HW11

Aidan Pizzo

November 15, 2021

#HW10.R

```
library(httr)
library(dplyr)
```

```
##
## 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
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
myKey <- "E5QQEg3149FFlf6WRnCOhlTfUg9CjomM"

movie_info <- function(query, start_date = "", end_date = "", key = X)
{
  base <- "https://api.nytimes.com/svc/movies/v2/reviews/search.json?"
  dates <- paste("opening-date=", start_date, ":" , end_date, sep="")
  query_info <- paste("query=" , query, sep="")
  key_info <- paste("api-key=", key, sep="")
  if (nchar(start_date) == 0)
  {
    url1 <- paste(base, query_info, "&", key_info, sep="")
  }
  else
```

```
  {
    if (nchar(query) == 0)
      url1 <- paste(base, dates, "&", key_info, sep="")
    else
      url1 <- paste(base, query_info, "&", dates, "&", key_info, sep="")
  }
  x <- httr::GET(url=url1)
  j <- jsonlite::fromJSON(rawToChar(x$content))
  out_df <- data.frame( movie_title = character(),
                        opening_date = character(),
                        summary_short = character(),
                        link = character(),
                        stringsAsFactors = F)
  offset <- 20
  while(!(is.null(j$results)))
  {
    add_this_df <- dplyr::select(j$results,
                                 movie_title = display_title,
                                 opening_date,
                                 summary_short)
    add_this_df <- cbind(add_this_df, link = j$results$link$url)
    out_df <- rbind(out_df, add_this_df)
    url2 <- paste(url1, "&offset=", offset, sep="")
    x <- httr::GET(url=url2)
    j <- jsonlite::fromJSON(rawToChar(x$content))
    offset <- offset + 20
  }
  return(out_df)
}


show_review <- function(mi, i)
{
  theLink <- mi$link[i]
  Sys.setenv(google="/Applications/Google Chrome.app/Contents/MacOS/Google Chrome")
  Sys.setenv(article = theLink)
  system('"$google" $article')
}

say_summary <- function(mi, i)
{
  say_this <- mi$summary_short[i]
  Sys.setenv(say_string=say_this)
  system("say $say_string")
}

movies_info <- movie_info(query = "", start_date = "1980-07-01", end_date = "1990-07-31", key = myKey)
movies_length <- nrow(movies_info)
split <- strsplit(movies_info$opening_date, "-")
july_ndx <- rep(FALSE, movies_length)
for (i in 1:movies_length)
{
  if (split[[i]][2] == "07")
```

```
  { #create boolean vector to filter for movies in july
    july_ndx[i] <- TRUE
  }
}
movies_info <- dplyr::filter(movies_info, july_ndx)
max_summary <- nchar(movies_info$summary_short) %>% which.max
say_summary(movies_info, max_summary)
show_review(movies_info, max_summary)
```