STAT 240 - Assignment 6

Problem 3

```r
library(rvest)
library(stringr)
library(zoo)
```

1

```r
# scrape box office performance & critical & public response tables
# obtain the tables in a single data frame
url = "https://en.wikipedia.org/wiki/List_of_Marvel_Cinematic_Universe_films"
url_table = read_html(url)
length(html_nodes(url_table, "table"))
```

```
## [1] 30
```

```r
# box office performance
performance = html_table(html_nodes(url_table, "table")[[6]])
# critical & public response
response = html_table(html_nodes(url_table, "table")[[7]])
# clean tables
performance = performance[performance[, "Film"]!="Phase One" &
                            performance[, "Film"]!="Phase Two" &
                            performance[, "Film"]!="Phase Three", ]
performance = performance[3:25, ]
response = response[response[, "Film"]!="Phase One" &
                      response[, "Film"]!="Phase Two" &
                      response[, "Film"]!="Phase Three", ]
response = response[3:25, ]
# merge tables
marvel_df = merge(performance, response,
                  by.x="Film", by.y="Film")
# can not print data frame in LaTeX b/c Public CinemaScore col contains minus sign
#head(marvel_df)
# however, this works if we exclude last col
#head(marvel_df[1:11])
```

2

```
names(marvel_df)
```

```
##  [1] "Film"              "U.S. release date" "Box office gross" 
##  [4] "Box office gross"  "Box office gross"  "All-time ranking" 
##  [7] "All-time ranking"  "Budget"            "Ref(s)"           
## [10] "Critical"          "Critical.1"        "Public"
```

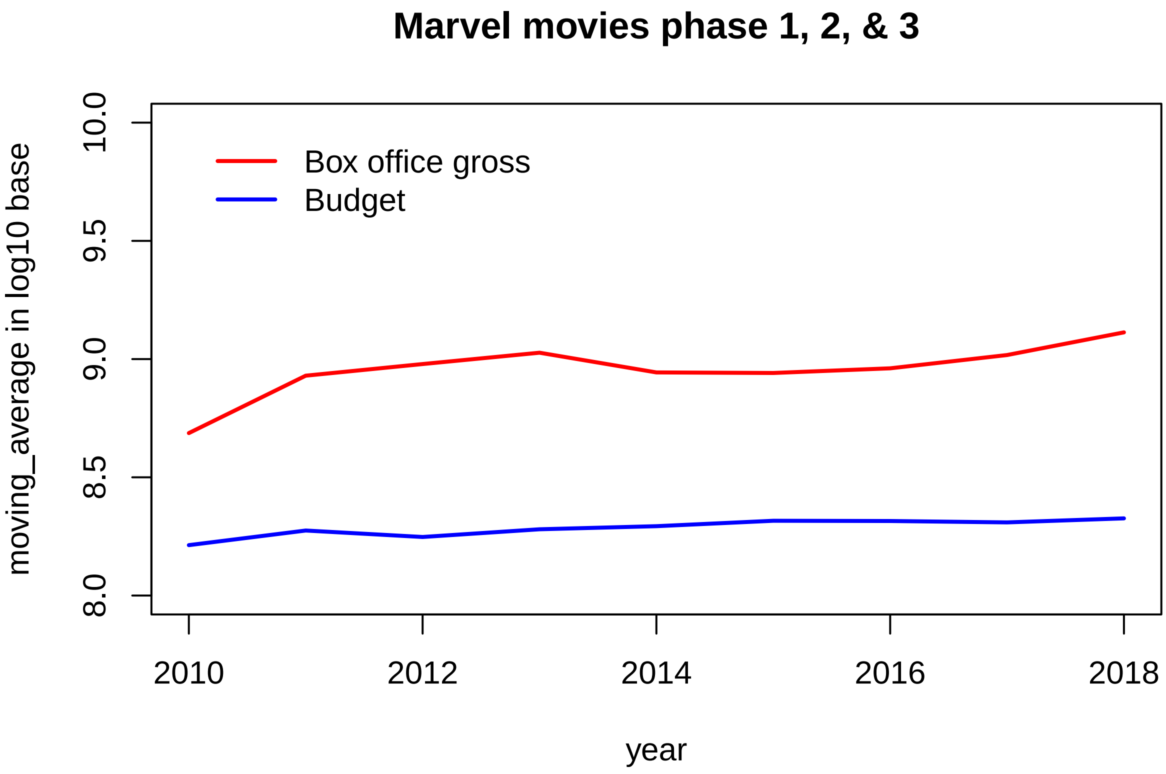
```
# rename cols
names(marvel_df)[2] = "Year"
names(marvel_df)[5] = "Worldwide Box Office Gross"
names(marvel_df)[10] = "Rotten Tomatoes"
names(marvel_df)[11] = "Metacritic Scores"
# new data-frame
marvel_movies = marvel_df[, c(1, 2, 5, 8, 10, 11)]
# change Release year to numeric
marvel_movies$`Year` = as.numeric(
  str_replace(marvel_movies$`Year`, ".+,\\s", "")
  )
# change Worldwide Box office gross to numeric
marvel_movies$`Worldwide Box Office Gross` = as.numeric(
  str_replace_all(marvel_movies$`Worldwide Box Office Gross`, "\\$|,", "")
  )
# change Budget to numeric, taking lower-bound value
marvel_movies$Budget = as.numeric(
  str_extract(marvel_movies$Budget, "\\d+\\.?\\d+?\\b")
  ) * 1000000 # convert to million
# change Rotten Tomatoes to numeric
marvel_movies$`Rotten Tomatoes` = as.numeric(
  str_extract(marvel_movies$`Rotten Tomatoes`, "\\d+\\b")
  )
# change Metacritic Scores to numeric
marvel_movies$`Metacritic Scores` = as.numeric(
  str_extract(marvel_movies$`Metacritic Scores`, "\\d+\\b")
  )
marvel_movies[1:10, ]
```

```
##                                   Film Year Worldwide Box Office Gross
## 1                              Ant-Man 2015                  519311965
## 2                 Ant-Man and the Wasp 2018                  622674139
## 3              Avengers: Age of Ultron 2015                 1402805868
## 4                    Avengers: Endgame 2019                 2797800564
## 5               Avengers: Infinity War 2018                 2048359754
## 6                        Black Panther 2018                 1347280161
## 7           Captain America: Civil War 2016                 1153296293
## 8   Captain America: The First Avenger 2011                  370569774
## 9  Captain America: The Winter Soldier 2014                  714421503
## 10                      Captain Marvel 2019                 1128275263
##       Budget Rotten Tomatoes Metacritic Scores
## 1  109300000              83                64
## 2  162000000              87                70
## 3  365500000              76                66
## 4  356000000              94                78
## 5  325000000             85              68
## 6  200000000             96              88
## 7  230000000             90              75
## 8  140000000             80              66
## 9  177000000             90              70
## 10 150000000             79              64
```

3

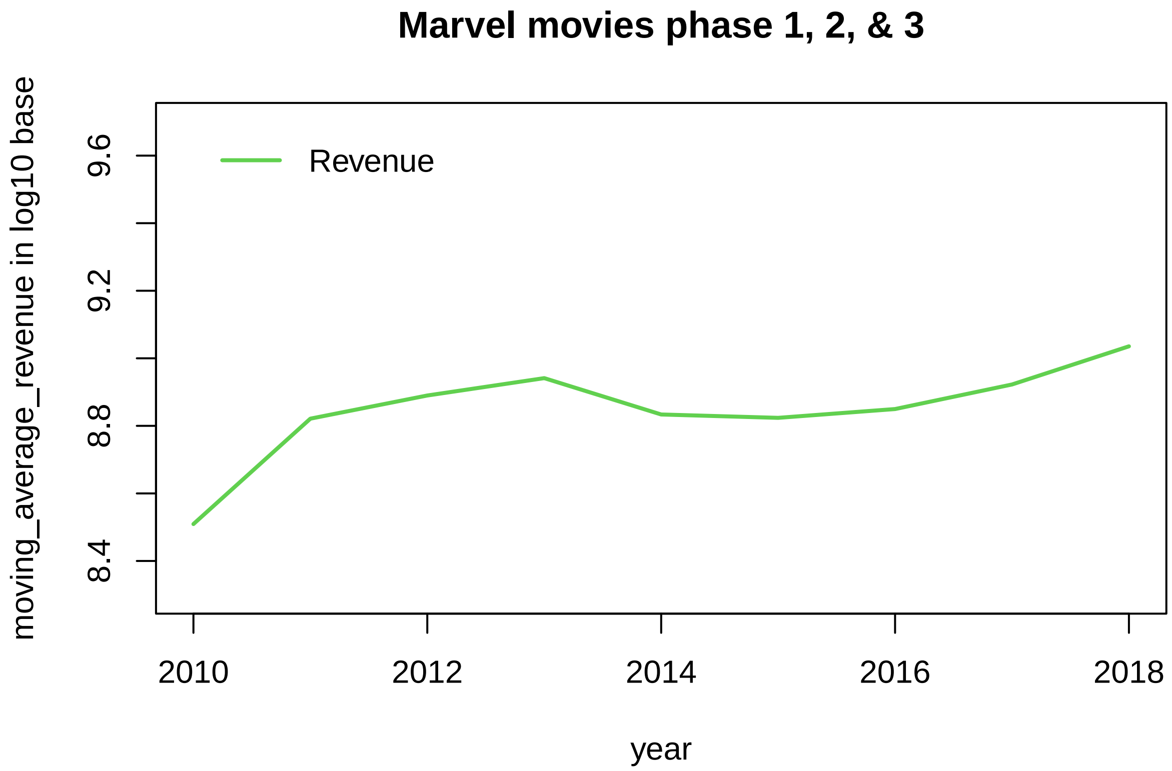
```r
# plot moving avg of Worldwide-Box-office-gross & budget vs. time in a single plot
# for clarity, use log10 for dollar amounts
# moving average of box-office-gross & budget
max_year = max(marvel_movies$Year)
min_year = min(marvel_movies$Year)
box_office_ma = vector(mode="numeric")
budget_ma = vector(mode="numeric")
year = vector(mode="numeric")
for(i in min_year:max_year) {
  temp_vec = vector(mode="numeric")
  for(j in 1:length(marvel_movies[[1]])) {
    if(marvel_movies$Year[j] == i) {
      temp_vec[length(temp_vec)+1] = j
    }
  }
  if(length(temp_vec) > 1) {
    tot_bud = 0
    tot_box = 0
    for(j in 1:length(temp_vec)){
      tot_bud = tot_bud + marvel_movies$Budget[temp_vec[j]]
      tot_box = tot_box +marvel_movies$`Worldwide Box Office Gross`[temp_vec[j]]
    }
    budget_ma[length(budget_ma)+1] = tot_bud / length(temp_vec)
    box_office_ma[length(box_office_ma)+1] = tot_box / length(temp_vec)
    year[length(year)+1] = i
  }
  else if(length(temp_vec) > 0 ) {
    budget_ma[length(budget_ma)+1] = marvel_movies$Budget[temp_vec[1]]
    box_office_ma[length(box_office_ma)+1] = marvel_movies$`Worldwide Box Office Gross`[temp_vec[1]]
    year[length(year)+1] = i
  }
}
# using interval length == 3
budget_ma = rollmean(budget_ma, k=3)
box_office_ma = rollmean(box_office_ma, k=3)
# plot
# exclude first and last index of year b/c moving average interval == 3
plot(year[2:(length(year)-1)], log10(box_office_ma), type="l",
     main="Marvel movies phase 1, 2, & 3",
     xlab="year", ylab="moving_average in log10 base",
     ylim=c(8, 10),
     col="red", lwd=2)
```

```
# exclude first and last index of year b/c moving average interval == 3
lines(year[2:(length(year)-1)], log10(budget_ma),
      col="blue", lwd=2)
legend(2010, 10, c('Box office gross', 'Budget'),  bty="n", col=c('red', 'blue')
       , lty=1, lwd=2)
```



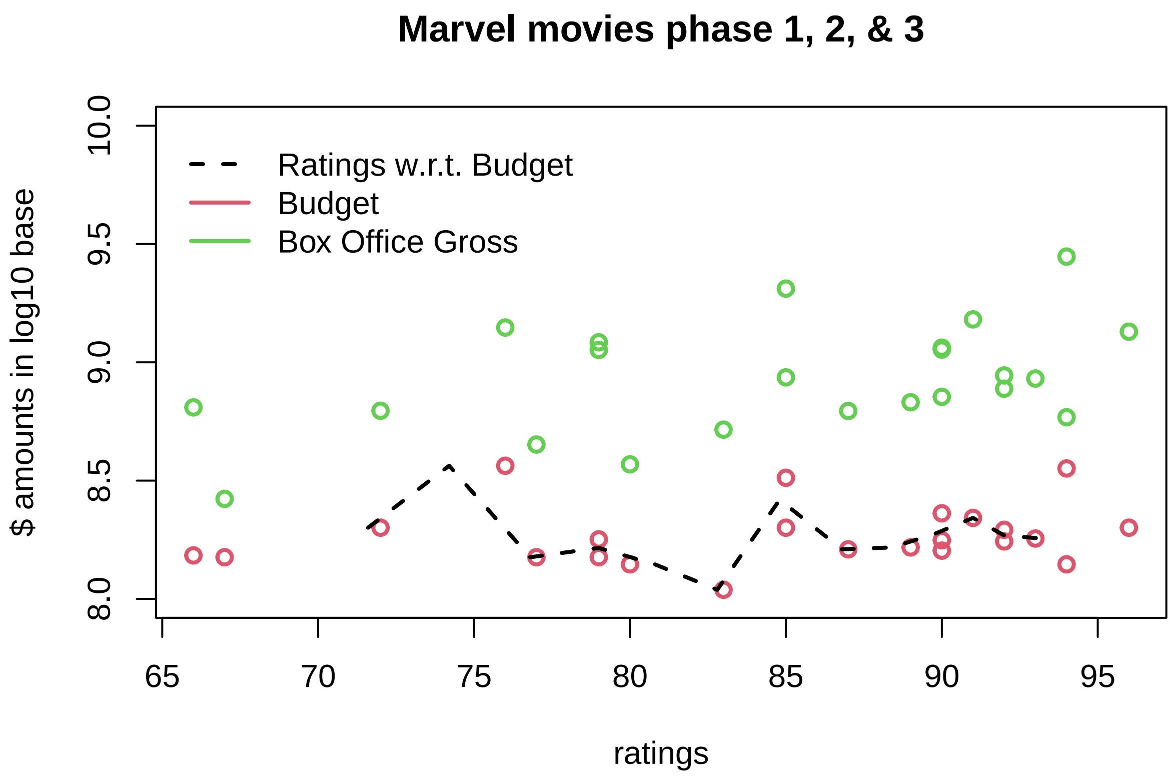
4

```
# plot log10(revenue) over time for marvel movies
revenue = vector(mode="numeric", length=length(box_office_ma))
for(i in 1:length(revenue)) {
  revenue[i] = box_office_ma[i] - budget_ma[i]
}
# exclude first and last index of year
plot(year[2:(length(year)-1)], log10(revenue), type="l",
     main="Marvel movies phase 1, 2, & 3",
     xlab="year", ylab="moving_average_revenue in log10 base",
     ylim=c(8.3, 9.7),
     col=3, lwd=2)
legend(2010, 9.7, c('Revenue'),  bty="n", col=3, lty=1, lwd=2)
```



5

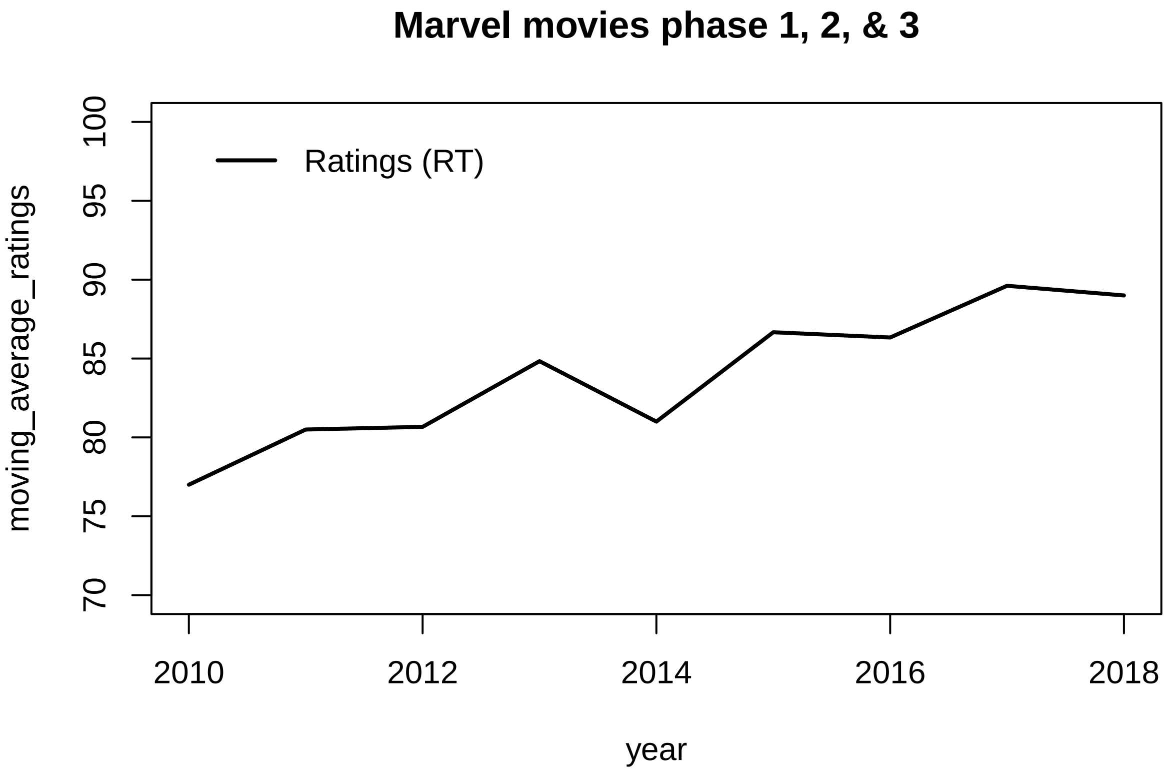
```r
# plot log10(budget_ma) & log10(box_office_ma) vs. Rotten Tomatoes
# include moving average for Rotten Tomatoes ratings w.r.t. budget
# moving average of rotten-tomatoes w.r.t. budget
max_rat = max(marvel_movies$`Rotten Tomatoes`)
min_rat = min(marvel_movies$`Rotten Tomatoes`)
ratings2 = vector(mode="numeric")
budget_ma2 = vector(mode="numeric")
for(i in min_rat:max_rat) {
  temp_vec = vector(mode="numeric")
  for(j in 1:length(marvel_movies$`Rotten Tomatoes`)) {
    if(marvel_movies$`Rotten Tomatoes`[j] == i) {
      temp_vec[length(temp_vec)+1] = j
      }
    }
    if(length(temp_vec) > 1) {
      tot_bud = 0
      for(k in 1:length(temp_vec)) {
        tot_bud = tot_bud + marvel_movies$Budget[temp_vec[k]]
      }
      ratings2[length(ratings2)+1] = i
      budget_ma2[length(budget_ma2)+1] = tot_bud / length(temp_vec)
    }
    else if(length(temp_vec) > 0) {
      ratings2[length(ratings2)+1] = i
      budget_ma2[length(budget_ma2)+1] = marvel_movies$Budget[temp_vec[1]]
    }
}
# using interval length == 5
ratings2_ma = rollmean(ratings2, k=5)
# plot
plot(marvel_movies$`Rotten Tomatoes`, log10(marvel_movies$Budget), type="p",
     main="Marvel movies phase 1, 2, & 3",
     xlab="ratings", ylab="$ amounts in log10 base",
     ylim=c(8, 10), col=2, lwd=2)
points(marvel_movies$`Rotten Tomatoes`, log10(marvel_movies$`Worldwide Box Office Gross`),
       col=3, lwd=2)
lines(ratings2_ma, log10(budget_ma2[3:(length(budget_ma2)-2)]),
      col=1, lty=2, lwd=2)
legend(65, 10, c('Ratings w.r.t. Budget', 'Budget', 'Box Office Gross'),  bty="n", col=c(1, 2, 3), lty=
```



6

```
# plot ratings vs . time
# moving average of rotten-tomatoes w.r.t. time
```

```r
ratings1 = vector(mode="numeric")
for(i in min_year:max_year) {
  temp_vec = vector(mode="numeric")
  for(j in 1:length(marvel_movies[[1]])) {
    if(marvel_movies$Year[j] == i) {
      temp_vec[length(temp_vec)+1] = j
    }
  }
  if(length(temp_vec) > 1) {
    tot_rat = 0
    for(j in 1:length(temp_vec)){
      tot_rat = tot_rat + marvel_movies$`Rotten Tomatoes`[temp_vec[j]]
    }
    ratings1[length(ratings1)+1] = tot_rat / length(temp_vec)
  }
  else if(length(temp_vec) > 0 ) {
    ratings1[length(ratings1)+1] = marvel_movies$`Rotten Tomatoes`[temp_vec[1]]
  }
}
# using interval length == 3
ratings1_ma = rollmean(ratings1, k=3)
# plot
# exclude first and last index of year b/c moving average interval == 3
plot(year[2:(length(year)-1)], ratings1_ma, type="l",
     main="Marvel movies phase 1, 2, & 3",
     xlab="year", ylab="moving_average_ratings",
     ylim=c(70, 100), lwd=2)
legend(2010, 100, c('Ratings (RT)'),  bty="n", lwd=2)
```


Marvel movies phase 1, 2, & 3
Ratings (RT)
moving_average_ratings
100
95
90
85
80
75
70
2010
2012
2014
2016
2018
year