Problem 2

Yufei Yin

Question 2a, (2 points):

Write an R function named boxoffice. This function should take no arguments and return a dataframe with columns Name, BoxOffice, and PerWeek, and 10 rows for each of the day's top movies on the website https://www.imdb.com/chart/boxoffice (i.e., the function should scrape this website). The PerWeek column should contain the gross box office revenue (i.e., the second column) divided by the number of weeks the movie's been running for. Provide the R code.

```
library(rvest)
library(tidyverse)
library(httr)
boxoffice = function(){
  df = data.frame(Title = rep(NA,10),
                  BoxOffice = rep(NA,10),
                  PerWeek = rep(NA,10))
  movie_url = "https://www.imdb.com/chart/boxoffice"
  movie_table = read_html("https://www.imdb.com/chart/boxoffice")
  length(html_nodes(movie_table, "table"))
  zz = html_table(html_nodes(movie_table, "table")[[1]])
  df$Title = zz[,2]
  df$BoxOffice = zz[,4]
  df$PerWeek = paste0("$",sprintf("%.2f",parse_number(zz[,4])/zz[,5]),"M")
  return(df)
}

boxoffice()
```

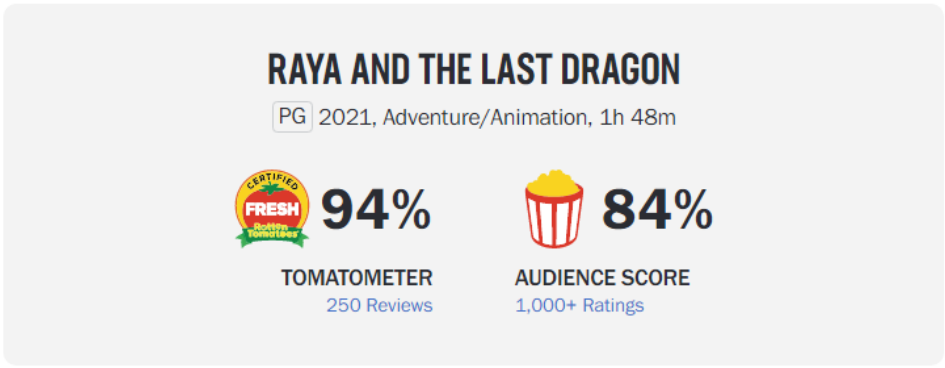
```
##                          Title BoxOffice PerWeek
## 1     Raya and the Last Dragon     $8.5M  $8.50M
## 2                Tom and Jerry    $23.0M $11.50M
## 3                Chaos Walking     $3.8M  $3.80M
## 4                       Boogie     $1.2M  $1.20M
## 5        The Croods: A New Age    $53.6M  $3.57M
## 6            The Little Things    $13.7M  $2.28M
## 7            Wonder Woman 1984    $44.4M  $4.04M
## 8                 The Marksman    $13.0M  $1.62M
## 9  Judas and the Black Messiah     $4.5M  $1.12M
## 10              Monster Hunter    $14.4M  $1.20M
```

Question 2b, (3 points + 2 bonus points):

Modify the function you wrote in the first part of this question to add a column named RT. The values of this column should be the rating that the movie received on https://www.rottentomatoes.com (you may use the tomatometer or the audience score). Note that you will have to construct the URL for the movie in order to scrape it from the rotten tomatoes website. For example, the movie Onward is returned in the IMDB boxoffice page with title Onward (i.e., with a capital O) and the corresponding rotten tomatoes website is https://www.rottentomatoes.com/m/onward/ with a lower case o. Provide the modified R function. Your solution can be approximate: it need not work for all movie titles (for example, special characters or in the movie title or long movie titles may be challenges, but it's likely that you can get large coverage without handling it). Use NA (not available) to indicate ratings of movies that you can't match. Bonus points will be awarded for exceptionally large coverage.

```
url = 'https://www.rottentomatoes.com/m/raya_and_the_last_dragon'
(score = read_html(url) %>%
        html_nodes("score-board") %>%
        html_attr("audiencescore"))
```

```
## [1] "84"
```



(The current score may be different than it appeared in the screenshot)

I figured out how to use functions from `rvest` packages to extract audience scores, but when i apply it to a for loop i get warning message "closing unused connection", i guess it related to trycatch() function; however, i can not solve it, so i choose to use regular expression to extract audience score.

```
# modified boxoffice function
boxoffice = function(){
  # add new column RT
  df = data.frame(Title = rep(NA,10),
                  BoxOffice = rep(NA,10),
                  PerWeek = rep(NA,10),
                  RT = rep(NA,10))
  movie_url = "https://www.imdb.com/chart/boxoffice"
  movie_table = read_html("https://www.imdb.com/chart/boxoffice")
  length(html_nodes(movie_table, "table"))
  zz = html_table(html_nodes(movie_table, "table")[[1]])
  df$Title = zz[,2]
  df$BoxOffice = zz[,4]
  df$PerWeek = paste0("$",sprintf("%.2f",parse_number(zz[,4])/zz[,5]),"M")

  # movies name
  name = df$Title
```

```
  # change to lower case
  lowercase.name = str_to_lower(name)

  # replace space with undercore
  underscore.name = str_replace_all(lowercase.name, " ", "_")

  # remove colon
  no.colon.name = str_replace_all(underscore.name, ":","")

  # paste to rottentomatoes website address
  url = paste0('https://www.rottentomatoes.com/m/', no.colon.name)

  # get current year
  currentyear = str_sub(Sys.Date(),1,4)

  # get last year
  lastyear = as.character(as.numeric(currentyear)-1)

  # for loop
  score = NULL
  # normal case
  for (i in 1:10){
      score[i] =  read.url(url[i])
  }
  # movie name + current year
  url2 = paste0(url,"_",currentyear)
  for (i in 1:10){
    if (!is.na(read.url(url2[i]))){
      score[i] =  read.url(url2[i])
    }
  }

  # movie name + last year
  url3 = paste0(url,"_",lastyear)
  for (i in 1:10){
    if (!is.na(read.url(url3[i]))){
      score[i] =  read.url(url3[i])
    }
  }

  # RT
  df$RT = str_c(score,"%")

  return(df)
}
```

```r
# get hint from
# https://stackoverflow.com/questions/12193779/how-to-write-trycatch-in-r #

# read.url() will return audience score if it runs properly, otherwise, it will return NA.
read.url <- function(url){
  tryCatch(
    expr = {
      s = GET(url)
      s = content(s, "text")
      s = str_replace_all(s, '[[:space:]]','')
      match = str_match_all(s, "audiencescore=\"(.*?)\"class")
      score = match[[1]][2]
      return(score)
    },
    error = function(e){
      return(NA)
    }
  )
}

boxoffice()
```

```
##                         Title BoxOffice PerWeek  RT
## 1     Raya and the Last Dragon     $8.5M  $8.50M 84%
## 2                Tom and Jerry    $23.0M $11.50M 84%
## 3                Chaos Walking     $3.8M  $3.80M 76%
## 4                       Boogie     $1.2M  $1.20M 69%
## 5        The Croods: A New Age    $53.6M  $3.57M 95%
## 6            The Little Things    $13.7M  $2.28M 66%
## 7            Wonder Woman 1984    $44.4M  $4.04M 74%
## 8                 The Marksman    $13.0M  $1.62M 85%
## 9  Judas and the Black Messiah     $4.5M  $1.12M 95%
## 10              Monster Hunter    $14.4M  $1.20M 70%
```