covid_eda

Irfan Ainuddin, Ashley Person, Chicago

4/17/2020

```
## import data and clean
## deal with scientific notation
options(scipen = 999)

df <- readRDS(file = "covid-tweets-2020-04-17.rds")

#who <- readRDS(file = "who_tweets.Rds")
#cdc <- readRDS(file = "cdc_tweets.Rds")
```

```
remove_reg <- "&amp;|&lt;|&gt;"
```

```
## filter out retweets and tweets from @CDCgov and @WHO
all <- df %>% filter(!screen_name == "CDCgov", !screen_name == "WHO", is_retweet == FALSE)
```

```
# get tweets from @CDCgov and filter out retweets
cdc <- df %>% filter(screen_name == "CDCgov", is_retweet == FALSE)
# get tweets from @CDCgov and filter out retweets
who <- df %>% filter(screen_name == "WHO", is_retweet == FALSE)


#fema_df <- df %>% filter(screen_name == "fema", is_retweet == FALSE)
#NHSuk_df <- df %>% filter(screen_name == "HHSGov", is_retweet == FALSE)
```

```
## create working tweet tibble
# tweets <- tibble(tweet_id=all$tweet_id, text=all$text)

# what is the point if tibblation?
tweets <- as_tibble(all)
```

```
## Preparing public twitter text for sentiment analysis
tweet_words <- tweets %>%
                  ## remove @mentions to users
                  filter(!str_detect(text,"^(@)")) %>%
                  ## remove the words from remove_reg variable above
                  mutate(text = str_remove_all(text, remove_reg)) %>%
                  ## tokenize the words for sentiment analysis
                  unnest_tokens(word, text, token="tweets", strip_url = TRUE) %>%
                  ## filter out stop words using all libraries(SMART, onix, snowball)
                  filter(!word %in% stop_words$word,
                         !word %in% str_remove_all(stop_words$word, "'"))
```

```
## Using `to_lower = TRUE` with `token = 'tweets'` may not preserve URLs.
```

```
## tweets broken into each word.
#tweet_words
```

```
## apply NRC sentiments to all tweets
tweet_words_nrc <- tweet_words %>%
  inner_join(get_sentiments("nrc"))
```

```
## Joining, by = "word"
```

```
cdc_tweets <- as_tibble(cdc)

## Preparing public twitter text for sentiment analysis
cdc_tweet_words <- cdc_tweets %>%
                  ## remove the words from remove_reg variable above
                  mutate(text = str_remove_all(text, remove_reg)) %>%
                  ## tokenize the words for sentiment analysis
                  unnest_tokens(word, text, token="tweets", strip_url = TRUE) %>%
                  ## filter out stop words using all libraries(SMART, onix, snowball)
                  filter(!word %in% stop_words$word,
                         !word %in% str_remove_all(stop_words$word, "'"))
```

```
## Using `to_lower = TRUE` with `token = 'tweets'` may not preserve URLs.
```

```
## tweets broken into each word.
#cdc_tweet_words
```

```
## apply NRC sentiments
cdc_tweet_words_nrc <- cdc_tweet_words %>%
          inner_join(get_sentiments("nrc"))
```

```
## Joining, by = "word"
```

```
who_tweets <- as_tibble(who)

## Preparing public twitter text for sentiment analysis
who_tweet_words <- who_tweets %>%
                  ## remove the words from remove_reg variable above
                  mutate(text = str_remove_all(text, remove_reg)) %>%
                  ## tokenize the words for sentiment analysis
                  unnest_tokens(word, text, token="tweets", strip_url = TRUE) %>%
                  ## filter out stop words using all libraries(SMART, onix, snowball)
                  filter(!word %in% stop_words$word,
                         !word %in% str_remove_all(stop_words$word, "'"))
```

```
## Using `to_lower = TRUE` with `token = 'tweets'` may not preserve URLs.
```

```
## tweets broken into each word.
#who_tweet_words
```

```
## apply NRC sentiments
who_tweet_words_nrc <- who_tweet_words %>%
          inner_join(get_sentiments("nrc"))
```

```
## Joining, by = "word"
```

++ BEGIN EDA ++

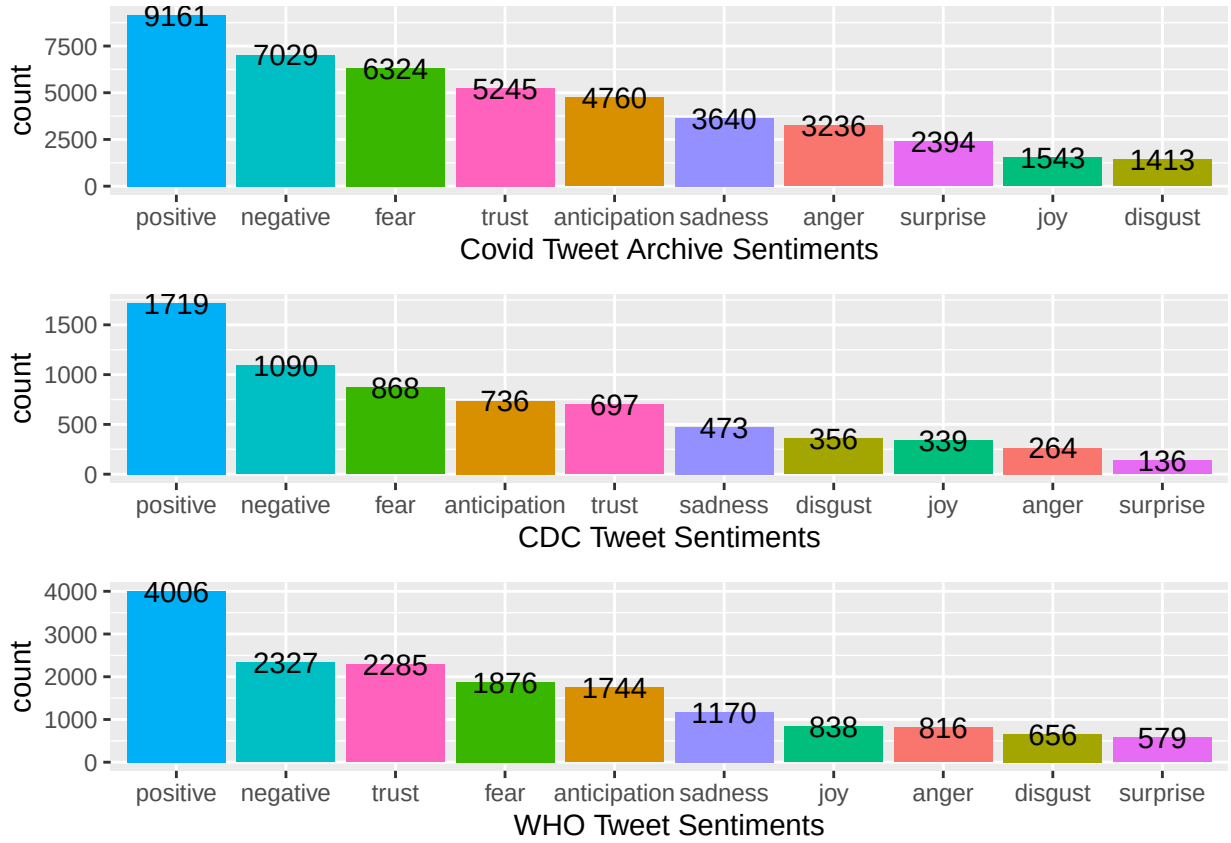
```
## Add counts for each sentiment w/ pivot_wider()
tweet_count <- tweet_words_nrc %>% group_by(tweet_id, sentiment) %>% count() %>% pivot_wider(names_from

## Add counts foreach sentiment w/ pivot_wider()
cdc_tweet_count <- cdc_tweet_words_nrc %>% group_by(tweet_id, sentiment) %>% count() %>% pivot_wider(nar

## Add counts for each sentiment w/ pivot_wider()
who_tweet_count <- who_tweet_words_nrc %>% group_by(tweet_id, sentiment) %>% count() %>% pivot_wider(nar
```

```
## Sentiment Counts
all_sentiment_count <- colSums(tweet_count[,-1], na.rm = TRUE)

## CDC sentiment count
cdc_sentiment_count <- colSums(cdc_tweet_count[,-1], na.rm = TRUE)

# WHO sentiment count
who_sentiment_count <- colSums(who_tweet_count[,-1], na.rm = TRUE)

## convert from named list to data frame for all
all_sentiment_sum = data.frame(count=all_sentiment_count, sentiment=names(all_sentiment_count))
## convert from named list to data frame for CDC
cdc_sentiment_sum = data.frame(count=cdc_sentiment_count, sentiment=names(cdc_sentiment_count))
## convert from named list to data frame for WHO
who_sentiment_sum = data.frame(count=who_sentiment_count, sentiment=names(who_sentiment_count))
```

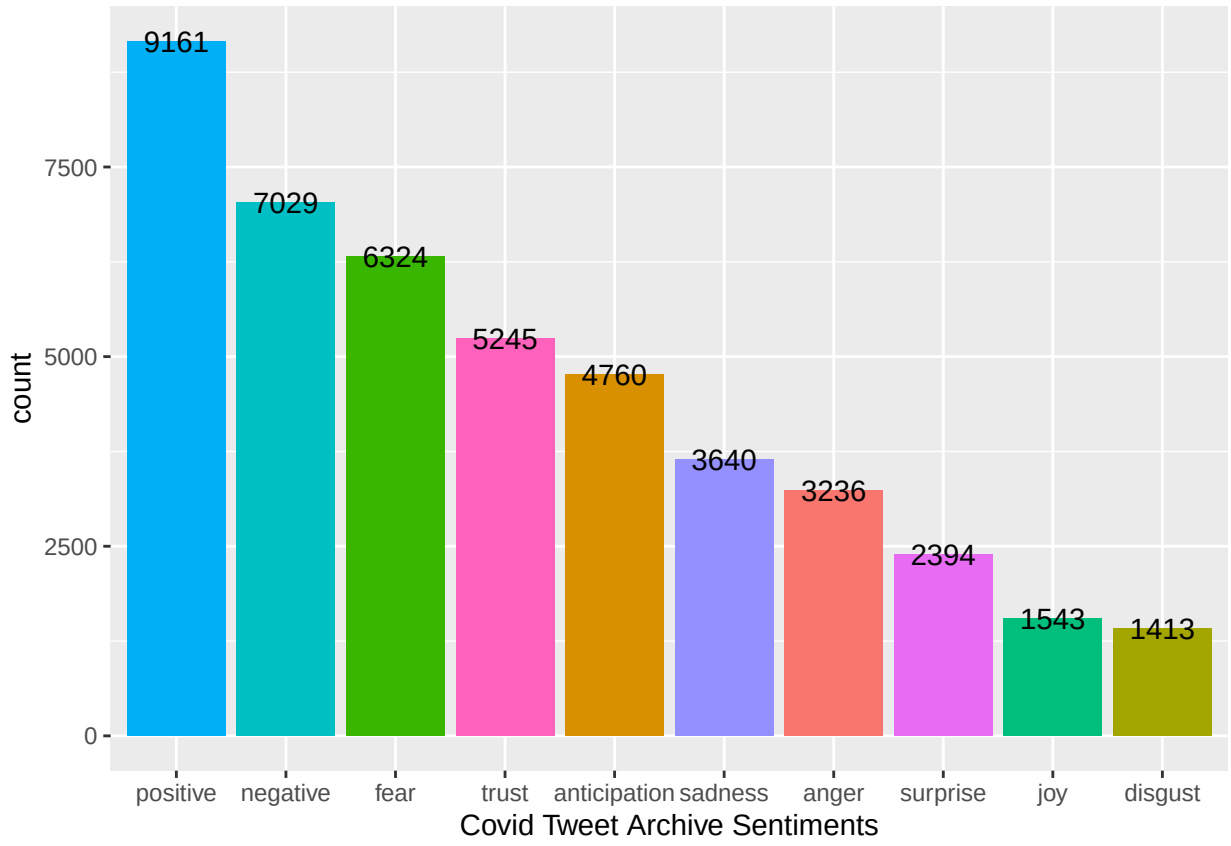
```
## set factor levels for all
#all_sentiment_sum$sentiment = factor(all_sentiment_sum$sentiment, levels=all_sentiment_sum$sentiment[o
## set factor levels for CDC
#cdc_sentiment_sum$sentiment = factor(cdc_sentiment_sum$sentiment, levels=cdc_sentiment_sum$sentiment[o
## set factor levels for WHO
#who_sentiment_sum$sentiment = factor(who_sentiment_sum$sentiment, levels=who_sentiment_sum$sentiment[o
```

```
## Plot total number of sentiments for ALL tweets
gg_count_all <- all_sentiment_sum %>% ggplot(aes(reorder(sentiment, -count), count)) +
  geom_bar(stat="identity", aes(fill=sentiment)) +
  geom_text(aes(label=count)) +
  theme(legend.position = "none") +
  xlab('Covid Tweet Archive Sentiments')

## Plot total number of sentiments for CDC tweets
gg_count_cdc <- cdc_sentiment_sum %>% ggplot(aes(reorder(sentiment, -count), count)) +
  geom_bar(stat="identity", aes(fill=sentiment)) +
  geom_text(aes(label=count)) +
  theme(legend.position = "none") +
  xlab('CDC Tweet Sentiments')

## Plot total number of sentiments for WHO tweets
gg_count_who <- who_sentiment_sum %>% ggplot(aes(reorder(sentiment, -count), count)) +
  geom_bar(stat="identity", aes(fill=sentiment)) +
  geom_text(aes(label=count)) +
```

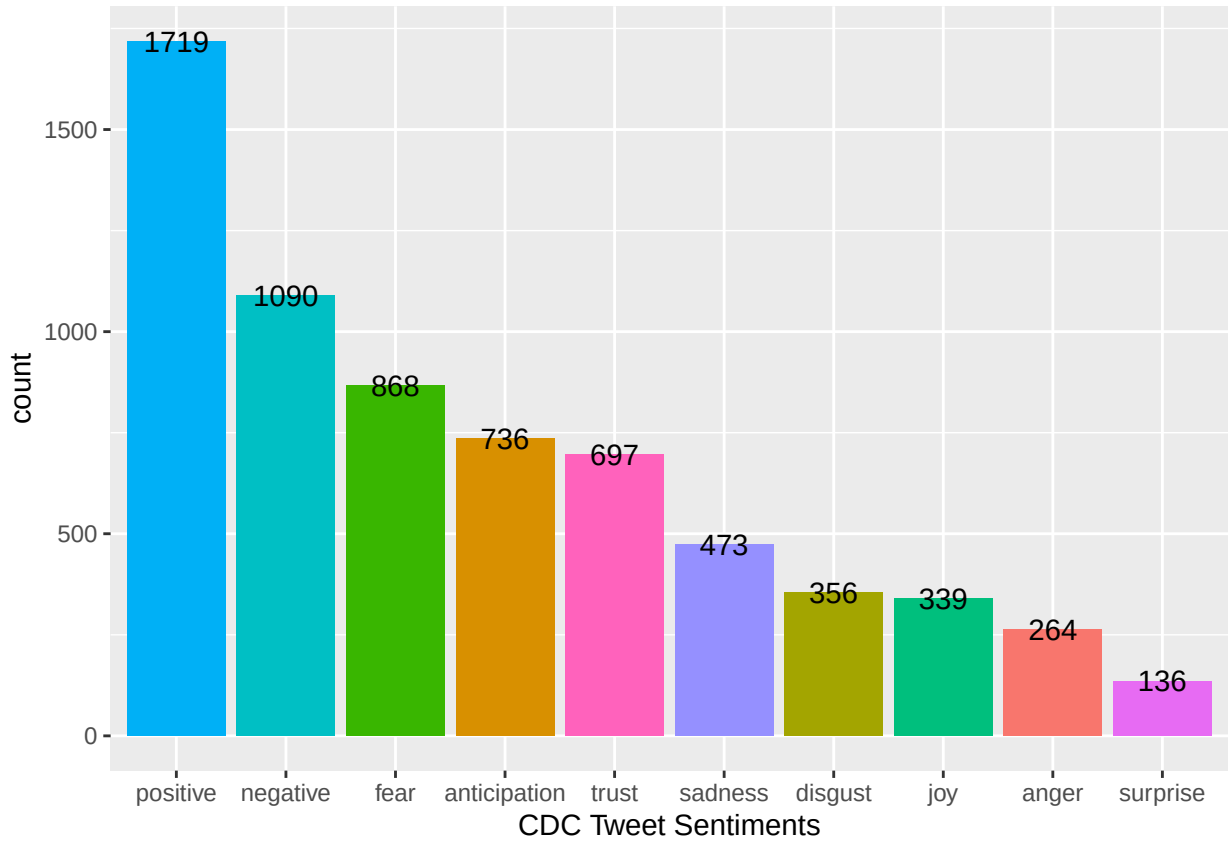
```
  theme(legend.position = "none") +
  xlab('WHO Tweet Sentiments')

grid.arrange(gg_count_all,gg_count_cdc,gg_count_who, ncol=1)
```



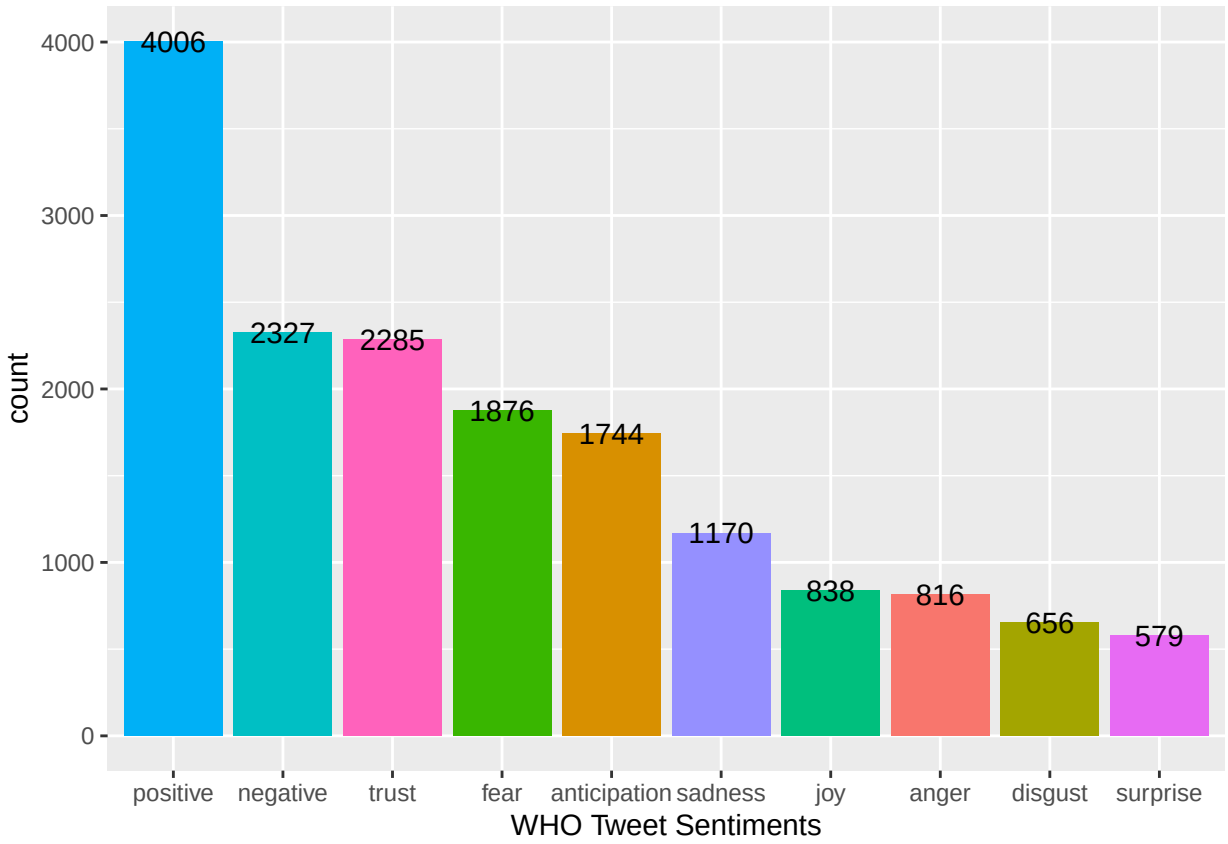

```
gg_count_all
```


9161
7029
6324
5245
4760
3640
3236
2394
1543
1413
7500
5000
2500
0
count
positive
negative
fear
trust
anticipation
sadness
anger
surprise
joy
disgust
Covid Tweet Archive Sentiments

```
gg_count_cdc
```


1719
1090
868
736
697
473
356
339
264
136
1500
1000
500
0
count
positive
negative
fear
anticipation
trust
sadness
disgust
joy
anger
surprise
CDC Tweet Sentiments

```
gg_count_who
```

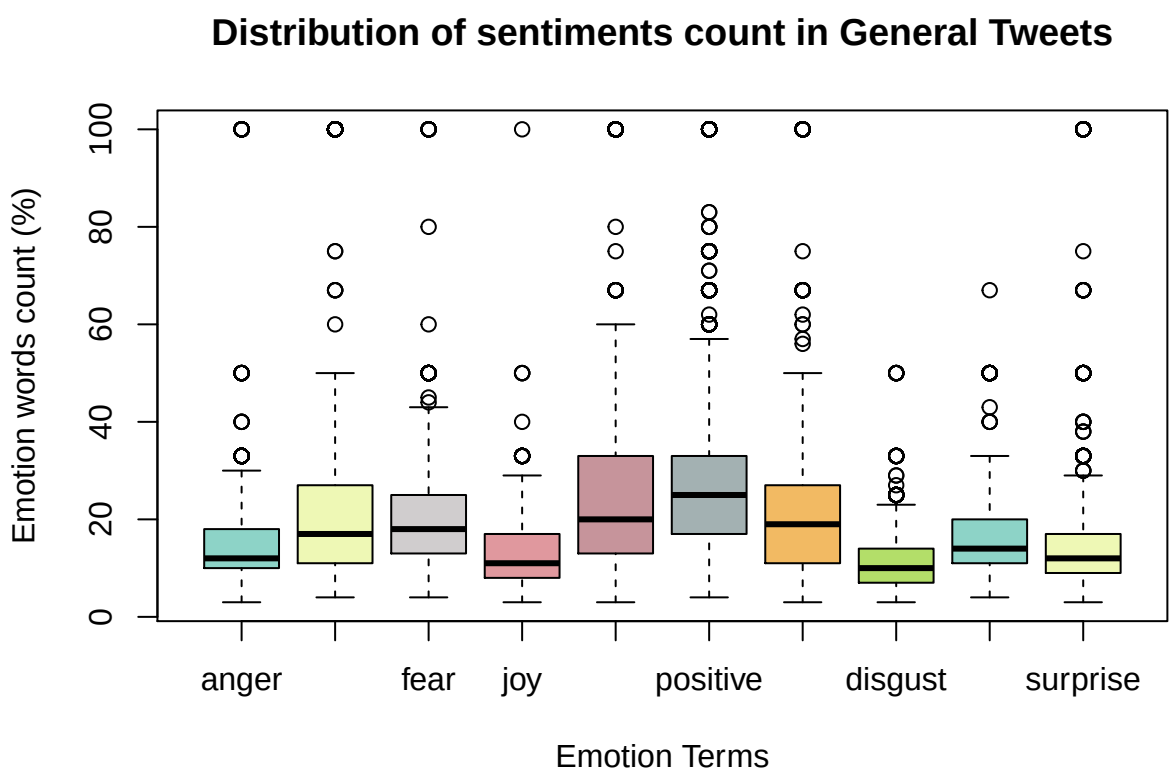


++ ATTEMPT TO APPLY PROPORTIONS ++++++++++++++++++++++++++++++++

```
## calculate total word count
nrc_word_total <- tweet_words %>%
  summarize(total= n())

## calculate NRC word sentiment total for ALL
nrc_sent_count <- tweet_words_nrc %>%
  group_by(tweet_id, sentiment) %>%
  summarize( freq = n()) %>%
  mutate(percent=round(freq/sum(freq)*100)) %>%
  ungroup()

cdc_sent_count <- cdc_tweet_words_nrc %>%
  group_by(tweet_id, sentiment) %>%
  summarize( freq = n()) %>%
  mutate(percent=round(freq/sum(freq)*100)) %>%
  ungroup()

who_sent_count <- who_tweet_words_nrc %>%
  group_by(tweet_id, sentiment) %>%
  summarize( freq = n()) %>%
  mutate(percent=round(freq/sum(freq)*100)) %>%
```

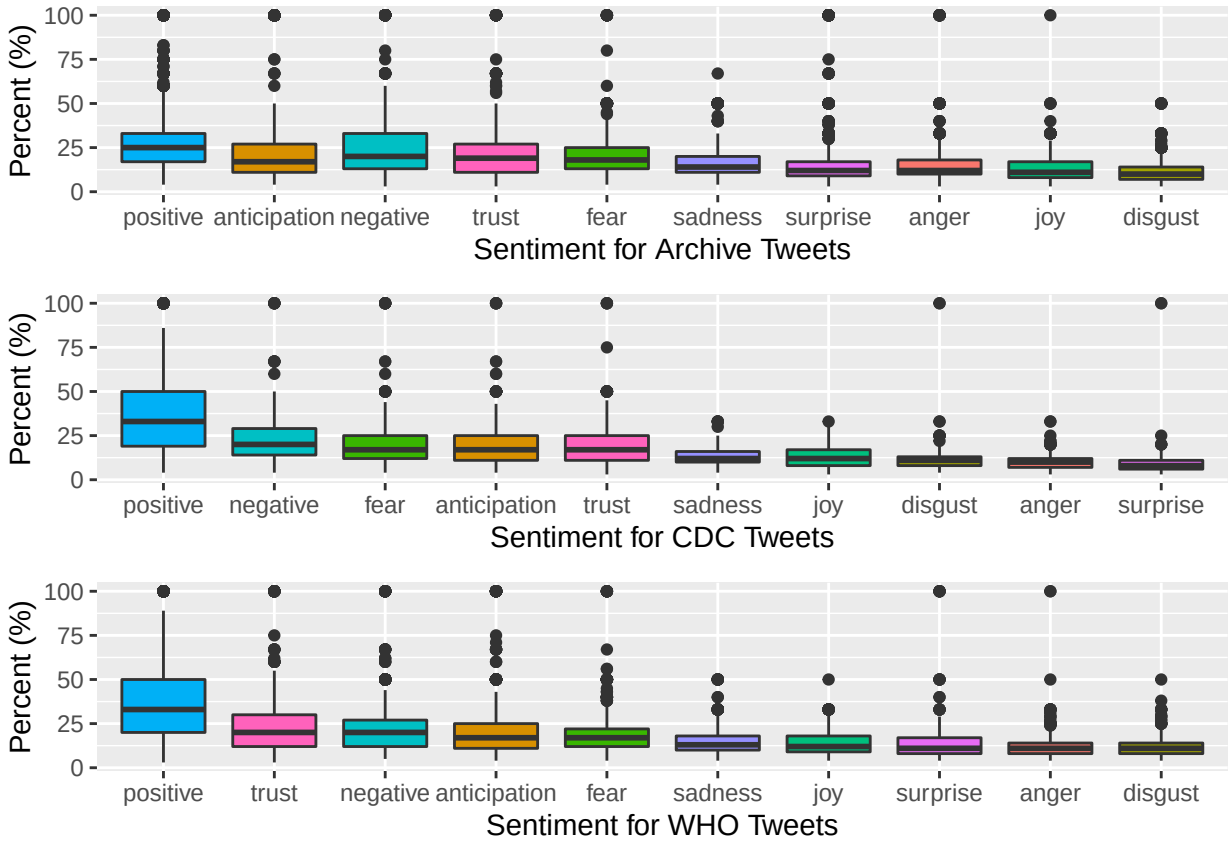
```
  ungroup()
```

```
## put into wide data
boxplotdata = nrc_sent_count %>%
  select(-freq) %>%
  pivot_wider(names_from = sentiment, values_from = percent, ) %>%
  ungroup()


cols  <- colorRampPalette(brewer.pal(7, "Set3"), alpha=TRUE)(8)

boxplot(boxplotdata[,c(2:11)], col=cols, textcolor="red",xlab="Emotion Terms", ylab="Emotion words coun
```

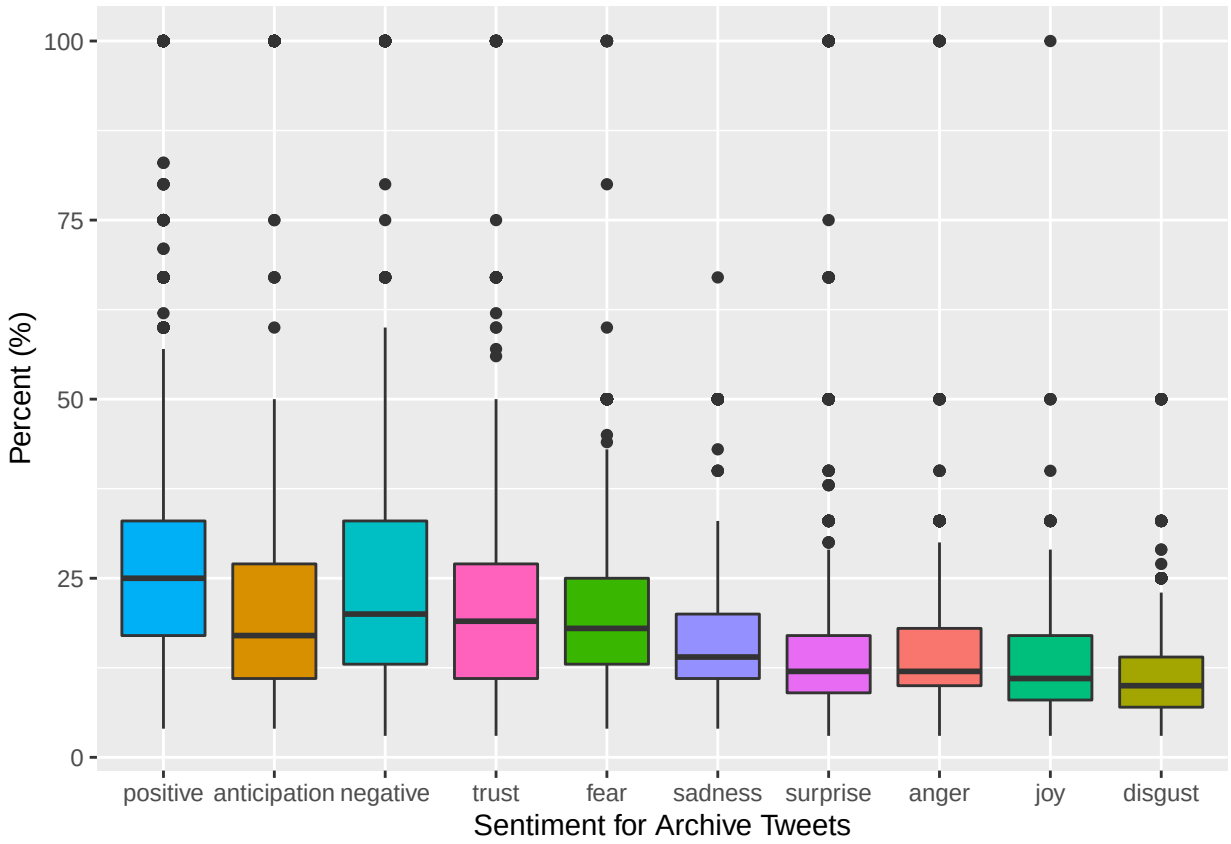



```
gg_prop_all <- nrc_sent_count %>%
  ggplot(aes(reorder(sentiment, -percent),percent)) +
  geom_boxplot(aes(fill=sentiment)) +
  theme(legend.position = "none") +
  xlab("Sentiment for Archive Tweets") +
  ylab("Percent (%)")


gg_prop_cdc <- cdc_sent_count %>%
  ggplot(aes(reorder(sentiment, -percent),percent)) +
  geom_boxplot(aes(fill=sentiment)) +
  theme(legend.position = "none") +
  xlab("Sentiment for CDC Tweets") +
  ylab("Percent (%)")
```

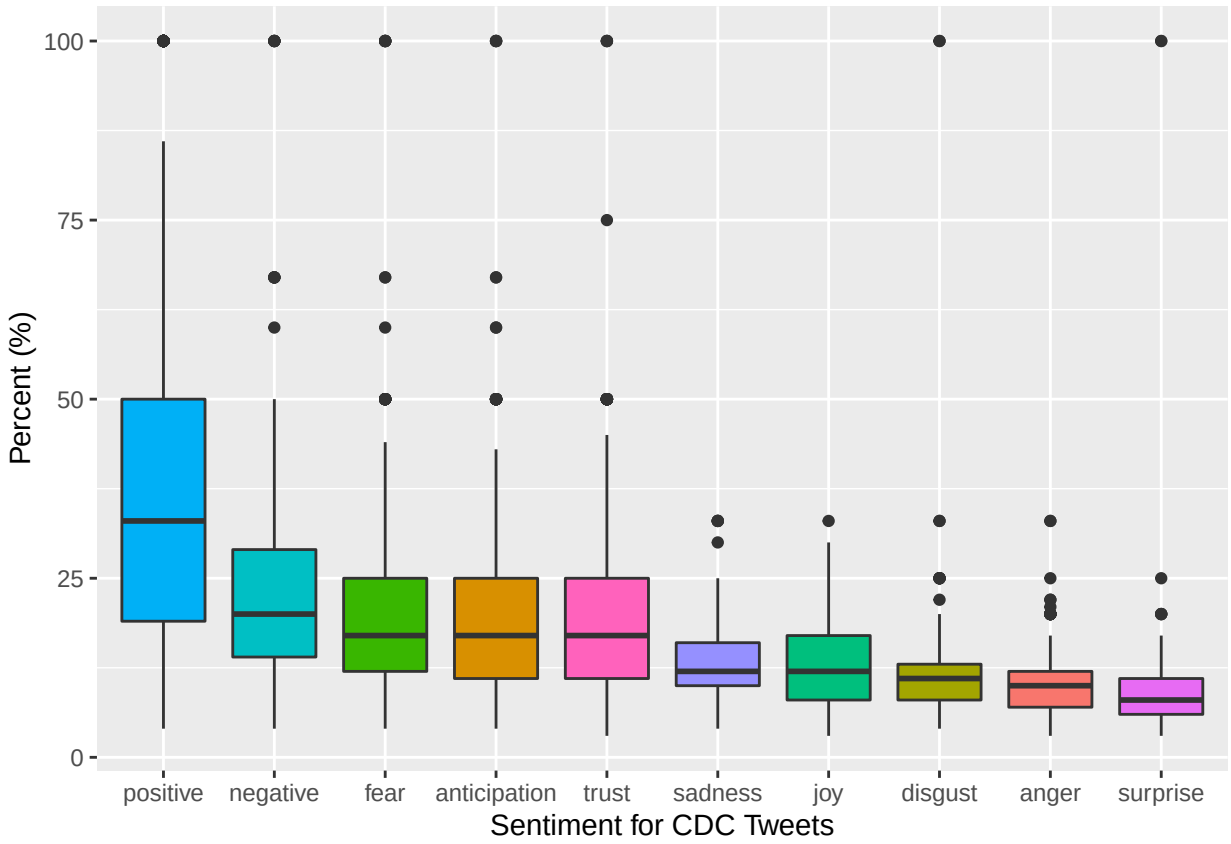
```
gg_prop_who <- who_sent_count %>%
  ggplot(aes(reorder(sentiment, -percent),percent)) +
  geom_boxplot(aes(fill=sentiment)) +
  theme(legend.position = "none") +
  xlab("Sentiment for WHO Tweets") +
  ylab("Percent (%)")

grid.arrange(gg_prop_all,gg_prop_cdc,gg_prop_who, ncol=1)
```



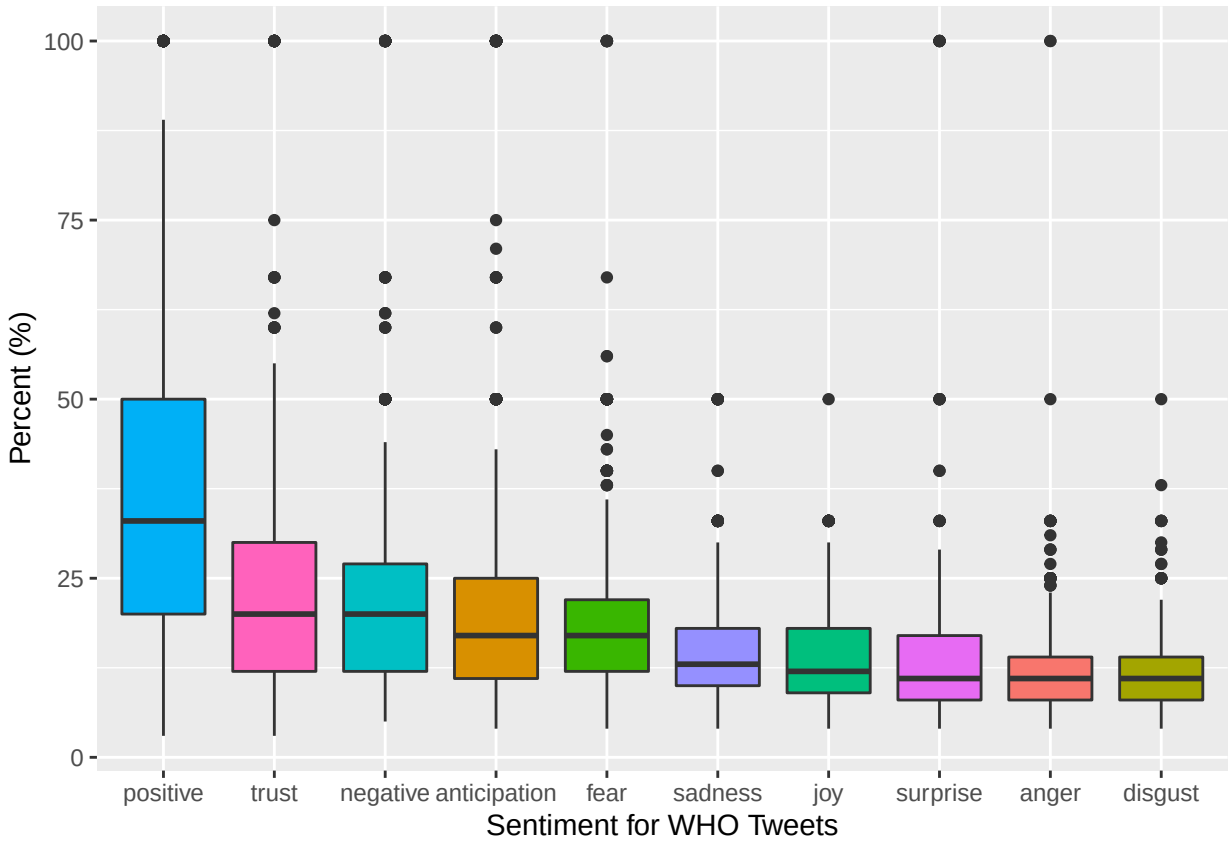

```
gg_prop_all
```


100
75
50
25
0
Percent (%)
positive
anticipation
negative
trust
fear
sadness
surprise
anger
joy
disgust
Sentiment for Archive Tweets

```
gg_prop_cdc
```


100
75
50
25
0
Percent (%)
positive
negative
fear
anticipation
trust
sadness
joy
disgust
anger
surprise
Sentiment for CDC Tweets

```
gg_prop_who
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.