Sentiment Analysis Using R over Movie Tweets: Report

Arghyadeep Sen

1864004

KIIT University

**Table of Contents**

[Introduction 3](#_Toc5129842)

[Sentiment Analysis and Data Visualization 3](#_Toc5129843)

[Sentiment Analysis Approaches and Its Benefits 4](#_Toc5129844)

[Overview of the Implementation 5](#_Toc5129845)

[Collection of Data and Prerequisites 5](#_Toc5129846)

[Cleaning Data and Pre-processing 5](#_Toc5129847)

[Visualization (Word cloud) 6](#_Toc5129848)

[Visualization (Charts) 6](#_Toc5129849)

[Main Packages used 8](#_Toc5129850)

[Limitations and Future Work 8](#_Toc5129851)

[Conclusion 9](#_Toc5129852)

[References 10](#_Toc5129853)

[Appendix: R Source Code 11](#_Toc5129854)

**Table of Figures**

[Figure 1: Visualization of Word Cloud 6](#_Toc5129780)

[Figure 2: Sentiment Value Chart over Wonder Woman movie tweets 6](#_Toc5129781)

[Figure 3: Sentiment Value Chart over Aquaman movie tweets 7](#_Toc5129782)

[Figure 4: Sentiment Value Chart over Captain Marvel movie tweets 7](#_Toc5129783)

[Figure 5: Sentiment Value Chart over Black Panther movie tweets 8](#_Toc5129784)

# Introduction

Twitter is an online news and social networking service that enables users to send and read short 140-character messages called "tweets". Registered users can read and post tweets, but those who are unregistered can only read them. Hence, Twitter is a public platform with a mine of public opinion of people all over the world and of all age categories. As of October 2016, Twitter has more than 315 million monthly active users. Twitter Sentiment Analysis is the process of determining the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions and emotions expressed within an online mention [1]. Tweets are imported using R and the data is cleaned by removing irrelevant elements such as punctuation marks, blank spaces, hash-tags, usernames, and URLs. Lexical Analysis is used to predict the sentiment of tweets and subsequently the opinions are plotted through ggplot and wordcloud function.

# Sentiment Analysis and Data Visualization

Sentiment Analysis examines the problem of studying texts, like posts and reviews, uploaded by users on microblogging platforms, forums, and electronic businesses, regarding the opinions they have about a product, service, event, person or idea. The most common use of Sentiment Analysis is to classify a text to a class. Depending on the dataset and the reason, Sentiment Classification can be binary (positive or negative) or multi-class (3 or more classes) problem [2]. In addition, among researchers and stakeholders, either similar or completely different opinions can be found concerning the relation between emotion detection and sentiment analysis, depending on their perspective. However, regardless the result or approach, they all adopt the same techniques.

# Sentiment Analysis Approaches and Its Benefits

The applications for sentiment analysis are endless. ​It is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics However, it is also practical for use in business analytics and situations in which text needs to be analysed [1]. Sentiment analysis is in demand because of its efficiency. Thousands of text documents can be processed for sentiment in seconds, compared to the hours it would take a team of people to manually complete. Because it is so efficient (and accurate – Semantic has 80% accuracy for English content) many businesses are adopting text and sentiment analysis and incorporating it into their processes. Applications: The applications of sentiment analysis are broad and powerful [3]. Shifts in sentiment on social media have been shown to correlate with shifts in the stock market.

For example, the Obama administration used sentiment analysis to gauge public opinion to policy announcements and campaign messages ahead of 2012 presidential election [2]. The ability to quickly understand consumer attitudes and react accordingly is something that Expedia Canada took advantage of when they noticed that there was a steady increase in negative feedback to the music used in one of their television adverts.

Sentiment Analysis can be executed using two major approaches. Such as; *Machine Learning:* This approach uses a machine-learning technique and diverse features to construct a classifier that can identify text that expresses sentiment. Nowadays, deep-learning methods are popular because they fit on data learning representations.

*Lexicon-Based:* This method uses a variety of words annotated by polarity score, to decide the general assessment score of a given content [4]. The strongest asset of this technique is that it does not require any training data, while its weakest point is that a large number of words and expressions are not included in sentiment lexicons.

In this study, I have chosen the lexicon-based analytics to obtain data visualization outcomes to represent the sentiments of chosen movie tweets.

# Overview of the Implementation

## Collection of Data and Prerequisites

Twitter Authentication to access API is obtained through developer twitter account. In order to execute the provided source code in appendix, developer account and sentiment analysis application is required. Developer account provides access over four credentials such as Consumer\_key, Consumer\_secret, Access\_token, and Access\_secret. Tweets are extracted using twitter application created in twitter interface. In R studio, “twitteR” library provides an interface to the Twitter web API and “ROAuth” library provides R Interface for authorization with using Consumer\_key, Consumer\_secret, Access\_token, and Access\_secret.

## Cleaning Data and Pre-processing

In order to perform data pre-processing, the following aspects are considered to reduce the discrepancies from the sentiment analysis work.

1. Extra punctuation marks
2. Redundant Blank spaces
3. Usernames
4. Hash-tags
5. Stop words
6. URLS
7. Algorithm used

*Lexical Analysis:*​ I have used uni-gram comparison to the pre-loaded text database, the tweets are assigned over sentiment score – positive or negative and overall score is calculated. I have presented the scores and the tweets with word cloud and column chart for better visualization.

## Visualization (Word cloud)

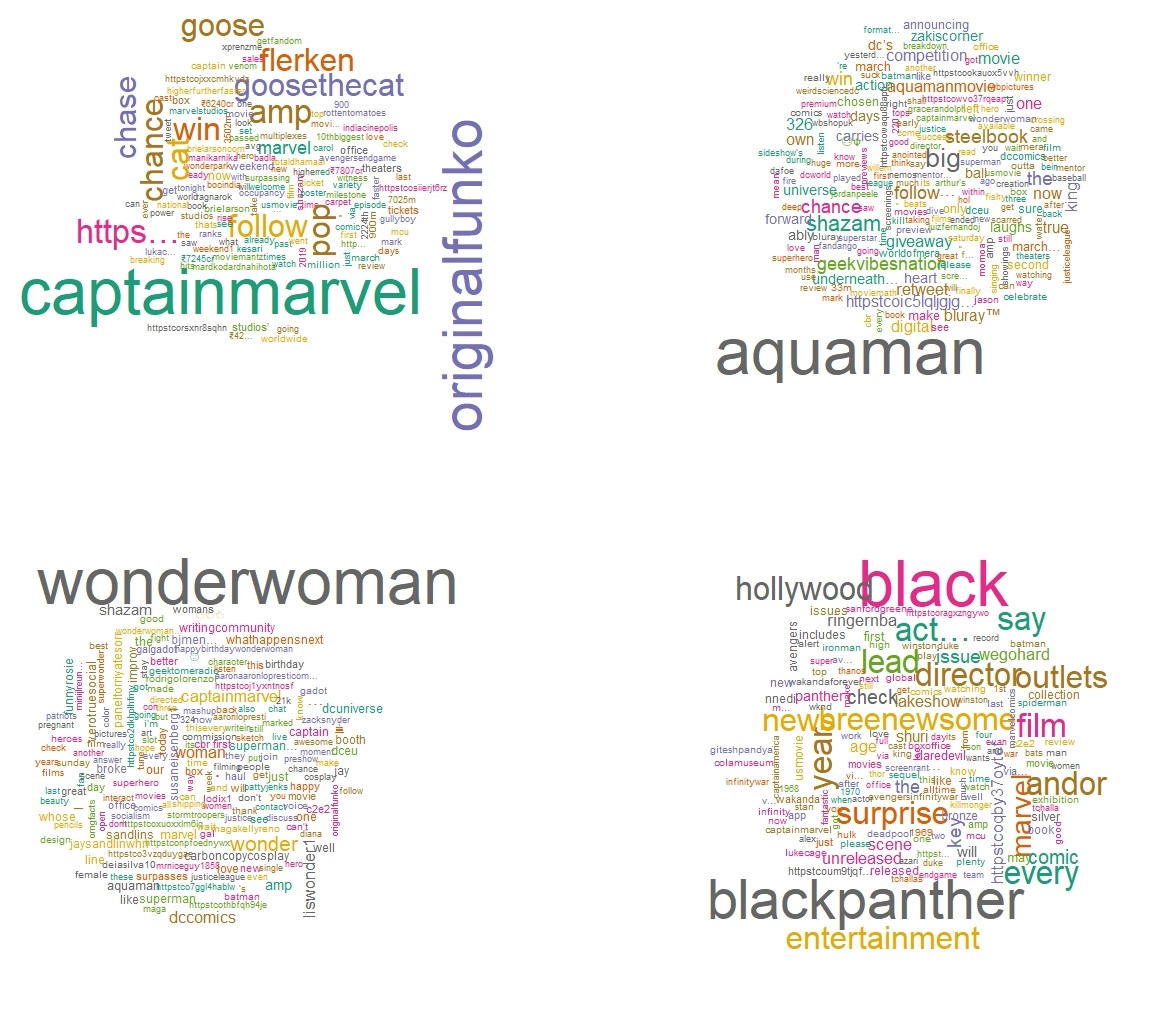


Figure 1: Visualization of Word Cloud

(Source: Created by author)

## Visualization (Charts)

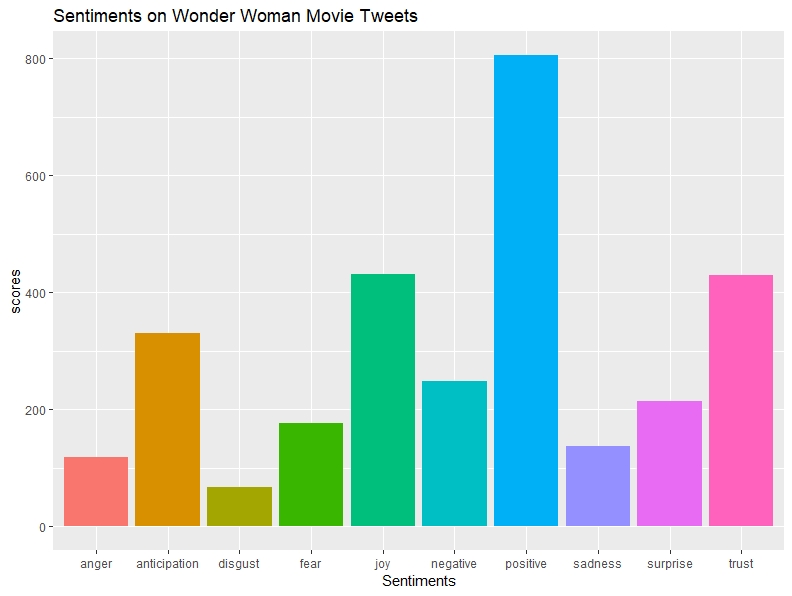


Figure 2: Sentiment Value Chart over Wonder Woman movie tweets

(Source: Created by author)

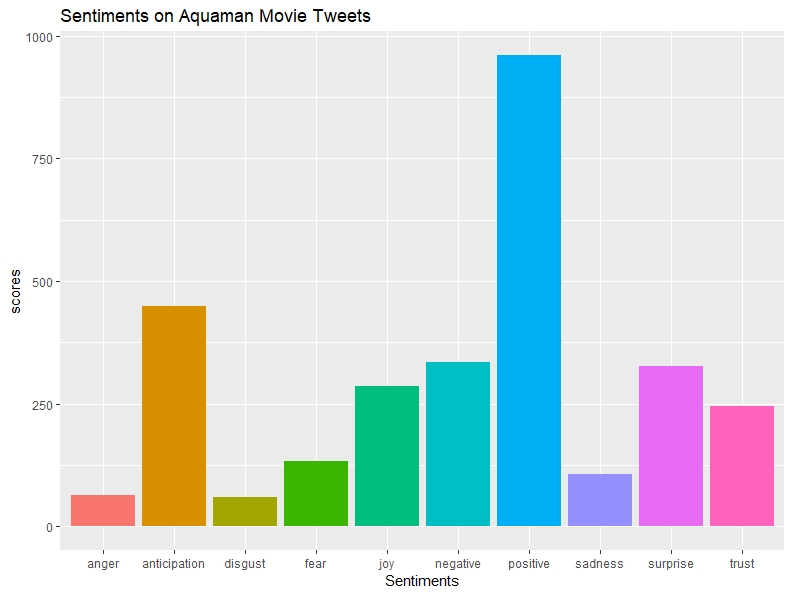


Figure 3: Sentiment Value Chart over Aquaman movie tweets

(Source: Created by author)

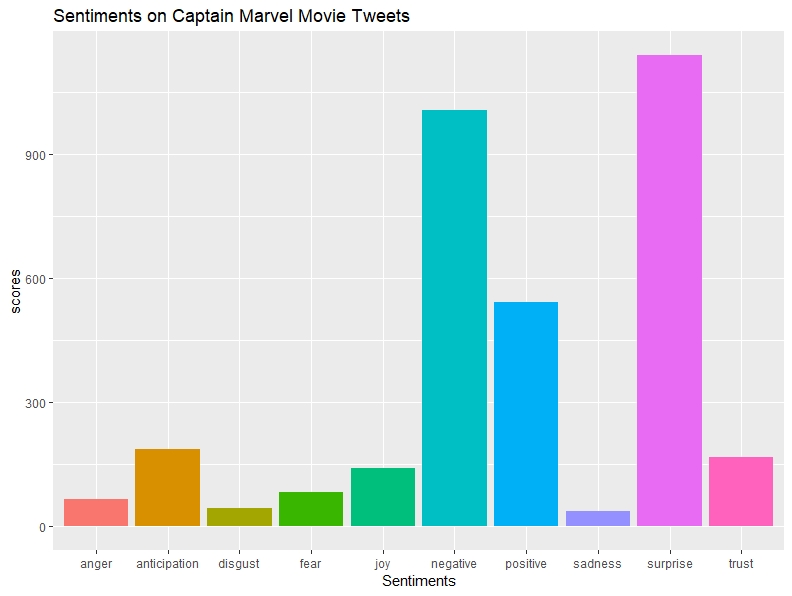


Figure 4: Sentiment Value Chart over Captain Marvel movie tweets

(Source: Created by author)

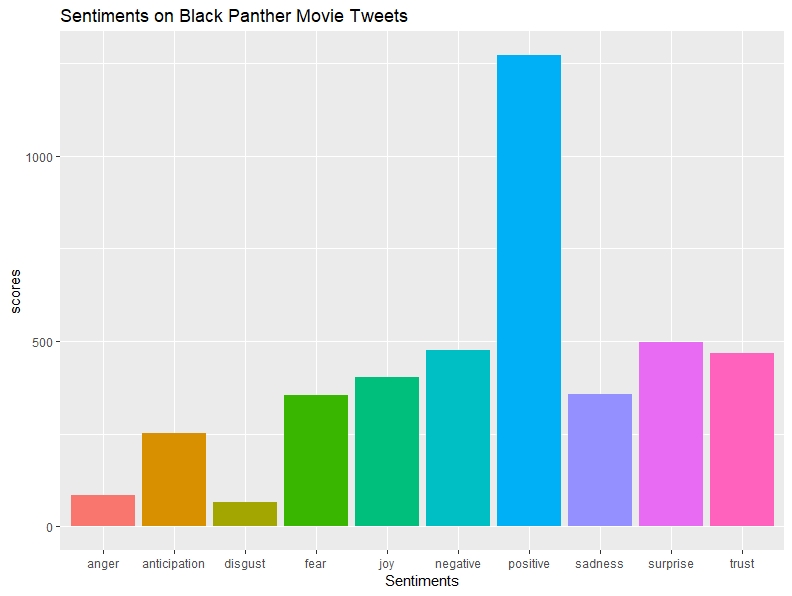


Figure 5: Sentiment Value Chart over Black Panther movie tweets

(Source: Created by author)

## Main Packages used

*twitteR​:* Provides an interface to the Twitter web API

*stringr​:* String operations in R

*ROAuth​:* Provides an interface to the OAuth 1.0 specification allowing users to authenticate via OAuth to the server of their choice.

*ggplot2​:* An implementation of the grammar of graphics in R. It combines the advantages of both base and lattice graphics: conditioning and shared axes are handled automatically, and you can still build up a plot step by step from multiple data sources.

*tm:* A framework for text mining applications within R.

## Limitations and Future Work

Some limitations exist in this work; mentioned as following:

1. The Twitter Search API can get tweets up to a maximum of 7 days old.
2. It cannot get 100% efficiency in analysing sentiment of tweets.
3. Detect sarcasm in tweets
4. Analyse images for emotions
5. Find number of mentions of a particular organizations

# Conclusion

Data visualization is the presentation of data in a pictorial or graphical format. It enables decision makers to see analytics presented visually, so they can grasp difficult concepts or identify new patterns. With interactive visualization, you can take the concept a step further by using technology to drill down into charts and graphs for more detail, interactively changing what data you see and how it’s processed. Tables, bar plots, histograms and pie charts can be used for visualization. I have presented the implementation over visualization so that collected primary data could be viewed along with its values.

# References

1. Anandarajan, M., Hill, C. and Nolan, T., 2019. Sentiment Analysis of Movie Reviews Using R. In Practical Text Analytics(pp. 193-220). Springer, Cham.
2. Jagdale, R.S., Shirsat, V.S. and Deshmukh, S.N., 2019. Sentiment analysis on product reviews using machine learning techniques. In Cognitive Informatics and Soft Computing (pp. 639-647). Springer, Singapore.
3. Redhu, S., Srivastava, S., Bansal, B. and Gupta, G., 2018. Sentiment Analysis Using Text Mining: A Review. International Journal on Data Science and Technology, 4(2), p.49.
4. Satija, T. and Thareja, R., 2018. Sentiment Analysis Using Twitter Mining in R Language.

# Appendix: R Source Code

library("NLP")

library("twitteR")

library("syuzhet")

library("tm")

library("SnowballC")

library("stringi")

library("topicmodels")

library("syuzhet")

library("twitteR")

library("ROAuth")

consumer\_key <- "Twitter Consumer Key"

consumer\_secret <- "Twitter Consumer Secret"

access\_token <- "Twitter Access Token"

access\_secret <- "Twitter Access Secret"

setup\_twitter\_oauth(consumer\_key, consumer\_secret, access\_token, access\_secret)

tweets\_cm <- searchTwitter("#captainmarvel", n=1000,lang = "en")

tweets\_aq <- searchTwitter("#aquaman", n=1000,lang = "en")

tweets\_ww <- searchTwitter("#wonderwoman", n=1000, lang = "en")

tweets\_bp <- searchTwitter("#blackpanther", n=1000, lang = "en")

cm\_tweets <- twListToDF(tweets\_cm)

aq\_tweets <- twListToDF(tweets\_aq)

ww\_tweets <- twListToDF(tweets\_ww)

bp\_tweets <- twListToDF(tweets\_bp)

View(cm\_tweets)

View(aq\_tweets)

View(ww\_tweets)

View(bp\_tweets)

cm\_text<- cm\_tweets$text

aq\_text<- aq\_tweets$text

ww\_text<- ww\_tweets$text

bp\_text<- bp\_tweets$text

# Convert all text to lower case

cm\_text<- tolower(cm\_text)

aq\_text<- tolower(aq\_text)

ww\_text<- tolower(ww\_text)

bp\_text<- tolower(bp\_text)

# Replace blank space ("rt")

cm\_text <- gsub("rt", "", cm\_text)

aq\_text <- gsub("rt", "", aq\_text)

ww\_text <- gsub("rt", "", ww\_text)

bp\_text <- gsub("rt", "", bp\_text)

# Remove the URLs

cm\_text <- gsub("http.\*","", cm\_text)

cm\_text <- gsub("https.\*","", cm\_text)

aq\_text <- gsub("http.\*","", aq\_text)

aq\_text <- gsub("https.\*","", aq\_text)

ww\_text <- gsub("http.\*","", ww\_text)

ww\_text <- gsub("https.\*","", ww\_text)

bp\_text <- gsub("http.\*","", bp\_text)

bp\_text <- gsub("https.\*","", bp\_text)

# Remove the hash-tags

cm\_text <- gsub("#.\*","", cm\_text)

aq\_text <- gsub("#.\*","", aq\_text)

ww\_text <- gsub("#.\*","", ww\_text)

bp\_text <- gsub("#.\*","", bp\_text)

# Replace @UserName

cm\_text <- gsub("@\\w+", "", cm\_text)

aq\_text <- gsub("@\\w+", "", aq\_text)

ww\_text <- gsub("@\\w+", "", ww\_text)

bp\_text <- gsub("@\\w+", "", bp\_text)

# Remove punctuation

cm\_text <- gsub("[[:punct:]]", "", cm\_text)

aq\_text <- gsub("[[:punct:]]", "", aq\_text)

ww\_text <- gsub("[[:punct:]]", "", ww\_text)

bp\_text <- gsub("[[:punct:]]", "", bp\_text)

# Remove tabs

cm\_text <- gsub("[ |\t]{2,}", "", cm\_text)

aq\_text <- gsub("[ |\t]{2,}", "", aq\_text)

ww\_text <- gsub("[ |\t]{2,}", "", ww\_text)

bp\_text <- gsub("[ |\t]{2,}", "", bp\_text)

# Remove blank spaces at the beginning

cm\_text <- gsub("^ ", "", cm\_text)

aq\_text <- gsub("^ ", "", aq\_text)

ww\_text <- gsub("^ ", "", ww\_text)

bp\_text <- gsub("^ ", "", bp\_text)

# Remove blank spaces at the end

cm\_text <- gsub(" $", "", cm\_text)

aq\_text <- gsub(" $", "", aq\_text)

ww\_text <- gsub(" $", "", ww\_text)

bp\_text <- gsub(" $", "", bp\_text)

library("wordcloud")

#generate wordcloud

wordcloud(cm\_tweets$text,min.freq = 10,colors=brewer.pal(8, "Dark2"),random.color = TRUE,max.words = 500)

wordcloud(aq\_tweets$text,min.freq = 10,colors=brewer.pal(8, "Dark2"),random.color = TRUE,max.words = 500)

wordcloud(ww\_tweets$text,min.freq = 10,colors=brewer.pal(8, "Dark2"),random.color = TRUE,max.words = 500)

wordcloud(bp\_tweets$text,min.freq = 10,colors=brewer.pal(8, "Dark2"),random.color = TRUE,max.words = 500)

#getting emotions using in-built function

mysentiment\_cm<-get\_nrc\_sentiment((cm\_text))

mysentiment\_aq<-get\_nrc\_sentiment((aq\_text))

mysentiment\_ww<-get\_nrc\_sentiment((ww\_text))

mysentiment\_bp<-get\_nrc\_sentiment((bp\_text))

#calculationg total score for each sentiment

Sentimentscores\_cm<-data.frame(colSums(mysentiment\_cm[,]))

Sentimentscores\_aq<-data.frame(colSums(mysentiment\_aq[,]))

Sentimentscores\_ww<-data.frame(colSums(mysentiment\_ww[,]))

Sentimentscores\_bp<-data.frame(colSums(mysentiment\_bp[,]))

names(Sentimentscores\_cm)<-"Score"

Sentimentscores\_cm<-cbind("sentiment"=rownames(Sentimentscores\_cm),Sentimentscores\_cm)

rownames(Sentimentscores\_cm)<-NULL

names(Sentimentscores\_aq)<-"Score"

Sentimentscores\_aq<-cbind("sentiment"=rownames(Sentimentscores\_aq),Sentimentscores\_aq)

rownames(Sentimentscores\_aq)<-NULL

names(Sentimentscores\_ww)<-"Score"

Sentimentscores\_ww<-cbind("sentiment"=rownames(Sentimentscores\_ww),Sentimentscores\_ww)

rownames(Sentimentscores\_ww)<-NULL

names(Sentimentscores\_bp)<-"Score"

Sentimentscores\_bp<-cbind("sentiment"=rownames(Sentimentscores\_bp),Sentimentscores\_bp)

rownames(Sentimentscores\_bp)<-NULL

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#plotting the sentiments with scores

library("ggplot2")

ggplot(data=Sentimentscores\_cm,aes(x=sentiment,y=Score))+geom\_bar(aes(fill=sentiment),stat = "identity")+

theme(legend.position="none")+

xlab("Sentiments")+ylab("scores")+ggtitle("Sentiments on Captain Marvel Movie Tweets")

ggplot(data=Sentimentscores\_aq,aes(x=sentiment,y=Score))+geom\_bar(aes(fill=sentiment),stat = "identity")+

theme(legend.position="none")+

xlab("Sentiments")+ylab("scores")+ggtitle("Sentiments on Aquaman Movie Tweets")

ggplot(data=Sentimentscores\_ww,aes(x=sentiment,y=Score))+geom\_bar(aes(fill=sentiment),stat = "identity")+

theme(legend.position="none")+

xlab("Sentiments")+ylab("scores")+ggtitle("Sentiments on Wonder Woman Movie Tweets")

ggplot(data=Sentimentscores\_bp,aes(x=sentiment,y=Score))+geom\_bar(aes(fill=sentiment),stat = "identity")+

theme(legend.position="none")+

xlab("Sentiments")+ylab("scores")+ggtitle("Sentiments on Black Panther Movie Tweets")