labQ3.R

mirrien

2022-03-16

suppressWarnings(library(rvest))  
suppressWarnings(library(lubridate))

suppressWarnings(library(zoo))

suppressWarnings(library(xml2))  
suppressWarnings(library(tidyverse))

#################################################  
# Q3.1  
  
marvel\_url = "https://en.wikipedia.org/wiki/List\_of\_Marvel\_Cinematic\_Universe\_films"  
  
marvel = read\_html(marvel\_url)  
  
length(html\_nodes(marvel, "table"))

## [1] 30

bop <- html\_table(html\_nodes(marvel, "table")[6])  
bop <- bop[[1]]  
  
cpr <- html\_table(html\_nodes(marvel, "table")[7])  
cpr <- cpr[[1]]  
  
# Tidy col\_names  
for (i in seq\_along(names(bop))) {  
 if (names(bop)[i] != bop[[i]][1]) {  
 colnames(bop)[i] <- paste(bop[[i]][1], names(bop)[i], collapse = " ")  
 }  
}  
  
for (i in seq\_along(names(cpr))) {  
 if (names(cpr)[i] != cpr[[i]][1]) {  
 colnames(cpr)[i] <- paste(cpr[[i]][1], names(cpr)[i], collapse = " ")  
 }  
}

# Tidy rows  
# Delete empty rows, phase indicator rows, and the total row from the table bop  
  
bop <- bop[-c(1,2,3,nrow(bop)), ]  
bop <- bop[!grepl("Phase\\s",bop$Film),]  
  
cpr <- cpr[-c(1,2,3), ]  
cpr <- cpr[!grepl("Phase\\s",cpr$Film),]  
  
df <- merge(bop, cpr, by.x = "Film", by.y = "Film",sort = F)  
  
head(df)

## Film U.S. release date  
## 1 Iron Man May 2, 2008  
## 2 The Incredible Hulk June 13, 2008  
## 3 Iron Man 2 May 7, 2010  
## 4 Thor May 6, 2011  
## 5 Captain America: The First Avenger July 22, 2011  
## 6 Marvel's The Avengers May 4, 2012  
## U.S. and Canada Box office gross Other territories Box office gross  
## 1 $319,034,126 $266,762,121  
## 2 $134,806,913 $129,964,083  
## 3 $312,433,331 $311,500,000  
## 4 $181,030,624 $268,295,994  
## 5 $176,654,505 $193,915,269  
## 6 $623,357,910 $895,457,605  
## Worldwide Box office gross U.S. and Canada All-time ranking  
## 1 $585,796,247 74  
## 2 $264,770,996 454  
## 3 $623,933,331 80  
## 4 $449,326,618 257  
## 5 $370,569,774 273  
## 6 $1,518,815,515 8  
## Worldwide All-time ranking Budget Ref(s)  
## 1 170 $140 million [267]  
## 2 573 $150 million [268]  
## 3 151 $200 million [269]  
## 4 256 $150 million [270]  
## 5 348 $140 million [271]  
## 6 8 $220 million [272]  
## Rotten Tomatoes Critical Metacritic Critical CinemaScore[312] Public  
## 1 94% (281 reviews)[313] 79 (38 reviews)[314] A  
## 2 67% (238 reviews)[315] 61 (38 reviews)[316] A-  
## 3 72% (304 reviews)[317] 57 (40 reviews)[318] A  
## 4 77% (291 reviews)[319] 57 (40 reviews)[320] B+  
## 5 80% (274 reviews)[321] 66 (43 reviews)[322] A-  
## 6 91% (362 reviews)[323] 69 (43 reviews)[324] A+

#################################################  
# Q3.2  
  
# Tidy data, convert to corresponding types  
  
df$`U.S. and Canada Box office gross` <-  
 as.numeric(gsub("\\D","",df$`U.S. and Canada Box office gross`))  
   
df$`Other territories Box office gross` <-  
 as.numeric(gsub("\\D","",df$`Other territories Box office gross`))  
  
df$`Worldwide Box office gross` <-  
 as.numeric(gsub("\\D","",df$`Worldwide Box office gross`))  
  
df$Budget <- gsub("\\D","",df$Budget)  
  
# if Budget is a range (i.e., nchar==6), split string and find mean  
for (i in seq\_along(df$Budget)) {  
 if (nchar(df$Budget[i]) == 6) {  
 df$Budget[i] <- gsub("\\D","",  
 mean(c(as.numeric(str\_sub(df$Budget[i],1,3)),  
 as.numeric(str\_sub(df$Budget[i],-3,-1))  
 )  
 )  
 )  
 }  
}  
  
# if Budget has no decimal, replace "million" with "000000"  
# if Budget has one decimal, remove ".", replace "million with "00000"  
for (i in seq\_along(df$Budget)){  
 if (nchar(df$Budget[i]) == 3) {  
 df$Budget[i] <- paste0(df$Budget[i],"000000",collapse = "")  
 }  
 else if (nchar(df$Budget[i]) == 4) {  
 df$Budget[i] <- paste0(df$Budget[i],"00000",collapse = "")  
 }  
}  
  
# Convert to integer  
df$Budget <- as.integer(df$Budget)  
  
# convert Rotten Tomatoes score (%) into doubles (e.g., 0.90)  
df$`Rotten Tomatoes Critical` <- as.numeric(gsub("\\%\\s.\*\\]$","",df$`Rotten Tomatoes Critical`))/100  
  
# Convert Matecritic score into integers  
df$`Metacritic Critical` <- as.numeric(gsub("\\s.\*\\]$","",df$`Metacritic Critical`))  
  
# Select required columns, extract only years  
df1 <- df %>% select(`Worldwide Box office gross`,  
 Budget,  
 `Rotten Tomatoes Critical`,  
 `Metacritic Critical`  
 ) %>%  
 mutate(Year = year(mdy(df$`U.S. release date`)))  
  
# Reorder the result data frame  
df1 <- df1[,c(1,2,5,3,4)]  
  
# print the first 10 rows  
head(df1,n=10)

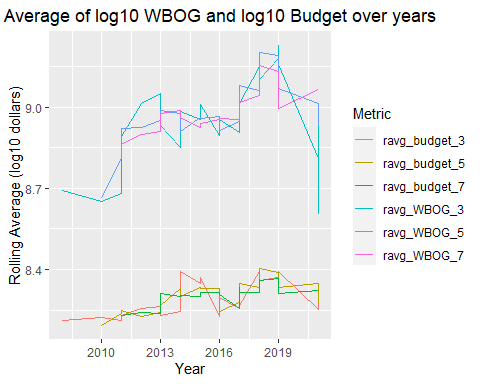
## Worldwide Box office gross Budget Year Rotten Tomatoes Critical  
## 1 585796247 140000000 2008 0.94  
## 2 264770996 150000000 2008 0.67  
## 3 623933331 200000000 2010 0.72  
## 4 449326618 150000000 2011 0.77  
## 5 370569774 140000000 2011 0.80  
## 6 1518815515 220000000 2012 0.91  
## 7 1214811252 178400000 2013 0.79  
## 8 644783140 152700000 2013 0.66  
## 9 714421503 177000000 2014 0.90  
## 10 773350147 195900000 2014 0.92  
## Metacritic Critical  
## 1 79  
## 2 61  
## 3 57  
## 4 57  
## 5 66  
## 6 69  
## 7 62  
## 8 54  
## 9 70  
## 10 76

#################################################  
# Q3.3  
  
# The question was confusing.  
# First, is the "time" in year or in date? In the first part I will assume  
# it in year. In the second part I will assume it in date.  
  
# Second, the moving averages could be understood in 2 ways:  
  
# (1) moving averages for each group of years, where we can use group\_by()  
# and calculate means for each group of years. By doing this, we can have one  
# mean for each year for each variable, and its graph will not contain  
# any vertical line segment (i.e., one x will only have one y).  
  
# (2) rolling averages with a certain rolling window width such as 3, 5,  
# or 7 which is not given in the question. In this case, we may have multiple  
# means for one year and many NAs, which results in a plot with many vertical  
# line segments. The plot will be too jagged.  
  
# Part I.  
  
# (1) Create a new data frame for moving averages for each group of years  
df2 <- df1 %>%  
 group\_by(Year) %>%  
 summarise(average\_budget = mean(Budget),  
 average\_gross = mean(`Worldwide Box office gross`))  
  
# use base r  
windows(7,7)  
plot(x=df2$Year,y=log10(df2$average\_gross),  
 type = "l",  
 ylim = c(8,9.5),  
 xlab = "Year",  
 ylab = "Log10 Dollars ($)",  
 main = "Moving Average of log WBOG and log Budget over Years",  
 col = 1)  
lines(x=df2$Year,y=log10(df2$average\_budget),col=2)  
legend(2008, 9.5, c("Average WBOG","Average Budget"),  
 col=1:2, lty=c(1,1), cex=0.8)



# (2) Create a new data frame for rolling average and use rollmean()  
# Since the rolling window width is not given, we will try k=3, 5, and 7.  
# Though we have a very small data set (n=27), k=3 is preferable.  
  
# Note that the table has an ascending year already  
  
df2\_2 <- df1 %>%  
 mutate(ravg\_budget\_3 = rollmean(Budget,3,fill=NA),  
 ravg\_budget\_5 = rollmean(Budget,5,fill=NA),  
 ravg\_budget\_7 = rollmean(Budget,7,fill=NA),  
 ravg\_WBOG\_3 = rollmean(`Worldwide Box office gross`,3,fill=NA),  
 ravg\_WBOG\_5 = rollmean(`Worldwide Box office gross`,5,fill=NA),  
 ravg\_WBOG\_7 = rollmean(`Worldwide Box office gross`,7,fill=NA)) %>%  
 select(!contains("Critical"))  
  
# use ggplot to plot rolling average of all three widths for budgets and WBOG  
# tidy the data first:  
df2\_2 %>% pivot\_longer(names\_to = "rolling\_mean\_key",  
 values\_to = "rolling\_mean\_value",  
 cols = c(ravg\_budget\_3,  
 ravg\_budget\_5,  
 ravg\_budget\_7,  
 ravg\_WBOG\_3,  
 ravg\_WBOG\_5,  
 ravg\_WBOG\_7)) %>%  
 ggplot(aes(x = Year,  
 y = log10(rolling\_mean\_value),  
 color = rolling\_mean\_key)) +  
 geom\_line() +  
 labs(color = "Metric",  
 x = "Year",  
 y = "Rolling Average (log10 dollars)",  
 title = "Rolling Average of log10 WBOG and log10 Budget over years") +  
 theme(plot.title = element\_text(hjust = 0.5))

## Warning: Removed 24 row(s) containing missing values (geom\_path).



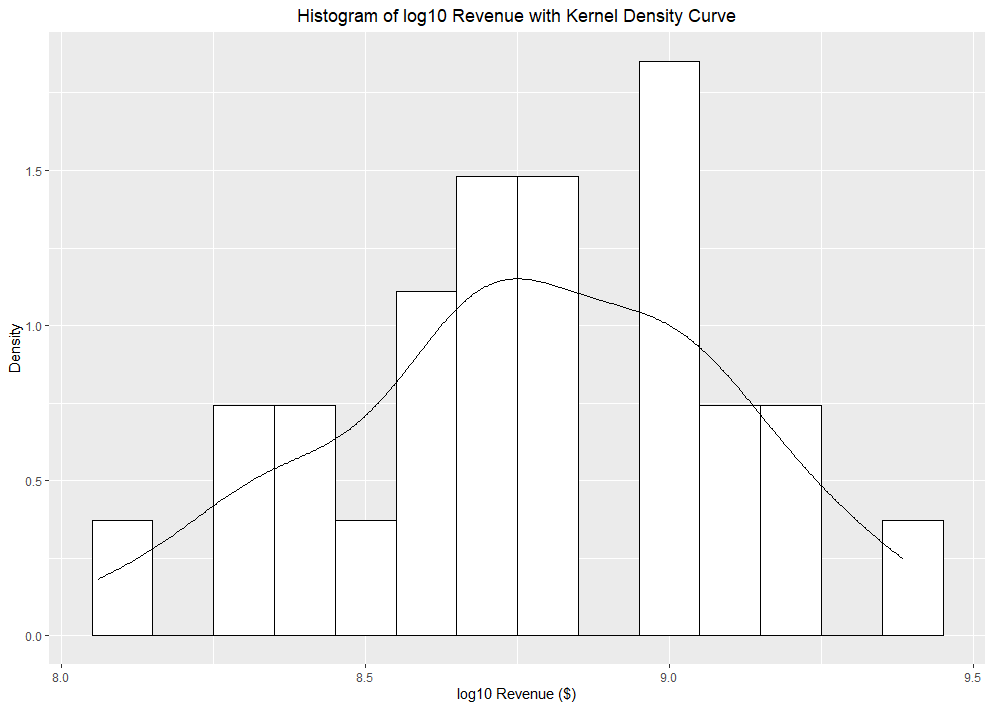
# Reference: https://www.storybench.org/how-to-calculate-a-rolling-average-in-r/  
  
  
# use base r to plot rolling averages with k=3  
windows(7,7)  
plot(x=df2\_2$Year,y=log10(df2\_2$ravg\_WBOG\_3),  
 type = "l",  
 ylim = c(8,9.5),  
 xlab = "Year",  
 ylab = "Log10 Dollars ($)",  
 main = "Rolling Average of log10 WBOG and log10 Budget over Years (k=3)",  
 col = 1)  
lines(x=df2\_2$Year,y=log10(df2\_2$ravg\_budget\_3),col=2)  
legend(2008, 9.5, c("WBOG","Budget"),  
 col=1:2, lty=c(1,1), cex=0.8)



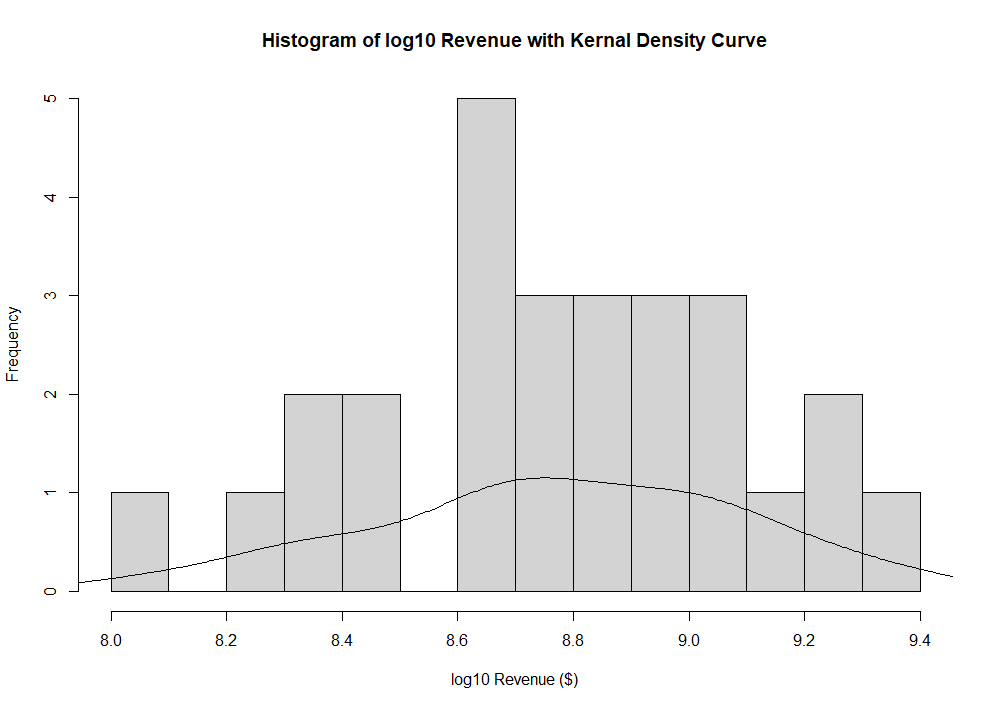
# Part II.  
# Assume that we take "Date" instead of "Year".  
# Also assume that moving averages is rolling averages.  
  
windows(8,6)  
df2\_3 <- df %>%  
 mutate(ravg\_budget\_3 = rollmean(Budget,3,fill=NA),  
 ravg\_WBOG\_3 = rollmean(`Worldwide Box office gross`,3,fill=NA),  
 date = mdy(`U.S. release date`)) %>%  
 select(date,ravg\_budget\_3,ravg\_WBOG\_3,)  
  
plot(x=df2\_3$date,y=log10(df2\_3$ravg\_WBOG\_3),  
 type = "l",  
 ylim = c(8,9.5),  
 xlab = "Date",  
 ylab = "Log10 Dollars $",  
 main = "Rolling Average of log10 WBOG and log10 Budget over Date (k=3)",  
 col = 1)  
lines(x=df2\_3$date,y=log10(df2\_3$ravg\_budget\_3),col=2)  
legend("topleft", c("WBOG","Budget"),  
 col=1:2, lty=c(1,1), cex=1)



#################################################  
# Q3.4  
  
# create a new data frame for revenue  
df3<- df1 %>%  
 mutate(Revenue = `Worldwide Box office gross` - Budget)  
  
# The following two methods both show the distribution of revenue for each film  
  
# 1. use ggplot to plot histogram overlaid with kernel density curve  
# Note that we use density instead of count on y-axis  
  
ggplot(df3, aes(x=log10(Revenue))) +   
 geom\_histogram(aes(y=..density..),   
 binwidth=.1,  
 colour="black", fill="white") +  
 geom\_density(alpha=.2) +  
 labs(x = "log10 Revenue ($)", y = "Density",  
 title = "Histogram of log10 Revenue with Kernel Density Curve") +  
 theme(plot.title = element\_text(hjust = 0.5))



# 2. use base r to plot histogram overlaid with kernel density curve  
# In this plot, we use count instead of density on y-axis  
  
hist(log10(df3$Revenue),  
 breaks=12,  
 xlab="log10 Revenue ($)",  
 main = "Histogram of log10 Revenue with Kernal Density Curve")  
lines(density(log10(df3$Revenue)))



#################################################  
# Q3.5  
  
# The relationship between budget and Rotten Tomatoes (RT) scores is not  
# intuitive. The visualization helps.  
  
# Note that the question requires a moving average for ratings over budget,  
# we assume here the moving average refers to a rolling average, otherwise  
# it would be meaningless to group by budget.  
# We will be using a window of width k=3.  
  
# New data frame with Budget sorted with respect to the scores  
# Mutate a new column with rolling means for RT ratings  
df4 <- df3 %>% select(-c(`Metacritic Critical`,Revenue)) %>%  
 arrange(Budget) %>% # sort budget to establish its relationship with ratings  
 mutate(ravg\_score\_3 = rollmean(`Rotten Tomatoes Critical`,3,fill=NA))  
  
# Additional data frame with WBOG sorted with respect to the scores  
df4\_2 <- df4 %>% arrange(`Worldwide Box office gross`)  
  
# Note that the original ratings are in percentage form and were  
# converted into doubles (i.e., 90% -> 0.90). For clarity, we will multiply  
# them by 100 and include its unit (%).  
  
windows(10,6)  
plot(x=log10(df4$Budget),y=100\*(df4$ravg\_score\_3),  
 type = "l",  
 xlim = c(8,9.5),  
 ylim = c(45,100),  
 xlab = "Log10 Dollars ($)",  
 ylab = "RT scores (%)",  
 main = "Log10 Budget and Log10 WBOG vs RT Score Overlaid with   
 Moving Average for Ratings to Budget ",  
 col = 1,  
 lwd = 2)  
  
# The question did not request a point or a line plot, where we will plot both.  
# If line plot, we need to make sure x-axis is sorted with respect to the score  
lines(x=log10(df4$Budget),  
 y=100\*(df4$`Rotten Tomatoes Critical`),col=2)  
  
points(x=log10(df4$Budget),  
 y=100\*(df4$`Rotten Tomatoes Critical`),  
 col=2,  
 cex = 0.5)  
  
  
lines(x=log10(df4\_2$`Worldwide Box office gross`),  
 y=100\*(df4\_2$`Rotten Tomatoes Critical`),col=3)  
  
points(x=log10(df4\_2$`Worldwide Box office gross`),  
 y=100\*(df4\_2$`Rotten Tomatoes Critical`),  
 col=3,  
 cex = 0.5)  
  
  
legend("bottomright",  
 c("RAVG RT Score","Budget","WBOG"),  
 col=c(1,2,3),  
 lty=c(1,1,1),  
 pch = c(NA,1,1),  
 lwd = c(2,1,1),  
 cex=0.8)



#################################################  
# Q3.6  
# First we plot points of RT and Metacritic ratings vs time (assumed to be date)  
df5 <- df %>% mutate(date = mdy(`U.S. release date`)) %>%  
 select(date,`Rotten Tomatoes Critical`,`Metacritic Critical`)  
  
windows(10,6)  
plot(df5$date,100\*df5$`Rotten Tomatoes Critical`,  
 xlab = "Date",  
 ylab = "Rating Scores",  
 col=1,  
 ylim=c(40,100))  
points(df5$date,df5$`Metacritic Critical`,  
 col = 2,  
 pch = 3)

lines(df5$date,100\*df5$`Rotten Tomatoes Critical`,  
 col=1)  
lines(df5$date,df5$`Metacritic Critical`,  
 col = 2,  
 lty = 2)  
  
legend("bottomleft",  
 c("RT Ratings", "Metacritic Ratings"),  
 col = c(1,2),  
 pch = c(1,3),  
 lty = c(1,2))



# From the plot, it's difficult to describe a general pattern of the   
# relationship between ratings and time as the scores fluctuates over time.  
# But the ratings from two companies follow similar trends.