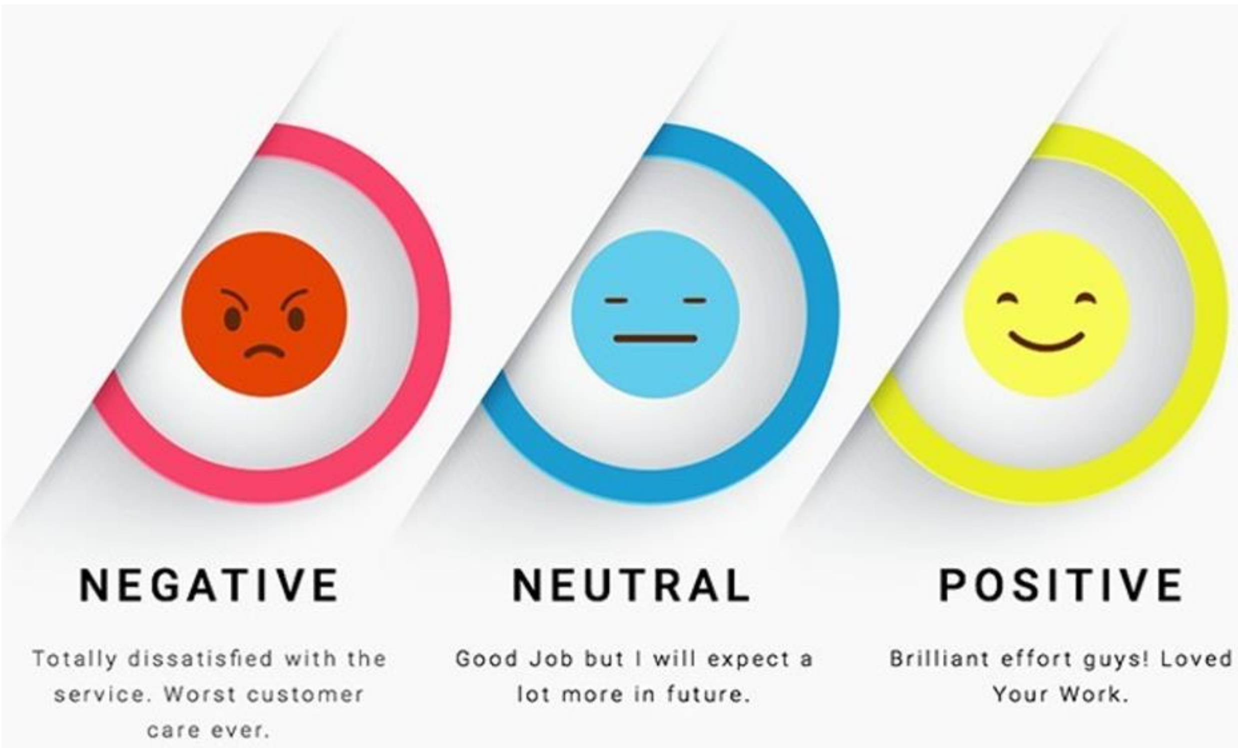
SENTIMENT ANALYSIS OF SURVEY COMMENTS

Image Source: https://houseofbots.com/news-detail/

Data Source: https://surveymonkey.com



EXECUTIVE SUMMARY

I am using survey data collected from over 200+ respondents for the UC Analytics Summit event surveys I had created in SurveyMonkey. I analyzed 4 training sessions by 4 different instructors spread over a span of two days. The survey had questions like

- Was the material presented in the right amount of detail and in the right depth?
- Will the content help you complete work tasks more effectively?
- Did the course instructor communicate clearly and kept your interest throughout the course?
- Was the course instructor knowledgeable in the subject matter?

All these questions had answer levels of

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly disagree

Although it was easy to analyze such questions, we cannot manually analyze accurately the comments we got. Hence, we chose to do a sentiment analysis.

PURPOSE

The problem at hand is to analyze the comments using text mining technique by building a sentiment analysis model in R.

We had received positive, negative and neutral comments. Below is a sample of comments.

- I would like to see more discussion on case studies, recommended practices, etc. regarding how best to structure the data analytics teams, how to set-up metrics for the performance of the data analytics teams, etc.
- Excellent training!
- Maybe sending out a more detailed list of functions to cover so that we could have digested them ahead of time.
- The course material was perfect. Topic details, test cases then detailed descriptions of how to complete the case.
- The course was great! Thank you!

My goal is to determine which course was the most popular and got the best feedback at the summit.

ANALYSIS AND EVIDENCE

How sentiment analysis works:

1. Create or find a list of words associated with strongly positive or negative or other sentiments.
2. Count the number of words for each sentiment in the text.
3. Analyze the mix of positive to negative words. Many positive words and few negative words indicate positive sentiment, while many negative words and few positive words indicates negative sentiment.

I am using Lexicon based approach to solve this problem. This method uses a variety of words annotated by polarity score, to decide the general assessment score of a given content. 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.

I am using **sentimentr** and **syuzhet** libraries in R for sentiment analysis.

Sentiment analysis is not perfect, and as with any automatic analysis of language, we may have errors in the results. It also cannot tell you why a writer is feeling a certain way at the time of writing the comment. However, it can be useful to quickly summarize some qualities of text.

SENTIMENT IN EACH TRAINING SESSION

MS POWER BI TRAINING

Created a table showing the sentiment score for each sentence. Anything below a score of -0.05 we tag as negative and anything above 0.05 we tag as positive. Anything in between inclusively, we tag as neutral.

```
sent_msbi <- sentiment(text_msbi, by = NULL)
```

	element_id	sentence_id	word_count	sentiment
1	1	1	17	-0.0485071250072666
2	1	2	10	0.173925271309261
3	1	3	9	-0.0266666666666667
4	1	4	34	0.370737066431818
5	1	5	10	0
6	1	6	11	0
7	1	7	11	-0.376889180722205
8	1	8	18	0.183847763108502
9	1	9	11	-0.241209075662211
10	1	10	9	0.166666666666667
11	1	11	24	0.376558016916606
12	1	12	13	0
13	1	13	4	0.54
14	1	14	5	0
15	1	15	13	-0.124807544150677
16	1	16	5	0.64398757751994
17	1	17	1	0
18	1	18	5	0.335410196624968
19	1	19	13	0
20	1	20	7	0.188982236504614
21	1	21	26	0.313785816221094
22	1	22	14	1.02361055652459
23	1	23	21	-0.0545544725589981
24	1	24	NA	0
25	1	25	2	0.707106781186547
26	1	26	31	0.323289543648195
27	1	27	14	0.919445487489218

```
```{r}
mean(sent_msbi$sentiment)
```

[1] 0.1998044
```

Result: The mean sentiment for this training session is 0.19 So we can conclude that the overall sentiment is highly positive. To study the sentiments further, I visualized the data.

By using **extract_sentiment_terms** function, I segrated negative words and the positive ones in each sentence.

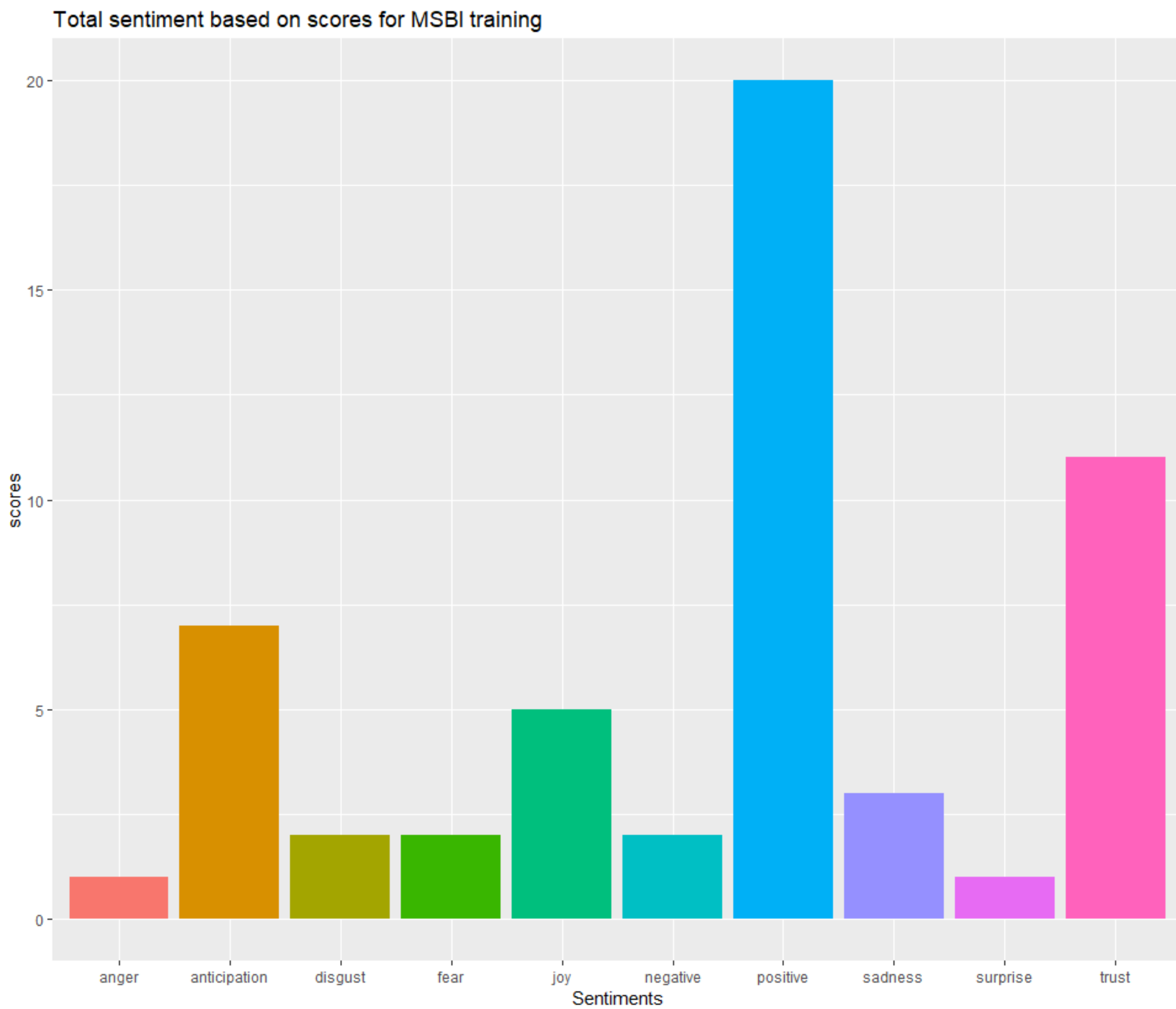
sent_terms_msbi <- extract_sentiment_terms(text_msbi, by = NULL)

	element_id	sentence_id	negative	neutral	positive	sentence
1	1	1	would be	c("it", "if", "we", "could", "with", "our", "own", "data", "se...	c("awesome", "work")	It would be awesome, if we could work with our own da...
2	1	2	slowed	c("was", "because", "of", "my", "unfamiliarity", "with", "t...	learning	Learning was slowed because of my unfamiliarity with ...
3	1	3	character(0)	c("in", "practicality", "not", "sure", "if", "this", "suggestio...	feasible	In practicality, not sure if this suggestion is feasible.
4	1	4	bad	c("could", "be", "a", "a", "idea", "but", "based", "on", "pa...	c("working", "partner", "helps")	Could be a bad idea but based on past trainings worki...
5	1	5	character(0)	c("the", "the", "end", "of", "first", "day", "felt", "a", "little"...	character(0)	The end of the first day felt a little rushed.
6	1	6	character(0)	c("also", "the", "last", "part", "security", "was", "a", "littl...	character(0)	Also, the last part (security) was a little over my head.
7	1	7	hard	c("it", "was", "to", "follow", "if", "i", "didn't", "database", "...	understand	It was hard to follow if I didn't understand database ter...
8	1	8	could have	c("also", "they", "they", "asked", "people", "how", "writin...	c("accomplished", "engaged")	Also, they could have asked people how they accompli...
9	1	9	character(0)	c("asking", "if", "people", "have", "questions", "repeate...	effective	Asking if people have questions repeatedly is not effe...
10	1	10	character(0)	c("i", "would", "to", "have", "seen", "more", "dax", "funct...	like	I would like to have seen more DAX functions.
11	1	11	depth	c("we", "covered", "covered", "a", "a", "number", "of", "f...	c("liked", "level")	We covered a number of functions in depth but I would...
12	1	12	character(0)	c("just", "to", "know", "some", "more", "commonly", "us...	character(0)	Just to know some more commonly used functions an...
13	1	13	character(0)	c("more", "on", "dashboards")	focus	More focus on dashboards.
14	1	14	character(0)	c("less", "emphasis", "on", "basic", "functionality")	character(0)	Less emphasis on basic functionality.
15	1	15	depth	c("get", "even", "more", "in", "with", "showing", "exampl...	character(0)	Get even more in depth with showing example of usin...
16	1	16	character(0)	c("this", "was", "done", "very")	well	This was done very well.
17	1	17	character(0)	nothing	character(0)	Nothing.
18	1	18	character(0)	c("the", "course", "material", "was")	perfect	The course material was perfect.
19	1	19	character(0)	c("topic", "details", "test", "cases", "then", "detailed", "...	character(0)	Topic details, test cases then detailed descriptions of ...
20	1	20	character(0)	c("i", "felt", "having", "two", "instructors", "was")	great	I felt having two instructors was great.
21	1	21	character(0)	c("one", "that", "can", "answer", "help", "with", "with", "i...	c("instructor", "teach", "helped")	One that can answer/help with individual questions whi...
22	1	22	character(0)	c("the", "team", "was", "very", "very", "and", "seemed",...	c("amend", "knowledgeable", "efficient")	the Amend Team was very knowledgeable and seeme...
23	1	23	could have	c("maybe", "sending", "out", "a", "more", "detailed", "lis...	ahead	Maybe sending out a more detailed list of functions to...
24	1	24	character(0)	NA	character(0)	.
25	1	25	character(0)	training	excellent	Excellent training!
26	1	26	character(0)	c("i", "would", "it", "to", "to", "be", "interact", "with", "wit...	c("like", "easier", "interested")	I would like it to be easier to interact with the group int...
27	1	27	character(0)	c("course", "is", "packed", "but", "very", "in", "delivering...	c("action", "effective", "well balanced")	Course is action packed but very effective in delivering...

By using **get_nrc_sentiment** function in R, we got the count of each emotion.

	sentiment	Score
1	anger	1
2	anticipation	7
3	disgust	2
4	fear	2
5	joy	5
6	sadness	3
7	surprise	1
8	trust	11
9	negative	2
10	positive	20

By plotting the above table using ggplot2 package, we get the below result.



RESULT

We can see that the positive comments had the highest score of 20 while trust had 11. There was 2 for negative. The algorithm counted would, could, and depth as negative.

The comments that had these words were:

- Get even more in depth with showing example of using Python and R
- It would be awesome, if we could work with our own data sets, while in the course
- Could be a bad idea but based on past trainings working with a partner helps me to think through the exercises

As we read the comments, we get a sense that people liked the training while they had some constructive feedback and suggestion.

ADVANCED TABLEAU TRAINING

Created a table showing the sentiment score for each sentence. Anything below a score of -0.05 we tag as negative and anything above 0.05 we tag as positive. Anything in between inclusively, we tag as neutral. This session had a lot of comments and showing each row of the table is not feasible.

```
sent_tableau <- sentiment(text_tableau, by = NULL)
```

element_id	sentence_id	word_count	sentiment
1	31	2	0.00000000
1	20	4	0.90000000
1	22	6	0.24494897
1	6	7	1.19058809
1	34	8	0.03535534
1	4	9	0.61666667
1	13	10	0.28460499
1	8	11	0.00000000
1	25	11	0.40704032
1	27	12	-0.15877132
1	11	13	0.00000000
1	28	13	0.24961509
1	29	13	0.15254255
1	7	14	0.16837458
1	30	14	0.48107024
1	3	15	0.48024993
1	23	16	-0.03750000
1	21	17	0.19402850
1	32	19	-0.01147079
1	1	20	1.27623580

Result: The mean sentiment for this training session is 0.23. So we can conclude that the overall sentiment is highly positive. To study the sentiments further, I visualized the data.

```{r}
mean(sent_tableau$sentiment)
```

```
[1] 0.2358727
```

By using **extract_sentiment_terms** function, I segrated negative words and the positive ones in each sentence.

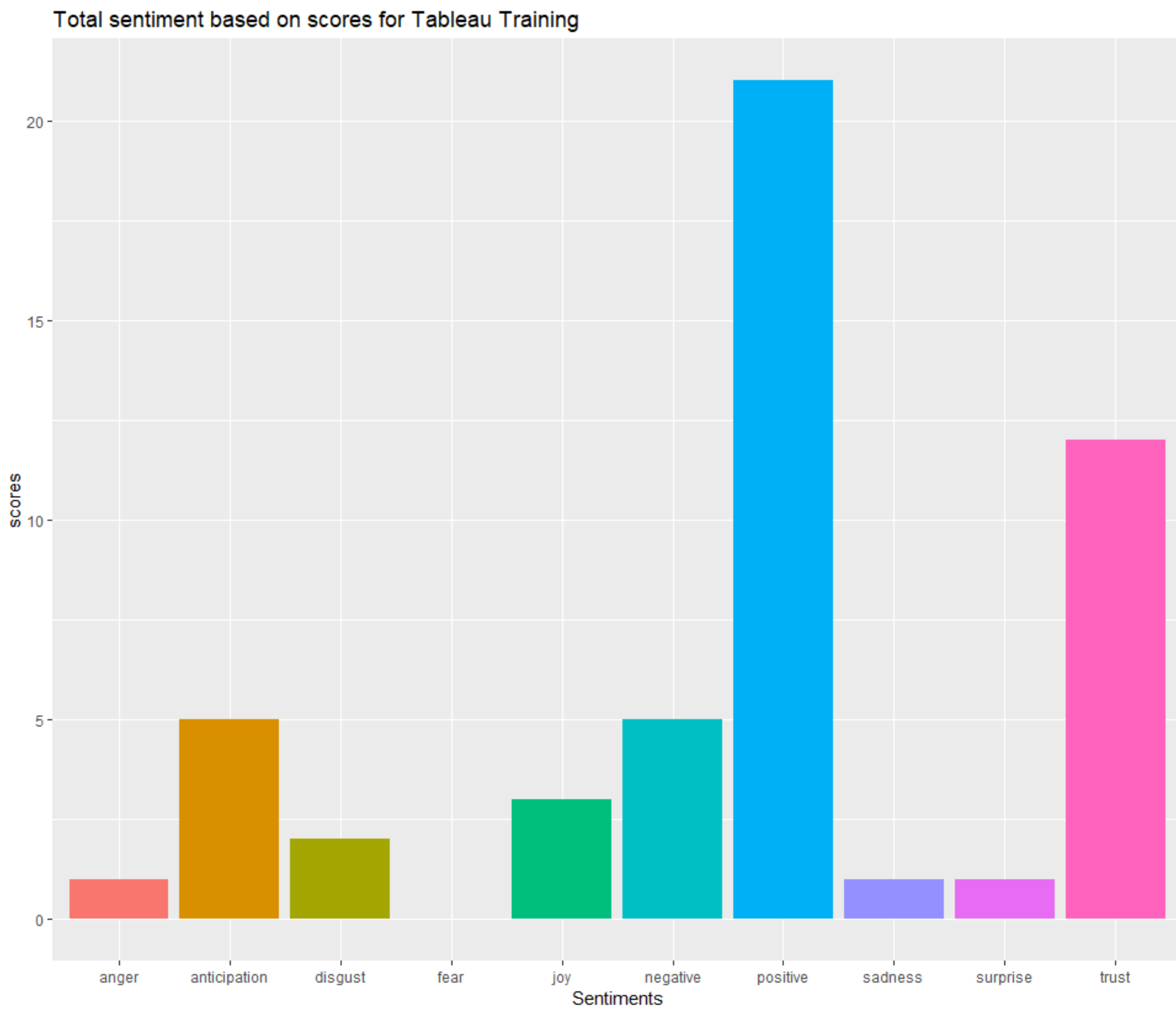
sent_terms_tableau<- extract_sentiment_terms(text_tableau, by = NULL)

element_id	sentence_id	negative	neutral	positive
1	3	character(0)	c("a", "a", "little", "pace", "maybe", "less", "with", "dem...	c("easier", "content", "content", "richer")
1	4	character(0)	c("i", "would", "to", "see", "it", "get", "more")	c("like", "advanced")
1	5	was hoping	c("i", "came", "in", "with", "a", "of", "of", "tableau", "and"...	c("fair", "knowledge", "discussion", "like")
1	6	character(0)	c("a", "more", "of", "course")	c("advanced", "level", "content")
1	7	depth	c("i", "was", "still", "for", "more", "in", "coverage", "of", "f...	hoping
1	8	character(0)	c("i", "was", "looking", "for", "more", "of", "of", "that", "in...	character(0)
1	9	character(0)	c("i", "i", "i", "did", "the", "the", "coverage", "of", "sets", ...	c("like", "like", "helpful", "knowledge", "good", "I
1	10	would have	c("if", "this", "is", "a", "follow", "up", "from", "the", "i", "le...	c("first class", "preferred", "new")
1	11	character(0)	c("spend", "less", "time", "on", "on", "the", "the", "mino...	character(0)
1	12	character(0)	c("felt", "we", "crunched", "in", "topics", "last", "last", "...	c("like", "important", "advanced")
1	13	character(0)	c("the", "the", "was", "at", "explaining", "topics", "discus...	c("instructor", "great")
1	14	character(0)	c("tableau", "should", "be", "an", "course", "rather", "th...	c("advanced", "advanced", "advanced")
1	15	remedial	c("there", "were", "several", "attendees", "who", "who", ...	character(0)
1	16	c("complicated", "should have")	c("several", "of", "of", "the", "more", "topics", "weren't",...	c("favor", "easier", "simpler", "knowledge", "ad
1	17	character(0)	c("more", "chart", "types", "less", "data", "data", "joinin...	c("work", "assist")
1	18	would be	c("in", "terms", "of", "of", "the", "venue", "equipment", "...	conducive
1	19	spent	c("much", "time", "was", "adjusting", "font", "layout", "si...	c("ensure", "clearly")
1	20	character(0)	c("went", "much", "too")	fast
1	21	character(0)	c("there", "was", "no", "time", "to", "think", "or", "or", "e...	learning
1	22	character(0)	c("we", "just", "kept", "plowing", "through")	content
1	23	would have	c("i", "a", "little", "time", "to", "try", "things", "and", "app...	c("liked", "learning")
1	24	miss	c("i", "saw", "a", "lot", "of", "things", "but", "there", "was...	interesting
1	25	character(0)	c("the", "course", "felt", "more", "intermediate", "than",...	advanced
1	26	missed	c("i", "may", "have", "it", "but", "if", "they", "weren't", "lis...	helpful
1	27	would have	c("i", "the", "course", "to", "start", "where", "it", "actuall...	liked
1	28	character(0)	c("i", "was", "was", "very", "in", "the", "the", "sankey", "d...	interested
1	29	would be	c("if", "you", "could", "an", "course", "maybe", "that")	c("offer", "expert", "level", "great")

By using **get_nrc_sentiment** function in R, we got the count of each emotion.

	sentiment	Score
1	anger	1
2	anticipation	5
3	disgust	2
4	fear	0
5	joy	3
6	sadness	1
7	surprise	1
8	trust	12
9	negative	5
10	positive	21

By plotting the above table using ggplot2 package, we get the below result.



RESULT

We can see that the positive comments had the highest score of 21 while trust had 12. There was 0 for fear. The algorithm counted would, could, miss, was hoping, and depth as negative and offer, liked, learning as positive.

The comments that had these words were:

- I liked learning that I could pivot my data and make it Tableau useful
- **If you could offer an expert level course maybe, that would be great**
- I would like to see it get more advanced
- I was still hoping for more in depth coverage of formula's
- I saw a lot of interesting things, but there was no time to write notes or go back for another look, or we would miss the next thing

As we read the comments, we get a sense that people liked the training while they had some constructive feedback and suggestion. We also saw that the sentence containing the word "offer" was more of a feedback than positive. Lexicon algorithm works for measuring the overall sentiment.

MACHINE LEARNING TRAINING

Created a table showing the sentiment score for each sentence. Anything below a score of -0.05 we tag as negative and anything above 0.05 we tag as positive. Anything in between inclusively, we tag as neutral. This session had a lot of comments and showing each row of the table is not feasible.

sent_ML <- sentiment(text_ML, by = NULL)

	element_id	sentence_id	word_count	sentiment
1	1	1	5	0.60373835
2	1	2	NA	0.00000000
3	1	3	5	0.00000000
4	1	4	12	0.14433757
5	1	5	15	0.36147845
6	1	6	25	0.68400000
7	1	7	34	0.40516541
8	1	8	49	0.67000000
9	1	9	10	0.01581139
10	1	10	19	0.41294832
11	1	11	10	-0.07905694
12	1	12	6	0.32659863
13	1	13	27	-0.25980762
14	1	14	37	0.22522661
15	1	15	8	0.63639610
16	1	16	18	-0.04714045
17	1	17	NA	0.00000000
18	1	18	11	0.25628464
19	1	19	13	-0.22188008
20	1	20	NA	0.00000000

We can see a few NA values. That is because people entered the textbox and left without writing any valid word. The comments contain spaces. To find the mean, I removed NA values.

```
> library(tidyr)
> sent_ML %>% drop_na()
```

```
> colSums(is.na(sent_ML))
 element_id sentence_id  word_count   sentiment 
          0           0           3           0 
> cat(which(is.na(sent_ML$word_count)==TRUE))
2 17 20
> sent_ML <- sent_ML[-c(2,17,20),]
> colSums(is.na(sent_ML))
 element_id sentence_id  word_count   sentiment 
          0           0           0           0 
>
```

I found out which rows had NA values and then I removed those rows for further analysis.

```
> mean(sent_ML$sentiment)
[1] 0.2280998
```

Result: The mean sentiment for this training session is 0.228. So we can conclude that the overall sentiment is highly positive. To study the sentiments further, I visualized the data.

By using **extract_sentiment_terms** function, I segrated negative words and the positive ones in each sentence.

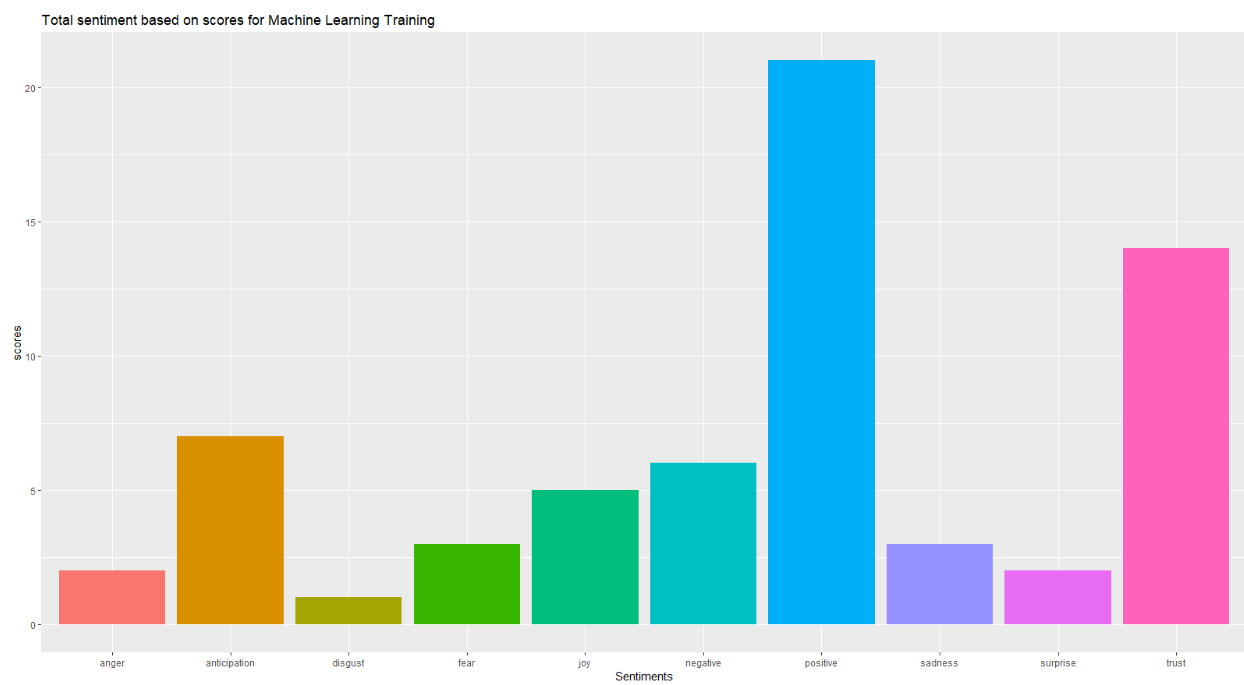
negative	neutral	positive	sentence
character(0)	c("the", "course", "was", "really")	good	The course was really good!
character(0)	NA	character(0)	.
character(0)	c("a", "lot", "condensed", "in", "days")	character(0)	A lot condensed in 2 days.
character(0)	c("we", "could", "shrink", "the", "the", "the", "scope", "or...	increase	We could shrink the scope or increase
character(0)	c("i", "would", "to", "see", "some", "more", "theory", "an...	c("like", "logical")	I would like to see some more theory
character(0)	c("for", "me", "the", "the", "the", "most", "part", "of", "of...	c("valuable", "practical", "model")	For me, the most valuable part of the
c("would have", "would have", "spent")	c("spending", "more", "time", "time", "walking", "throug...	c("helpful", "quickly", "greater", "focus", "helped")	Spending more time walking through
issue	c("i", "got", "a", "a", "lot", "lot", "more", "more", "more",...	c("instructor", "liking", "content", "relevant", "actionabl...	I got a lot more out of Day 1 than Day
spent	c("i'd", "to", "have", "more", "time", "on", "both", "days")	like	I'd like to have spent more time on bo
character(0)	c("day", "was", "a", "a", "revision", "but", "i", "still", "got"...	new	Day 1 was a revision, but I still got a lo
character(0)	c("i", "don't", "with", "these", "topics", "on", "a", "daily", ...	work	I don't work with these topics on a dail
character(0)	c("day", "was", "all", "to", "me")	new	Day 2 was all new to me.
c("break", "would have", "depth")	c("perhaps", "you", "can", "further", "the", "the", "class"...	option	Perhaps you can further break the cla
break	c("you", "you", "could", "could", "a", "broad", "overview"...	c("provide", "offer", "practice")	You could provide a broad overview of
character(0)	c("course", "materials", "on", "day", "needed", "more", ...	quality	Course materials on Day 2 needed m
c("missing", "improperly")	c("a", "large", "percentage", "of", "the", "the", "slides", "...	saved	A large percentage of the slides were
character(0)	NA	character(0)	.
character(0)	c("i", "it", "to", "follow", "pdp", "an", "shapley", "values", ...	c("found", "tough")	I found it tough to follow PDP an Shap
character(0)	c("however", "i", "don't", "have", "any", "feedback", "ho...	better	However, I don't have any feedback h
character(0)	NA	character(0)	.
would be	c("this", "way", "it", "to", "see", "the", "differences", "acr...	easier	This way it would be easier to see the
character(0)	c("i", "plan", "to", "come", "back", "next", "year")	character(0)	I plan to come back next year.

By using **get_nrc_sentiment** function in R, we got the count of each emotion.

	sentiment	Score
anger	anger	2
anticipation	anticipation	7
disgust	disgust	1
fear	fear	3
joy	joy	5
sadness	sadness	3
surprise	surprise	2
trust	trust	14
negative	negative	6
positive	positive	21

I plotted the sentiments chart using ggplot2 library.

```
ggplot(data=SentimentScores_ML,aes(x=sentiment,y=Score))+geom_bar(aes(fill=sentiment),stat = "identity")+
  theme(legend.position="none")+
  xlab("Sentiments")+ylab("scores")+ggtitle("Total sentiment based on scores for Machine Learning Training")
```



We can see that the positive comments had the highest score of 21 while negative had 6. There were 9 for fear, disgust, anger, and sadness. The algorithm counted “would have”, “improperly” etc. as negative and “good”, “logical”, “practical” as positive.

The comments that had these words were:

- Overall good class, other than there was confusion on the start time.
- The course was really good!
- A large percentage of the slides were missing the images because they were saved from web pages improperly.
- **I would like to see some more theory and logical explanations of the ML algorithms.**

As we read the comments, we get a sense that people liked the training while they had some constructive feedback and suggestion. We also saw that the sentence containing the word "logical" was more of a constructive feedback than positive. Lexicon algorithm works for measuring the overall sentiment.

BIG DATA TRAINING

Created a table showing the sentiment score for each sentence. Anything below a score of -0.05 we tag as negative and anything above 0.05 we tag as positive. Anything in between inclusively, we tag as neutral. This session had only 9 comments because less people participated in this session.

element_id	sentence_id	word_count	sentiment
1	1	14	0.1603567
1	2	16	0.6750000
1	3	4	0.0000000
1	4	6	0.5511352
1	5	7	0.9524705
1	6	5	0.5142956
1	7	32	0.1149049
1	8	4	0.2500000
1	9	2	0.3535534

Result: The mean sentiment for this training session is 0.40. So we can conclude that the overall sentiment is highly positive. To study the sentiments further, I visualized the data.

```
> mean(sent_BigData$sentiment)
[1] 0.3968574
```

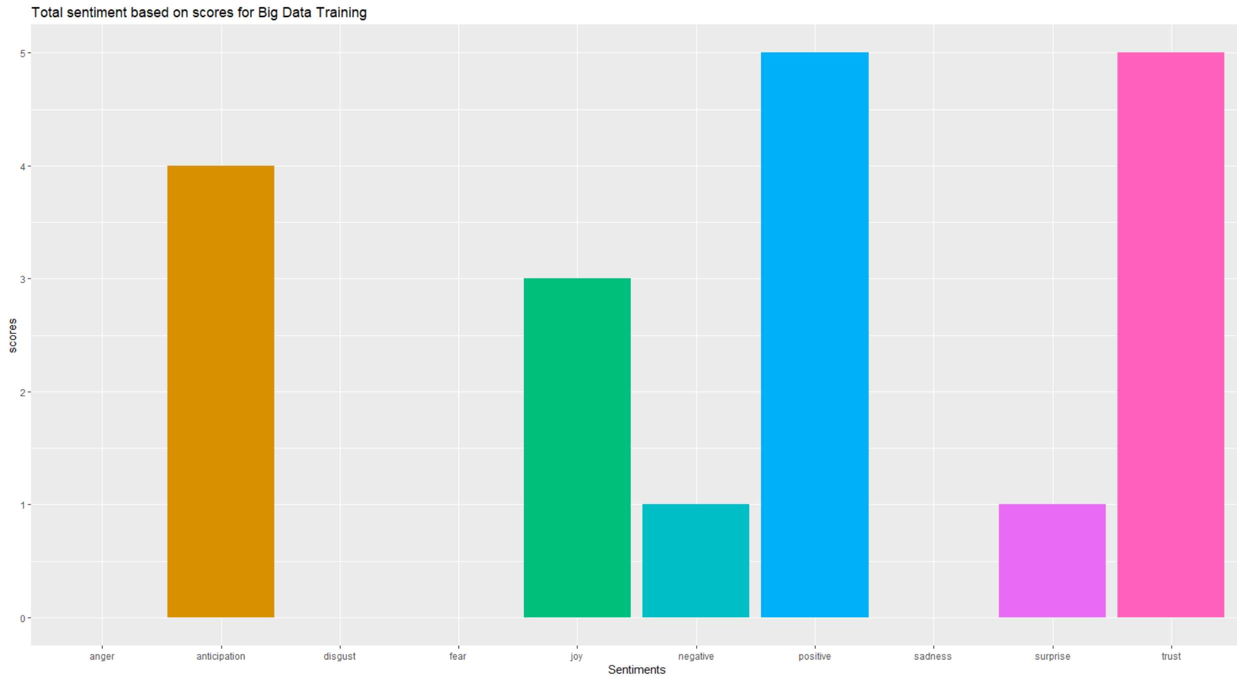
By using **extract_sentiment_terms** function, I segrated negative words and the positive ones in each sentence.

element_id	sentence_id	negative	neutral	positive	sentence
1	1	character(0)	c("i", "would", "to", "see", "a", "demo", "of", "of", "teraby...	c("like", "spark")	I would like to see a demo of terabytes of data analyz...
1	2	character(0)	c("i", "thought", "thought", "the", "course", "was", "out",...	c("well", "well", "great", "content")	I thought the course was well thought out, well presen...
1	3	character(0)	c("perhaps", "more", "live", "demos")	character(0)	Perhaps more live demos.
1	4	character(0)	c("both", "the", "instructors", "were", "really")	good	Both the instructors were really good.
1	5	character(0)	c("they", "deep", "of", "the", "topic")	c("demonstrated", "understanding")	They demonstrated deep understanding of the topic.
1	6	character(0)	c("andrew", "was", "a")	c("fantastic", "instructor")	Andrew was a fantastic instructor.
1	7	c("complaints", "freezing")	c("my", "my", "only", "about", "the", "the", "the", "two", ...	c("found", "properly")	My only complaints about the two days were 1) I tried ...
1	8	character(0)	c("the", "course", "was")	great	The course was great!
1	9	character(0)	you	thank	Thank you!

By using **get_nrc_sentiment** function in R, we got the count of each emotion.

sentiment	Score
anger	0
anticipation	4
disgust	0
fear	0
joy	3
sadness	0
surprise	1
trust	5
negative	1
positive	5

I plotted the sentiments chart.



We can see that the positive comments had the highest score of 5 while negative had 1. There was 0 for fear, disgust, anger, and sadness. There were very less negative words like “freezing” and “complaints”. The algorithm counted “great”, “good”, “well” etc as positive. We see that “Spark” is counted as positive but it’s the name of the technology.

The comments that had these words were:

- **I would like to see a demo of terabytes of data analyzed through Spark.**
- I thought the course was well thought out, well presented, and had a lot of great content.
- Both the instructors were really good.

As we read the comments, we get a sense that people liked the training while they had some constructive feedback and suggestion. We also saw that the sentence containing the word “Spark” was more of a constructive feedback than positive. Lexicon algorithm works for measuring the overall sentiment.

RESULT

Training Session	Mean Sentiment Score
MS Power BI	0.19
Advanced Tableau	0.23
Big Data	0.39
Machine Learning	0.228

Big data class has the highest sentiment score. However, it has the least number of survey respondents. Based on these results, we can conclude that Big Data was the most popular class but Tableau and Machine learning also have similar score and those classes were popular too.