5 Thu                     2
## 6 Tue                     1
## 7 Wed                     2
```

Eid was observed mostly on Monday from the year 2015 - 2025 - this is the future ☺

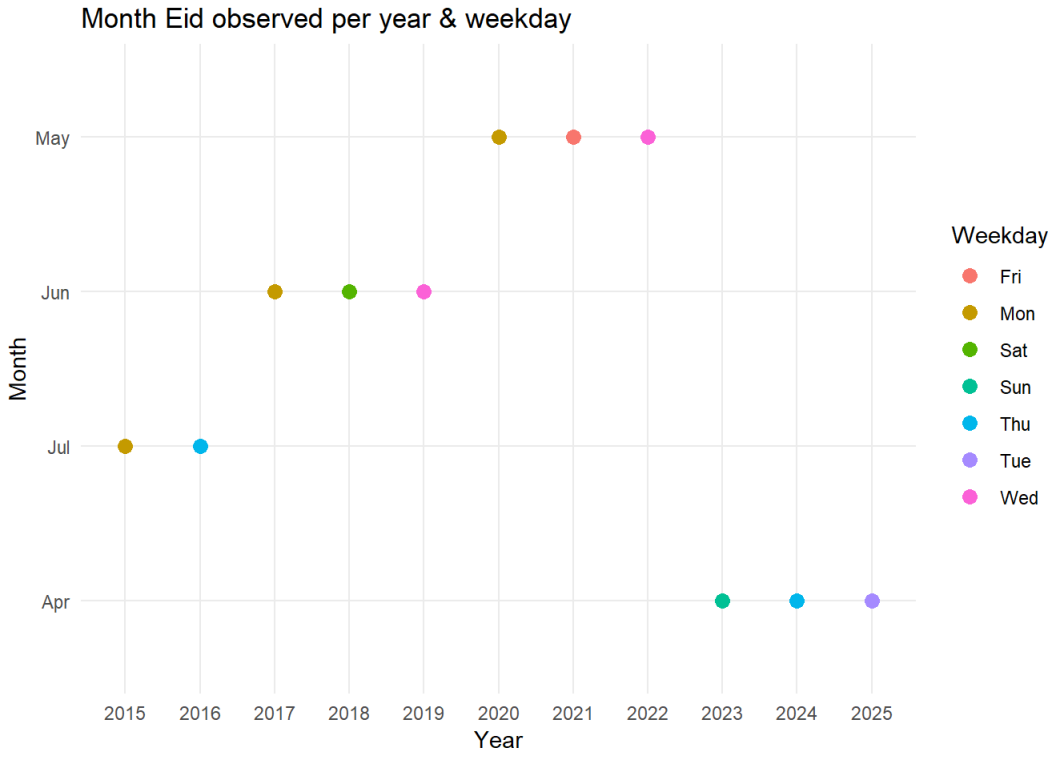
Visualization

Our fourth objective was to visualize the data and get an insight on the month or day that Eid is observed for the 12 years.

```
ggplot(Eid, aes(Year, Weekday)) +
  geom_point(color="steelblue", shape=20, size =3)+
  scale_x_discrete(limits = c(2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2024, 2025))+ #To preorder the x axis
  labs (title = " Weekday Eid observed per Year")+
  theme_minimal()
```




```
ggplot(Eid, aes(Year, Month))+
  geom_point(aes(colour=Weekday), size = 3)+
  scale_x_discrete(limits = c(2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2024, 2025))+ #To preorder the x axis
  labs(title = "Month Eid observed per year & weekday")+
  theme_minimal()
```



> Take home notes : this was a refresher for me and aluta continua

I hope you have enjoyed this - though short 😄

> From this data, the prediction is that next year - Eid will occur on a Monday in the month of May - I shall sit tight and wait for it.

> Have a blessed Eid