STAT240 Lab9

Jiajun Zhang

March 22, 2019

Question1

```r
library(stringr)
course_url = "https://www.sfu.ca/outlines.html?2019/spring/stat/240/d100"
course_page = readLines(course_url)
```

a)

```r
h3_headings=function(webpage){
  headings=grep("<h3", webpage, value=T)
  #remove spaces
  headings=gsub("\\s{2,}", "", headings)
  #remove the beginning html <> patterns
  headings=gsub("^.*?>", "", headings)
  #remove the ending <> patterns
  headings=gsub("*?<+.+$", "", headings)
  headings
}
h3_headings(course_page)
```

```
## [1] "Class Number: 3431"         "Delivery Method: In Person"
```

b)

```r
course_name=function(webpage){
  h1_headings=grep("<h1 id=\"name\">", webpage, value=T)
  # h1_headings
  #remove beginning spaces
  h1_headings=gsub("\\s{2,}", "", h1_headings)
  #remove the beginning html <> pattern
  h1_headings=gsub("^.*?>","", h1_headings)
  #remove everything after STAT240
  h1_headings=gsub("<.+>", "", h1_headings)
  #remove everything before -
  course=gsub("^.+?\\-","", h1_headings)
  #trim the leading/trailing spaces
  (course=trimws(course))
}
course_name(course_page)
```

```
## [1] "STAT 240"
```

c)

```r
course_title=function(webpage){
  h2_index=grep("<h2 id=\"title\">", webpage)
  title=webpage[h2_index+1]
  #remove the leading space
  (title=trimws(title))
}
course_title(course_page)
```

```
## [1] "Introduction to Data Science"
```

d)

```r
# grep("<h4>", course_page, value=T)
#Instructor info is b/w <h4>Instructor:</h4> and <h4>Prerequisites:</h4>
course_instructor=function(webpage){
  h4_index1=grep("<h4>Instructor:</h4>", webpage)
  h4_index2=grep("<h4>Prerequisites:</h4>", webpage)
  # c(h4_index1, h4_index2)
  # course_page[248:262]
  instructor=webpage[h4_index1+1]
  #remove the beginning <> pattern and leading/ending spaces
  instructor=trimws(gsub("^.*?>","", instructor))
  (instructor=gsub("*?<+.+$", "", instructor))
}
course_instructor(course_page)
```

```
## [1] "David Campbell"
```

e)

```r
# grep("<h4>", course_page, value=T)
course_time_place=function(webpage){
  h4_index3=grep("<h4>Course Times \\+ Location:", webpage)
  time_place=webpage[h4_index3+1]
  time_place=gsub("^.*?>", "", time_place)
  #change $ndash to -
  time_place=str_replace_all(time_place, "&ndash;", "-")
  #remove <br> and </p>
  time_place=gsub("<.*?>"," ", time_place)
  (time_place=trimws(time_place))
}
course_time_place(course_page)
```

```
## [1] "Mo 12:30 PM - 2:20 PM EDB 7618, Burnaby"
```

f)

```r
# Long function used for second question
course_book=function(webpage){
  book_index=grep("<h4>REQUIRED READING:</h4>", webpage)
  book_index1=grep("<h4>RECOMMENDED READING:</h4>", webpage)
  if(length(book_index1)==0){
      textbook=webpage[book_index+4]
      if(str_detect(textbook, "Required Textbook:")==TRUE){
          textbook=gsub("Required Textbook:", "", textbook)
      }
      if(str_detect(textbook, "&+[[:alpha:]]+\\;")==TRUE){
          textbook=gsub("&+[[:alpha:]]+\\;", "", textbook)
      }
      textbook=gsub("<.*?>", "", textbook)
      (textbook=trimws(textbook))
  }
  else if(length(book_index)==0){
      textbook1=webpage[book_index1+4]
      if(str_detect(textbook1, "Required Textbook:")==TRUE){
          textbook1=gsub("Required Textbook:", "", textbook1)
      }
      if(str_detect(textbook1, "&+[[:alpha:]]+\\;")==TRUE){
          textbook1=gsub("&+[[:alpha:]]+\\;", "", textbook1)
      }
      textbook1=gsub("<.*?>", "", textbook1)
      (textbook1=trimws(textbook1))
  }
}
course_book(course_page)
```

```
## [1] "Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining. Authors: S
```

g)

```r
# c(grep("<h4>Exam Times \\+ Location:</h4>", course_page),
#   grep("<h4>Instructor:</h4>", course_page))
# course_page[234:248]
# course_page[236:241]

course_2exams=function(webpage){
  date=trimws(gsub("<.*?>", "", webpage[c(236, 240)]))
  time=trimws(webpage[c(237, 241)])
  time=str_replace_all(time, "&ndash;", "-")
  #remove everything after time, starting from - since we only want the start time
  time=trimws(gsub("\\-+.+$","", time))

  place=trimws(webpage[c(237, 241)])
  #remove everything before place
  place=gsub("^.*?>", "", place)
  #Exam TAKES PLACE IN TWO DIFFERENT ROOMS OR TIMES
  matrix(c(time, date, place), nrow=2, byrow=F)
}
course_2exams(course_page)
```

```
##      [,1]       [,2]           [,3]
## [1,] "12:00 PM" "Apr 15, 2019" "WMC 2502, Burnaby"
## [2,] "12:00 PM" "Apr 15, 2019" "AQ 3144, Burnaby"
```

Question2

```r
EVSC100_url="https://www.sfu.ca/outlines.html?2017/spring/evsc/100/d100"
EVSC_page=readLines(EVSC100_url)
STAT452_url="https://www.sfu.ca/outlines.html?2018/fall/stat/452/d100"
STAT452_page=readLines(STAT452_url)
#Courses offered this term with different sections
STAT100_url="https://www.sfu.ca/outlines.html?2019/spring/stat/100/d100"
STAT100_page=readLines(STAT100_url)
STAT203_url_c100="https://www.sfu.ca/outlines.html?2019/spring/stat/203/c100"
STAT203_page_c100=readLines(STAT203_url_c100)
STAT203_url_d100="https://www.sfu.ca/outlines.html?2019/spring/stat/203/d100"
STAT203_page_d100=readLines(STAT203_url_d100)
STAT270_url_c100="https://www.sfu.ca/outlines.html?2019/spring/stat/270/c100"
STAT270_page_c100=readLines(STAT270_url_c100)
STAT270_url_d100="https://www.sfu.ca/outlines.html?2019/spring/stat/270/d100"
STAT270_page_d100=readLines(STAT270_url_d100)
STAT270_url_d900="https://www.sfu.ca/outlines.html?2019/spring/stat/270/d900"
STAT270_page_d900=readLines(STAT270_url_d900)
```

```r
h3_headings=function(webpage){
  headings=grep("<h3", webpage, value=T)
  headings=gsub("\\s{2,}", "", headings)
  headings=gsub("^.*?>", "", headings)
  headings=gsub("*?<+.+$", "", headings)
  paste(headings[1], headings[2], sep=" & ")
}

#Use previous codes, now create a function to find the exam info for rest of the courses
#info is located line 237-238
course_1exam=function(webpage){
  date=trimws(gsub("<.*?>", "", webpage[237]))
  time=trimws(webpage[c(238)])
  time=str_replace_all(time, "&ndash;", "-")
  #remove everything after time, starting from - since we only want the start time
  time=trimws(gsub("\\-+.+$","", time))
  place=trimws(webpage[c(238)])
  #remove everything before place
  place=gsub("^.*?>", "", place)

  paste(time, date, place)
}

# EVSC is 236-237
course_1exam_evsc=function(webpage){
  date=trimws(gsub("<.*?>", "", webpage[236]))
  time=trimws(webpage[c(237)])
  time=str_replace_all(time, "&ndash;", "-")
  #remove everything after time, starting from - since we only want the start time
  time=trimws(gsub("\\-+.+$","", time))
  place=trimws(webpage[c(237)])
  #remove everything before place
  place=gsub("^.*?>", "", place)
  paste(time, date, place)
}

#Many courses have 2 exams info on it, recreate function to find them
course_2exams=function(webpage){
  date=trimws(gsub("<.*?>", "", webpage[c(236, 240)]))
  time=trimws(webpage[c(237, 241)])
  time=str_replace_all(time, "&ndash;", "-")
  #remove everything after time, starting from - since we only want the start time
  time=trimws(gsub("\\-+.+$","", time))
  place=trimws(webpage[c(237, 241)])
  #remove everything before place
  place=gsub("^.*?>", "", place)

  #Make it to one line b/c we need it for the following data frame
  paste( paste(time[1], date[1], place[1]),
         paste(time[2], date[2], place[2]), sep=" & ")
}


#Recreate instructor function b/c of online courses
course_instructor=function(webpage){
  index1=grep("<h4>Instructor:</h4>", webpage)
  #Distance Education sections do not have instructor name
  if(length(index1)==0){
    return("Distance Education")
    break
  }
  else{
    instructor1=webpage[index1+1]
    #remove the beginning <> pattern and leading/ending spaces
    instructor1=trimws(gsub("^.*?>","", instructor1))
    (instructor1=gsub("*?<+.+$", "", instructor1))
  }
}


course_timeplace=function(webpage){
  index=grep("(<h4>Course Times \\+ Location:)", webpage)
    time_place2=webpage[(index+1):(index+2)]
    time_place2=gsub("^.*?>", "", time_place2)
    #change $ndash to -
    time_place2=str_replace_all(time_place2, "&ndash;", "-")
    #remove <br> and </p>
    time_place2=gsub("<.*?>"," ", time_place2)
    time_place2=trimws(time_place2)
    paste(time_place2[1], time_place2[2], sep="  ")
}

#Create a function to find the course section
course_section=function(webpage){
  IND=grep("<h1 id=\"name\">", webpage, value=T)
  str_extract(IND, "[[:upper:]]+\\d{3}")
```

```
}


courses_df=data.frame( matrix( c( h3_headings(course_page), course_name(course_page),
                        course_section(course_page), course_title(course_page),
                        course_instructor(course_page), course_time_place(course_page),
                        course_2exams(course_page), course_book(course_page),
                h3_headings(EVSC_page),course_name(EVSC_page), course_section(EVSC_page),
                course_title(EVSC_page), course_instructor(EVSC_page),
                course_time_place(EVSC_page), course_1exam_evsc(EVSC_page), course_book(EVSC_page),

                h3_headings(STAT452_page), course_name(STAT452_page), course_section(STAT452_page),
                course_title(STAT452_page), course_instructor(STAT452_page),
                course_timeplace(STAT452_page), course_1exam(STAT452_page), course_book(STAT452_page),

                h3_headings(STAT100_page), course_name(STAT100_page), course_section(STAT100_page),
                course_title(STAT100_page), course_instructor(STAT100_page),
                course_timeplace(STAT100_page), course_1exam(STAT100_page), course_book(STAT100_page),

                h3_headings(STAT203_page_c100), course_name(STAT203_page_c100), course_section(STAT203_p
                course_title(STAT203_page_c100), course_instructor(STAT203_page_c100),
                course_timeplace(STAT203_page_c100), course_2exams(STAT203_page_c100), course_book(STAT2

                h3_headings(STAT203_page_d100), course_name(STAT203_page_d100), course_section(STAT203_p
                course_title(STAT203_page_d100), course_instructor(STAT203_page_d100),
                course_timeplace(STAT203_page_d100), course_1exam(STAT203_page_d100), course_book(STAT20

                h3_headings(STAT270_page_c100), course_name(STAT270_page_c100), course_section(STAT270_p
                course_title(STAT270_page_c100), course_instructor(STAT270_page_c100),
                course_timeplace(STAT270_page_c100), course_2exams(STAT270_page_c100), course_book(STAT2

                h3_headings(STAT270_page_d100), course_name(STAT270_page_d100), course_section(STAT270_p
                course_title(STAT270_page_d100), course_instructor(STAT270_page_d100),
                course_timeplace(STAT270_page_d100), course_1exam(STAT270_page_d100), course_book(STAT27

                h3_headings(STAT270_page_d900), course_name(STAT270_page_d900), course_section(STAT270_p
                course_title(STAT270_page_d900), course_instructor(STAT270_page_d900),
                course_timeplace(STAT270_page_d900), course_1exam(STAT270_page_d900), course_book(STAT27
       nrow=9, byrow=T) )

colnames(courses_df)=c("h3 Headings","Course Number", "Section", "Course Title", "Course Instructor",
                        "Course Time and Location", "Exam Start Time, Date, and Loaction", "Course Textb
courses_df
```

```
##                                             h3 Headings Course Number
## 1          Class Number: 3431 & Delivery Method: In Person      STAT 240
## 2          Class Number: 8909 & Delivery Method: In Person      EVSC 100
## 3          Class Number: 4648 & Delivery Method: In Person      STAT 452
## 4          Class Number: 3420 & Delivery Method: In Person      STAT 100
## 5 Class Number: 3430 & Delivery Method: Distance Education      STAT 203
## 6          Class Number: 3416 & Delivery Method: In Person      STAT 203
## 7 Class Number: 3425 & Delivery Method: Distance Education      STAT 270
## 8          Class Number: 3418 & Delivery Method: In Person      STAT 270
## 9          Class Number: 3426 & Delivery Method: In Person      STAT 270
##   Section                                    Course Title
## 1    D100                    Introduction to Data Science
## 2    D100           Introduction to Environmental Science
## 3    D100             Statistical Learning and Prediction
## 4    D100                       Chance and Data Analysis
## 5    C100 Introduction to Statistics for the Social Sciences
## 6    D100 Introduction to Statistics for the Social Sciences
## 7    C100         Introduction to Probability and Statistics
## 8    D100         Introduction to Probability and Statistics
## 9    D900         Introduction to Probability and Statistics
##    Course Instructor
## 1     David Campbell
## 2  Marnie Branfireun
## 3       Brad McNeney
## 4        Gaitri Yapa
## 5 Distance Education
## 6        Gaitri Yapa
## 7 Distance Education
## 8         Tim Swartz
## 9          Scott Pai
##                                                                  Course Time and Location
## 1                                                    Mo 12:30 PM - 2:20 PM EDB 7618, Burnaby
## 2                                                      Fr 2:30 PM - 4:20 PM SUR 5240, Surrey
## 3 Mo 9:30 AM - 10:20 AM SSCK 9500, Burnaby  We, Fr 9:30 AM - 10:20 AM SSCK 9500, Burnaby
## 4      Mo 2:30 PM - 4:20 PM SSCC 9001, Burnaby  We 2:30 PM - 3:20 PM SSCC 9001, Burnaby
## 5                                                                      Distance Education
## 6  Mo 10:30 AM - 12:20 PM SSCC 9002, Burnaby  We 10:30 AM - 11:20 AM SSCC 9002, Burnaby
## 7                                                                      Distance Education
## 8  Mo, Fr 9:30 AM - 10:20 AM WMC 3520, Burnaby  We 9:30 AM - 10:20 AM SSCC 9002, Burnaby
## 9          Tu 8:30 AM - 10:20 AM SUR 3240, Surrey  Th 8:30 AM - 9:20 AM SUR 3240, Surrey
##                                                    Exam Start Time, Date, and Loaction
## 1 12:00 PM Apr 15, 2019 WMC 2502, Burnaby & 12:00 PM Apr 15, 2019 AQ 3144, Burnaby
## 2                                           3:30 PM Apr 18, 2017 SUR 5280, Surrey
## 3                                        12:00 PM Dec 12, 2018 SWH 10081, Burnaby
## 4                                      12:00 PM Apr 18, 2019 GYM CENTRAL, Burnaby
## 5    7:00 PM Feb 26, 2019 AQ 3149, Burnaby & 3:30 PM Apr 16, 2019 AQ 3150, Burnaby
## 6                                        3:30 PM Apr 10, 2019 RCB IMAGTH, Burnaby
## 7    7:00 PM Feb 27, 2019 AQ 3150, Burnaby & 3:30 PM Apr 12, 2019 AQ 3003, Burnaby
## 8                                        3:30 PM Apr 13, 2019 RCB IMAGTH, Burnaby
## 9                                           8:30 AM Apr 11, 2019 SUR 5100, Surrey
##
## 1
## 2
## 3                                                                                        An Introducti
## 4                                                                      Statistics: Concepts and Contr
## 5
## 6 The Basic Practice of Statistics (8th ed.)  Sapling Plus(Sapling Plus is recommended, but not requi
## 7
## 8
## 9
```

Question3

```r
library(rvest)
```

```
## Warning: package 'rvest' was built under R version 3.5.3
```

```r
library(XML)
library(RCurl)
```

a)

```r
marvel_url="https://en.wikipedia.org/wiki/List_of_Marvel_Cinematic_Universe_films"
marvel=read_html(marvel_url)
BO_perform=html_table(html_nodes(marvel, "table"), fill=T)[[8]]
critical_response=html_table(html_nodes(marvel, "table"), fill=T)[[9]]

#Modify this, removing the first row
BO_perform1=NA
BO_perform1=BO_perform[(2:(nrow(BO_perform)-1)), ]
colnames(BO_perform1)=c("Film", "US Release Date", "US and Canada Box Office Gross",
                        "Other Territories Box Office Gross", "Worldwide Box Office Gross",
                        "US and Canada All-time Ranking", "Worldwide All_time Ranking",
                        "Budget", "Ref(s)")
# Modify the observation numbers
row.names(BO_perform1)=1:nrow(BO_perform1)
#Remove the last row
critical_response=critical_response[(1:(nrow(critical_response)-1)),]

(merge_table=merge(BO_perform1, critical_response, by="Film"))
```

```
##                                    Film   US Release Date
## 1                               Ant-Man     July 17, 2015
## 2                 Ant-Man and the Wasp      July 6, 2018
## 3               Avengers: Age of Ultron       May 1, 2015
## 4                Avengers: Infinity War    April 27, 2018
## 5                         Black Panther February 16, 2018
## 6             Captain America: Civil War       May 6, 2016
## 7   Captain America: The First Avenger     July 22, 2011
## 8  Captain America: The Winter Soldier     April 4, 2014
## 9                        Captain Marvel     March 8, 2019
## 10                      Doctor Strange  November 4, 2016
## 11              Guardians of the Galaxy    August 1, 2014
## 12      Guardians of the Galaxy Vol. 2       May 5, 2017
## 13                             Iron Man       May 2, 2008
## 14                           Iron Man 2       May 7, 2010
## 15                           Iron Man 3       May 3, 2013
## 16                Marvel's The Avengers       May 4, 2012
## 17               Spider-Man: Homecoming      July 7, 2017
## 18                  The Incredible Hulk     June 13, 2008
## 19                                 Thor       May 6, 2011
## 20                      Thor: Ragnarok  November 3, 2017
## 21                 Thor: The Dark World  November 8, 2013
##    US and Canada Box Office Gross Other Territories Box Office Gross
## 1                    $180,202,163                       $339,109,802
## 2                   $216,648,740                                  $406,025,399
## 3                   $459,005,868                                  $946,397,826
## 4                   $678,815,482                                $1,369,544,272
## 5                   $700,059,566                                  $646,853,595
## 6                   $408,084,349                                  $745,220,146
## 7                   $176,654,505                                  $193,915,269
## 8                   $259,766,572                                  $454,497,695
## 9                   $321,498,835                                  $588,800,000
## 10                  $232,641,920                                  $445,076,475
## 11                  $333,176,600                                  $440,152,029
## 12                  $389,813,101                                  $473,942,950
## 13                  $318,412,101                                  $266,762,121
## 14                  $312,433,331                                  $311,500,000
## 15                  $409,013,994                                  $805,797,258
## 16                  $623,357,910                                  $895,455,078
## 17                  $334,201,140                                  $545,965,784
## 18                  $134,806,913                                  $128,620,638
## 19                  $181,030,624                                  $268,295,994
## 20                  $315,058,289                                  $538,918,837
## 21                  $206,362,140                                  $438,209,262
##    Worldwide Box Office Gross US and Canada All-time Ranking
## 1                $519,311,965                            248
## 2                $622,674,139                            170
## 3              $1,405,403,694                             16
## 4              $2,048,359,754                              4
## 5              $1,346,913,161                              3
## 6              $1,153,304,495                             27
## 7                $370,569,774                            262
## 8                $714,264,267                            110
## 9                $910,298,835                             64
## 10               $677,718,395                            145
## 11               $773,328,629                             57
## 12               $863,756,051                             34
## 13               $585,174,222                             66
## 14               $623,933,331                             71
## 15             $1,214,811,252                             26
## 16             $1,518,812,988                              7
## 17               $880,166,924                             55
## 18               $263,427,551                            433
## 19               $449,326,618                            246
## 20               $853,977,126                             70
## 21               $644,571,402                            193
##    Worldwide All_time Ranking            Budget     Ref(s)
## 1                         196     $109.3 million [443][442]
## 2                         140       $162 million [455][456]
## 3                           8     $365.5 million [441][442]
## 4                           4   $316-400 million [453][454]
## 5                           9   $200-210 million [451][452]
## 6                          19       $230 million [444][445]
## 7                         324 $140-216.7 million      [432]
## 8                         105       $177 million [437][438]
## 9                          52  $97.8-152 million [457][458]
## 10                        118 $165-236.6 million [446][447]
## 11                         90     $195.9 million [439][440]
## 12                          66       $200 million      [448]
## 13                         158       $140 million      [428]
## 14                         139       $200 million      [430]
## 15                          17     $178.4 million [434][435]
## 16                           6       $220 million      [433]
## 17                          58       $175 million      [449]
## 18                         542       $150 million      [429]
## 19                         241       $150 million      [431]
## 20                          69       $180 million      [450]
## 21                         130     $152.7 million [436][435]
##           Rotten Tomatoes          Metacritic
## 1  82% (303 reviews)[483] 64 (44 reviews)[484]
## 2  88% (380 reviews)[499] 70 (56 reviews)[500]
## 3  75% (349 reviews)[481] 66 (49 reviews)[482]
## 4  85% (422 reviews)[497] 68 (53 reviews)[498]
## 5  97% (460 reviews)[495] 88 (55 reviews)[496]
## 6  91% (388 reviews)[485] 75 (53 reviews)[486]
## 7  80% (262 reviews)[469] 66 (43 reviews)[470]
## 8  90% (287 reviews)[477] 70 (48 reviews)[478]
## 9  78% (432 reviews)[501] 64 (55 reviews)[502]
## 10 89% (344 reviews)[487] 72 (49 reviews)[488]
## 11 91% (312 reviews)[479] 76 (53 reviews)[480]
## 12 83% (375 reviews)[489] 67 (48 reviews)[490]
## 13 93% (274 reviews)[461] 79 (38 reviews)[462]
## 14 73% (287 reviews)[465] 57 (40 reviews)[466]
## 15 80% (310 reviews)[473] 62 (44 reviews)[474]
## 16 92% (342 reviews)[471] 69 (43 reviews)[472]
## 17 92% (362 reviews)[491] 73 (51 reviews)[492]
## 18 67% (227 reviews)[463] 61 (38 reviews)[464]
## 19 77% (281 reviews)[467] 57 (40 reviews)[468]
## 20 92% (383 reviews)[493] 74 (51 reviews)[494]
## 21 66% (264 reviews)[475] 54 (44 reviews)[476]
```

b)

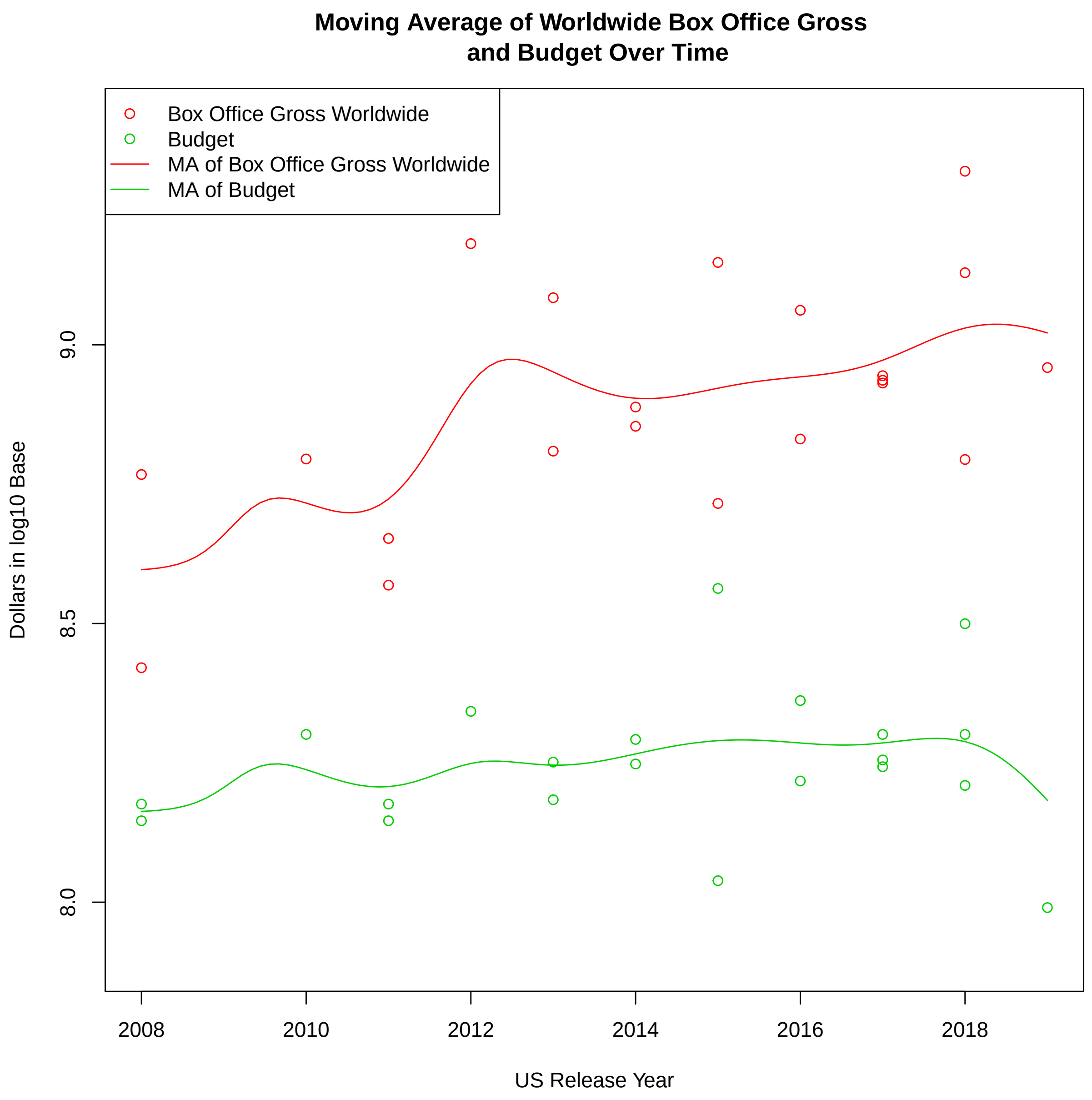
```
suppressPackageStartupMessages(library(tidyverse))
new_table=merge_table%>%select("Film", "Worldwide Box Office Gross",
                              "Budget", "US Release Date",
                              "Rotten Tomatoes", "Metacritic")
#Convert to numeric number
new_table$`Worldwide Box Office Gross`=
  as.numeric(gsub("[[:punct:]]", "", new_table$`Worldwide Box Office Gross`))
new_table$Budget=as.numeric(parse_number(new_table$Budget)*1000000)
#Want the RELEASE YEAR
new_table$`US Release Date`=trimws(str_extract(new_table$`US Release Date`,
                                               "[^(\\,)]+$"))
#Want the numeric number of last two cols
new_table$`Rotten Tomatoes`=as.numeric(str_extract(new_table$`Rotten Tomatoes`,
                                                   "\\d+[^(?%)]"))
new_table$Metacritic=as.numeric(str_extract(new_table$Metacritic,
                                            "\\d+[^[:space:]]"))
```

```r
colnames(new_table)[4]="US Release Year"
head(new_table, 10)
```

```
##                                   Film Worldwide Box Office Gross
## 1                              Ant-Man                  519311965
## 2                 Ant-Man and the Wasp                  622674139
## 3              Avengers: Age of Ultron                 1405403694
## 4               Avengers: Infinity War                 2048359754
## 5                        Black Panther                 1346913161
## 6           Captain America: Civil War                 1153304495
## 7   Captain America: The First Avenger                  370569774
## 8  Captain America: The Winter Soldier                  714264267
## 9                       Captain Marvel                  910298835
## 10                      Doctor Strange                  677718395
##       Budget US Release Year Rotten Tomatoes Metacritic
## 1  109300000            2015              82         64
## 2  162000000            2018              88         70
## 3  365500000            2015              75         66
## 4  316000000            2018              85         68
## 5  200000000            2018              97         88
## 6  230000000            2016              91         75
## 7  140000000            2011              80         66
## 8  177000000            2014              90         70
## 9   97800000            2019              78         64
## 10 165000000            2016              89         72
```

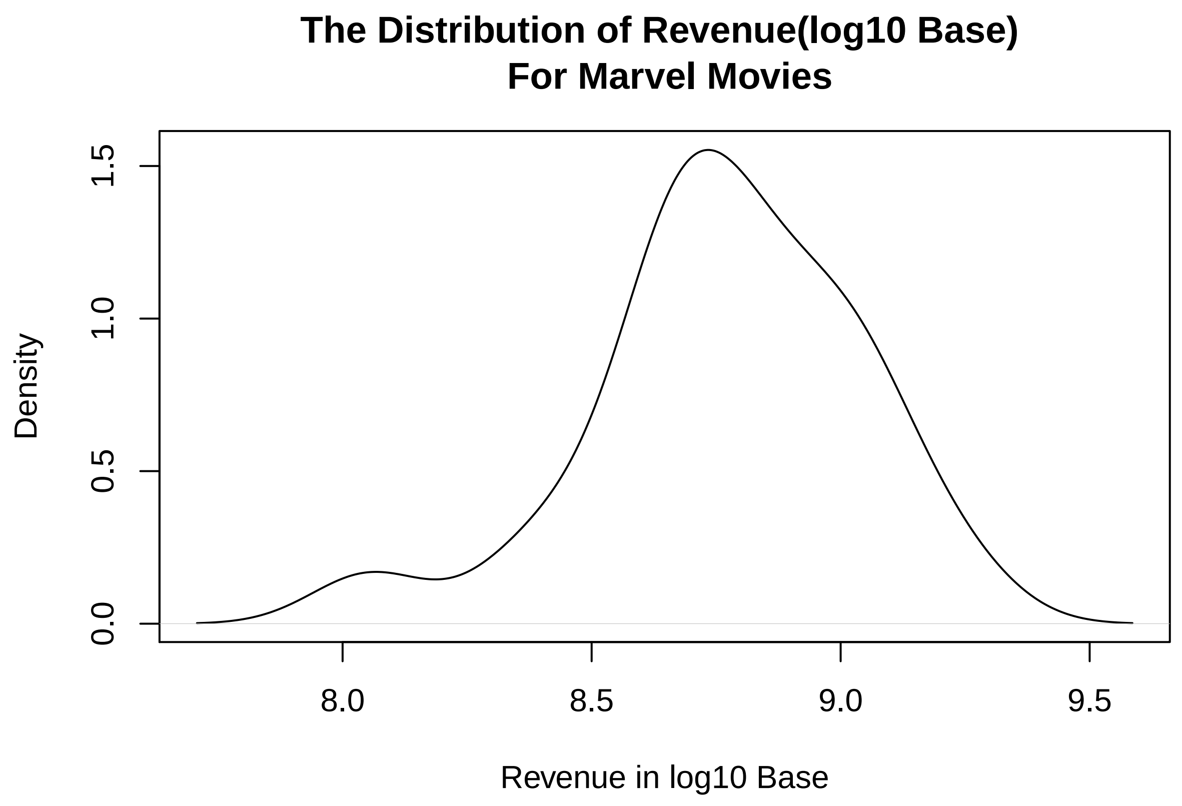
c)

```r
plot(x=new_table$`US Release Year`,
     y=log10(new_table$`Worldwide Box Office Gross`),
     col=2, ylim=c(7.9, 9.4),
     main="Moving Average of Worldwide Box Office Gross \n and Budget Over Time",
     xlab="US Release Year", ylab="Dollars in log10 Base")
points(x=new_table$`US Release Year`,
            y=log10(new_table$Budget), col=3)
lines(ksmooth(x=new_table$`US Release Year`,
            y=log10(new_table$`Worldwide Box Office Gross`),
            bandwidth=2, kernel="normal"), col=2)
lines(ksmooth(x=new_table$`US Release Year`,
            y=log10(new_table$Budget),
            bandwidth=2, kernel="normal"), col=3)
legend("topleft", pch=c(1,1,NA,NA), lty=c(NA,NA,1,1), col=c(2,3,2,3),
       c("Box Office Gross Worldwide", "Budget",
         "MA of Box Office Gross Worldwide", "MA of Budget"))
```



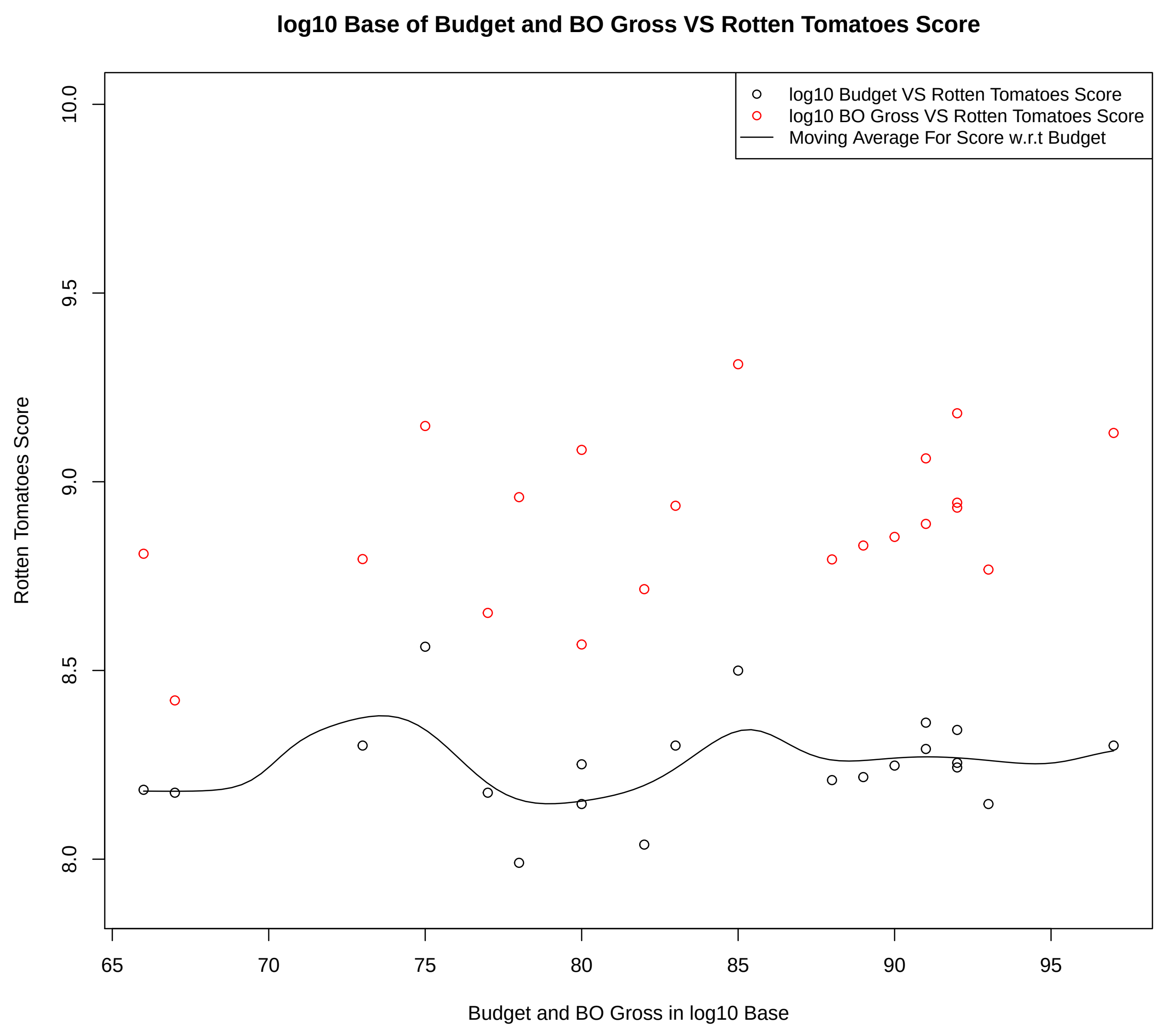
e)

```
#Revenue=Gross-Budget
revenue=new_table$`Worldwide Box Office Gross` - new_table$Budget
plot(density(log10(revenue)), xlab="Revenue in log10 Base",
     main="The Distribution of Revenue(log10 Base) \n For Marvel Movies")
```



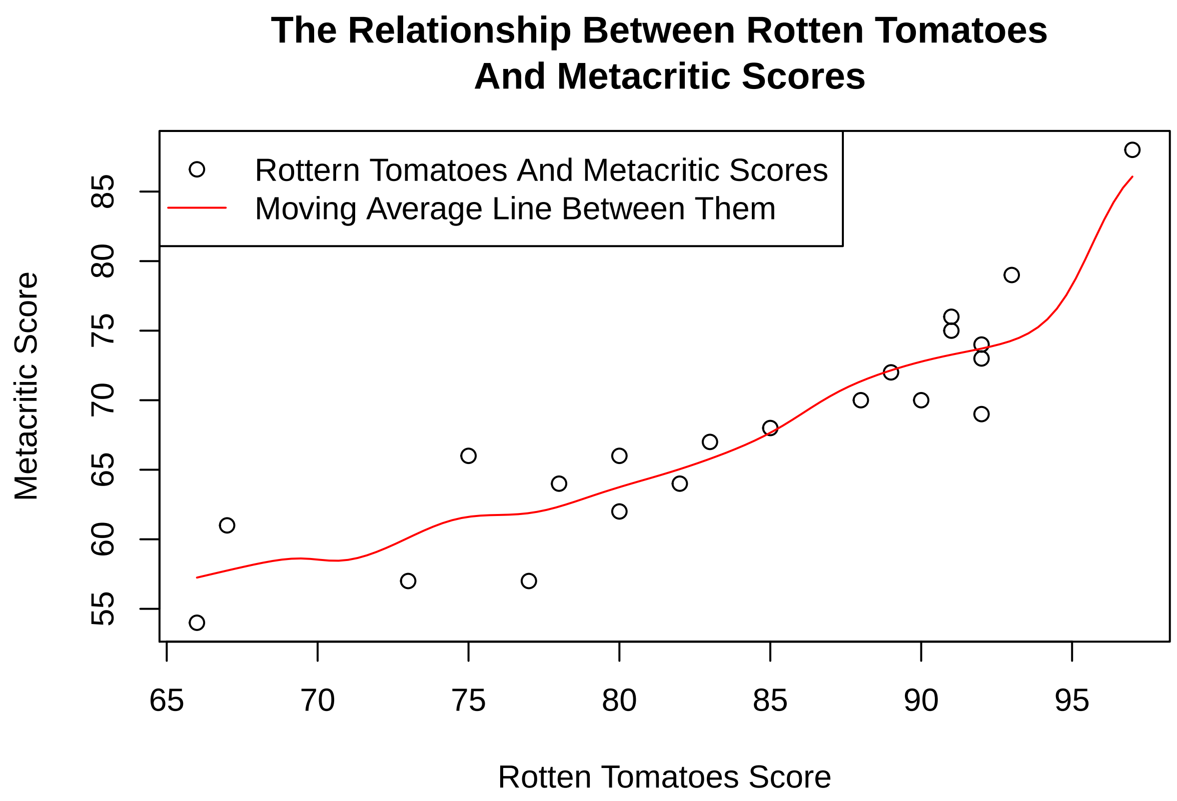
f)

```r
# max(new_table$`Rotten Tomatoes`); min(new_table$`Rotten Tomatoes`)
plot(y=log10(new_table$Budget), x=new_table$`Rotten Tomatoes`,
     ylim=c(7.9, 10), col=1, xlab="Budget and BO Gross in log10 Base",
     ylab="Rotten Tomatoes Score",
     main="log10 Base of Budget and BO Gross VS Rotten Tomatoes Score")
points(y=log10(new_table$`Worldwide Box Office Gross`),
       x=new_table$`Rotten Tomatoes`, col=2)
lines(ksmooth(y=log10(new_table$Budget), x=new_table$`Rotten Tomatoes`,
              bandwidth=5, kernel="normal"))
legend("topright", pch=c(1,1,NA), lty=c(NA,NA,1), col=c(1,2,1),
       c("log10 Budget VS Rotten Tomatoes Score", "log10 BO Gross VS Rotten Tomatoes Score",
         "Moving Average For Score w.r.t Budget "), cex=0.9)
```



g)

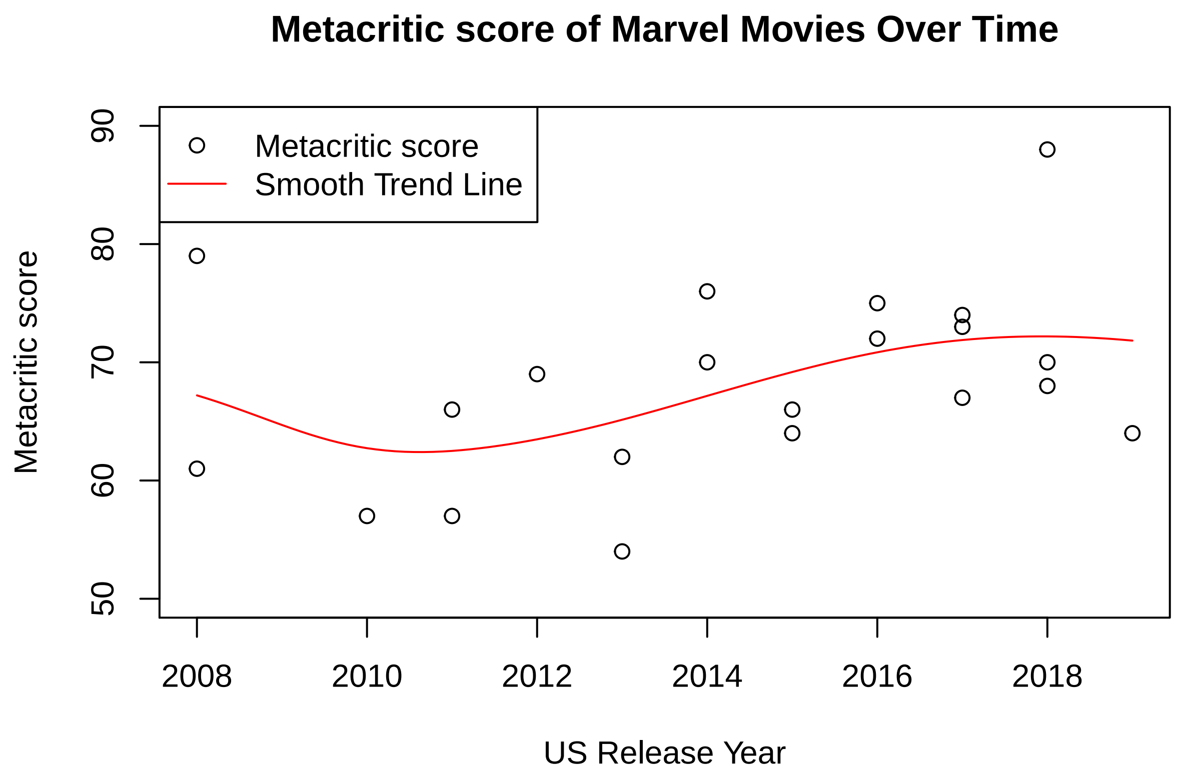
```
plot(new_table$`Rotten Tomatoes`, new_table$Metacritic, xlab="Rotten Tomatoes Score",
     ylab="Metacritic Score",
     main="The Relationship Between Rotten Tomatoes \n And Metacritic Scores")
lines(ksmooth(new_table$`Rotten Tomatoes`, new_table$Metacritic,
              bandwidth=5, kernel="normal"), col=2)
legend("topleft", pch=c(1,NA), col=c(1,2), lty=c(NA,1),
       c("Rottern Tomatoes And Metacritic Scores",
         "Moving Average Line Between Them"))
```



- The plot shows that the Rotten Tomatoes and Metacritic Scores agree with each other, as we can tell by the increasing trend from the graph.

h)

```
plot(new_table$`US Release Year`, new_table$Metacritic, ylim=c(50,90),
     xlab="US Release Year", ylab="Metacritic score",
     main="Metacritic score of Marvel Movies Over Time")
lines(ksmooth(new_table$`US Release Year`, new_table$Metacritic,
              kernel="normal", bandwidth=4), col=2)
legend("topleft", pch=c(1,NA), lty=c(NA,1), col=c(1,2),
       c("Metacritic score", "Smooth Trend Line"))
```


Metacritic score of Marvel Movies Over Time
Metacritic score
Smooth Trend Line
Metacritic score
90
80
70
60
50
2008
2010
2012
2014
2016
2018
US Release Year