

Business Insight Report:

Shoe Brands and Running

Noor Hejeeali

Hult International Business School

Text Analytics and Natural Language Processing (NLP) – DAT -5317

Professor Thomas Kurnicki

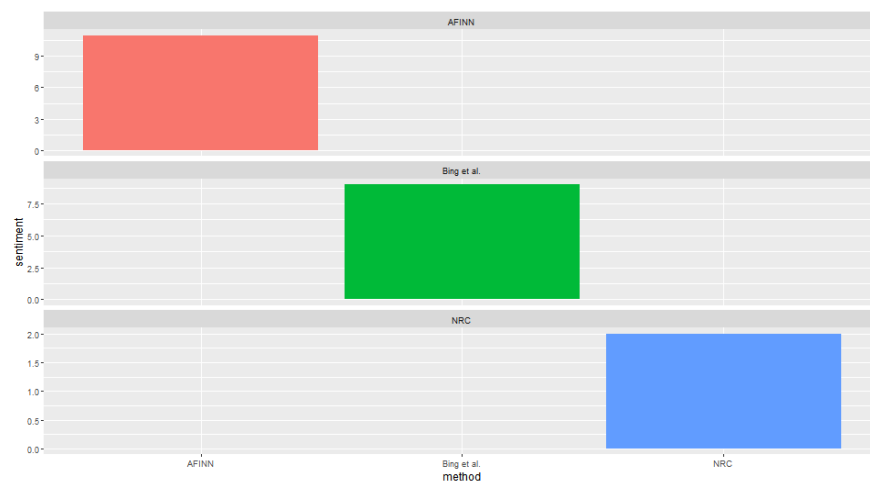
December 5th, 2021

Abstract

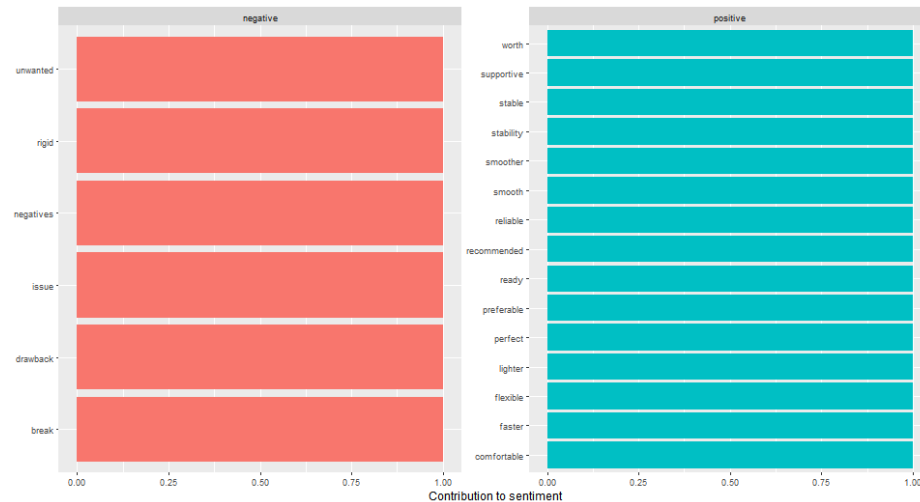
This business insight report is about four different running shoes from four different brands. Each shoe is analyzed through various frameworks to give recommendations on which shoe to purchase. The R code and R output is included in the appendix. The abstract, references page, and appendix are not included in the word count.

Finding the right running shoes is difficult but very important. Every person has a different preference on what they are looking for. Some look for comfort and stability while others look for a lightweight minimal support shoe. Some of the popular running brands are Brooks, Mizuno, Asics, and Nike. I picked 4 shoes which were Mizuno Wave Rider, Nike Pegasus, Asics Gel Kayano, and Brooks Adrenaline to analyze. The sentiment analysis, term frequency, tf-idf, n-grams analysis was used to get a better understanding of each brand. This analysis helps to understand each shoe and recommend the best one to purchase.

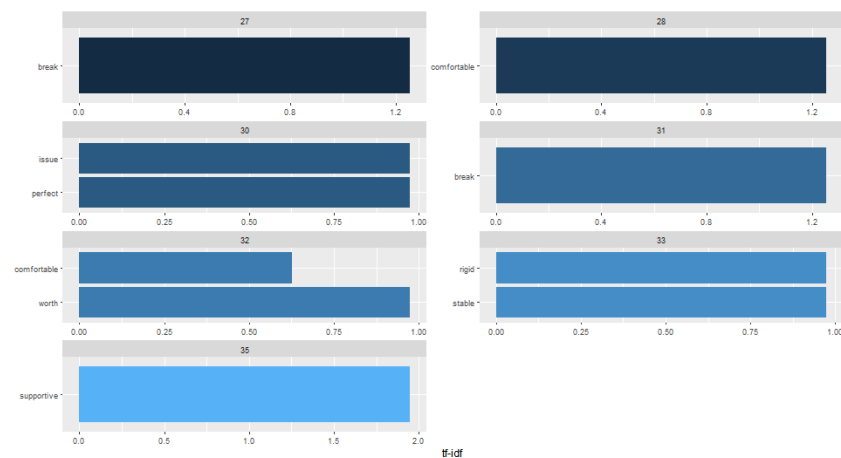
The first brand is Brooks. From the sentiment analysis, a variety of positive and negative words were shown. Even with the negative words, the AFINN and Bing charts showed no negative numbers. This is shown in the chart below.



The NRC gives us insight into what types of words were used. It helps us understand the emotions but also whether a word is a positive or negative word. A variety of positive and negative words were used. The chart below shows the different negative and positive words. The negative words which were surprising were break, rigid, and unwanted. Rigid is an interesting negative word because comfortable was a positive word.

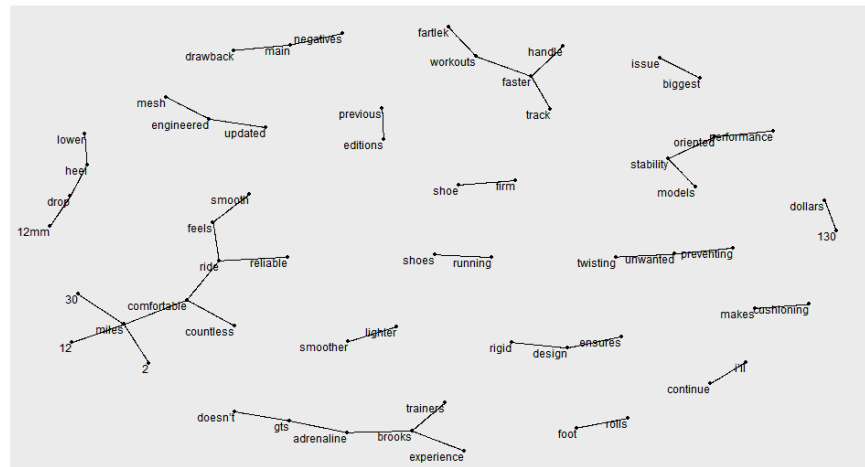


To dig deeper into the positive and negative words from the NRC, term frequency and tf-idf was done. Comfortable was used 51 times. Rigid was used 35 times. Issue and break were used frequently too. Rigid had a term frequency of 0.028 while comfortable had a term frequency of 0.038. This means that these terms were unique words and not common words used. The tf-idf help to define unique words which were used. The chart below shows the different unique terms used along with the calculated tf-idf.

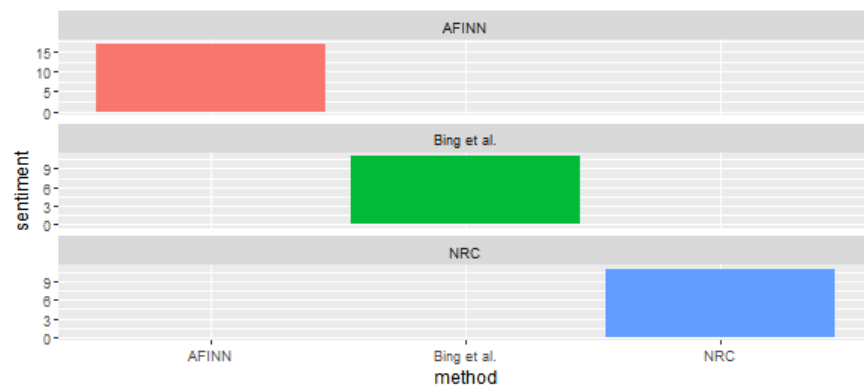


Besides this, the n-gram analysis was done to understand different words which were used together. The bigram was used because I wanted to see which 2 tokens are used together. The chart below shows the different words used together. The words that stood out were foot rolls,

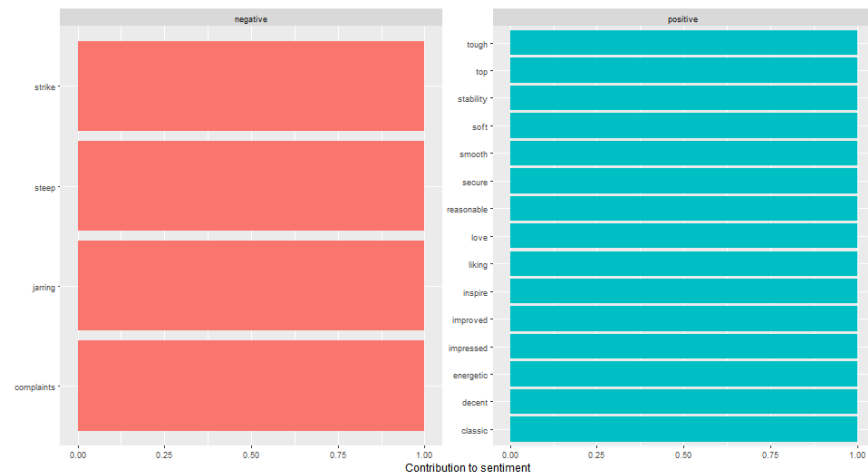
rigid design, and shoe firm. Foot rolls stood out because it seems this word was related to an injury. If a shoe is known for injuries, then a person may not want to purchase. It could be that the rigid design and shoe firmness caused the injury.



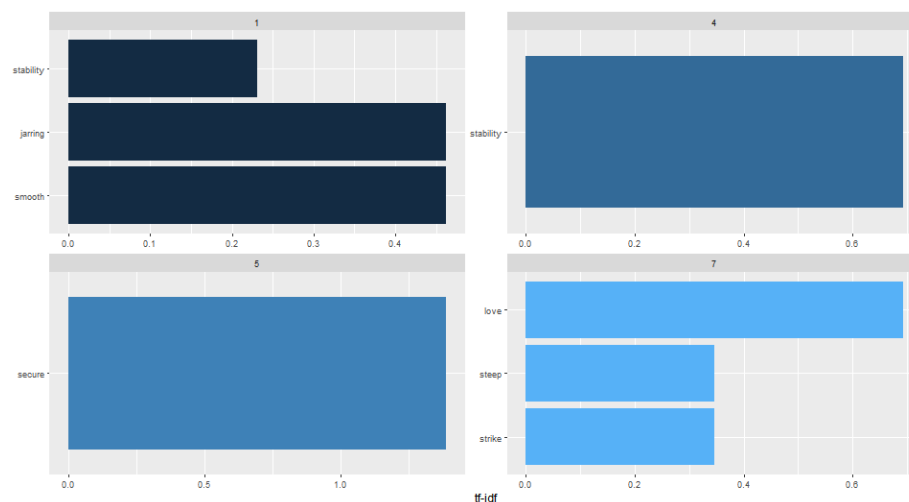
The second brand is Mizuno. The chart below shows that AFINN had a higher score for Mizuno compared to Brooks. There were no words that were associated with -1 for it. Similar to Brooks, Mizuno had no negative amounts associated with it.



The chart below shows the different positive and negative words associated with Mizuno Wave Rider shoe. Strike and complaints were negative words that struck out. This means that there were complaints about this specific shoe. Some of the positive words were secure, reasonable, and classic. Even though the negative word complaints was used, people believed the shoe was a classic too.

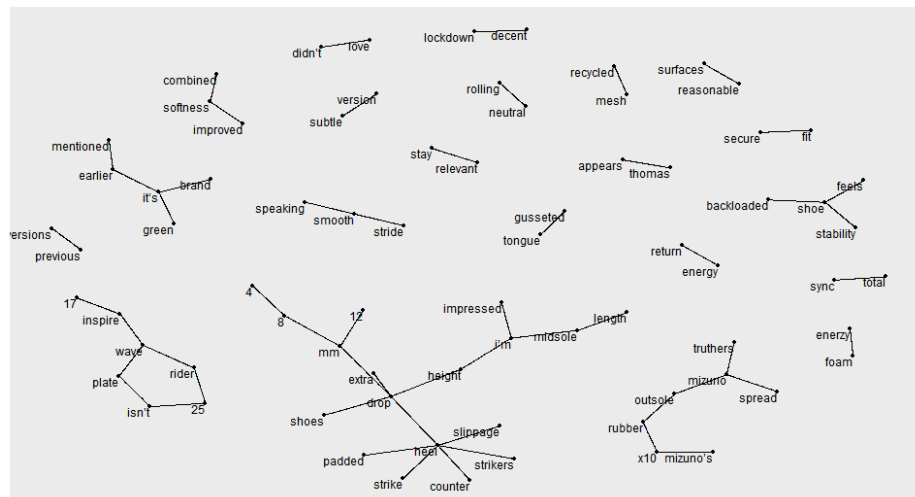


Stability and jarring were two of the most used terms. Jarring was used 72 times while stability was used 79 times. Strike was used 60 times. It is bad because two of the negative words are used frequently. Jarring and stability were in rank 1 but stability had a high tf-idf. Stability was not a unique word, but a frequent word used. Strike had a low tf-idf which was 0.346 meaning it was a unique word.

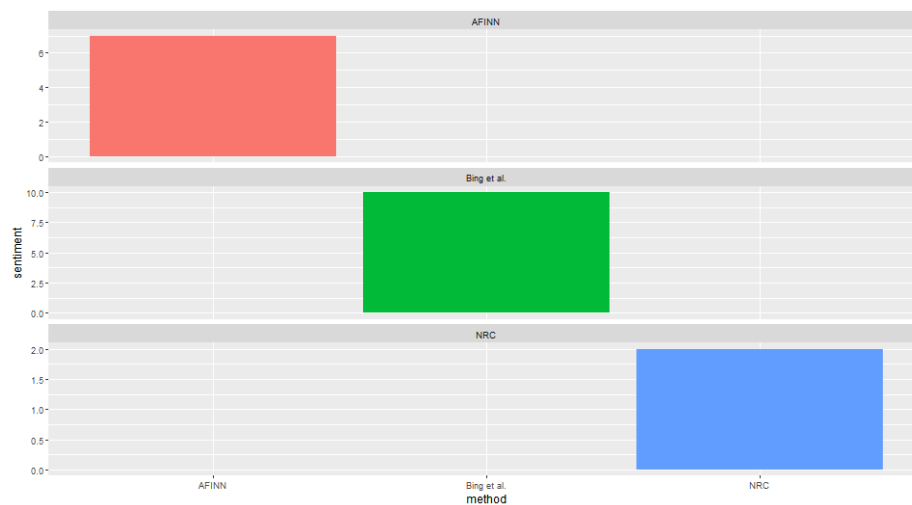


The words that struck out in the bigram chart shown below are heel strikers, heel slippage, and extra drop. From the sentiment analysis and tf-idf, strike was unique negative word used. It

seems that many people had issues with the heel design of this shoe. The extra drop may have caused it.

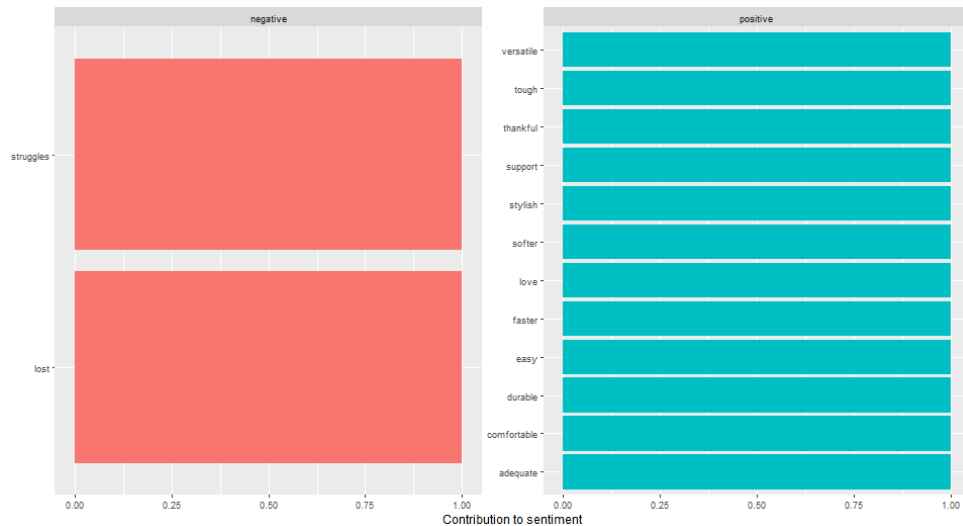


The third brand is Nike. The chart below shows that the AFINN is like Brooks and Mizuno. Also, Nike had no negative amounts too in Bing and AFINN. The NRC range was lower compared to Brooks and Mizuno.

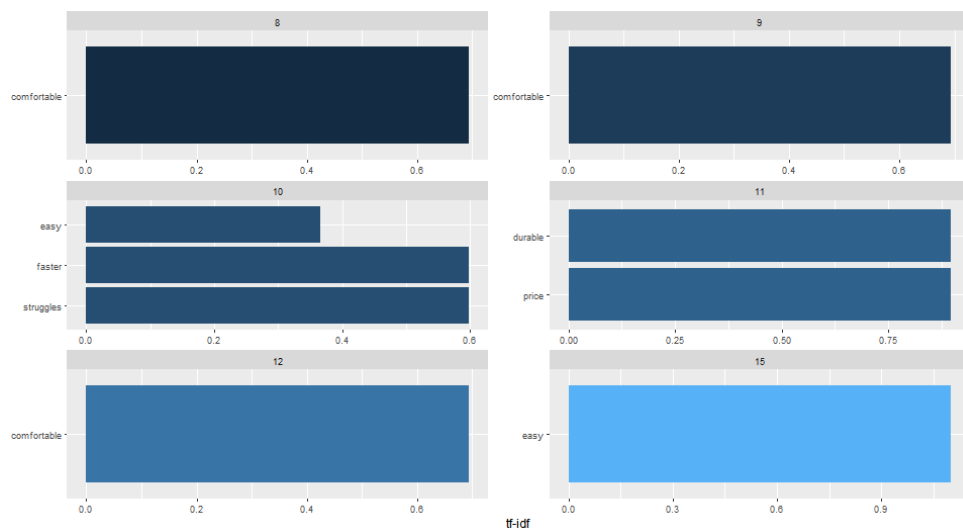


The chart below shows the various positive and negative words associated with the Nike Pegasus shoe. The negative word that struck out was struggles because it could mean something is wrong with design or fit of the shoe. Some of the positive words were versatile, comfortable,

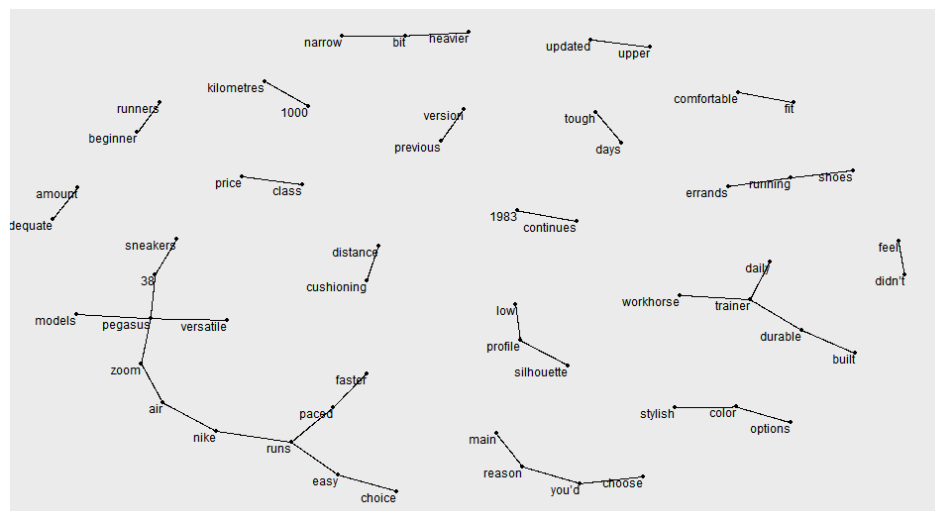
and support. Versatile was interesting because people may use a shoe for other activities besides running.



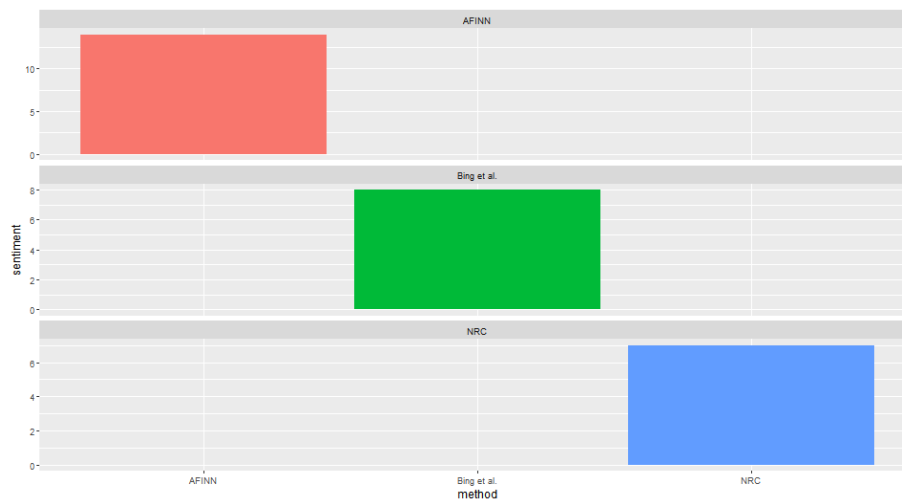
Easy was one of the most used terms with an amount of 51. Durable and price were used often too with an amount of 38. Struggles was used 28 times. The negative words were used less frequently here compared to the other 2 brands. Even though easy was used many times, it had a high tf-idf of 1.09. This means it is not a unique word. Struggle is a unique word used because the tf-idf was 0.59 for it. This is shown in the chart below.



From the bigram chart below, the words that struck out were beginner runners, bit narrow, and daily trainer. If a person is has just started running or is looking for a multipurpose shoe, then the Nike Pegasus shoe may be a good fit. The downside is the narrow fit which may not work for everyone.

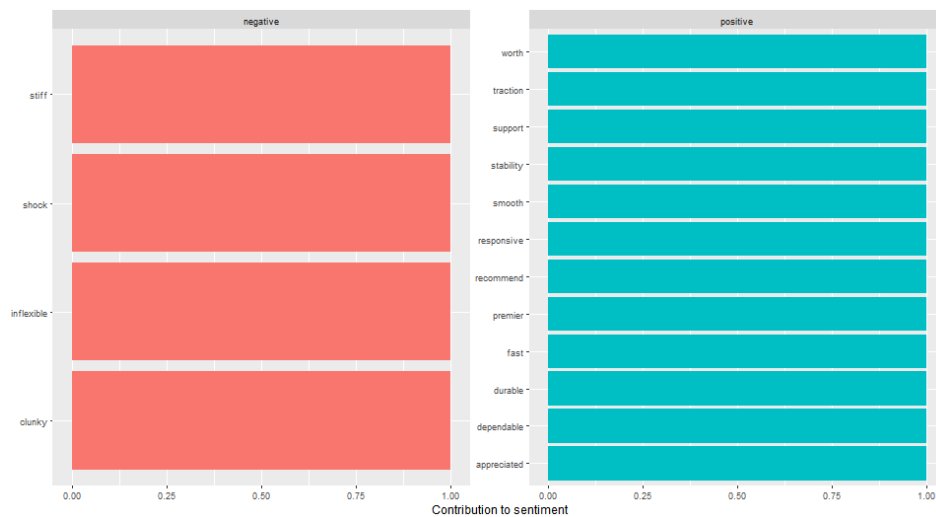


The final brand is Asics. Similar to the other brands, the AFINN and Bing had no negative amounts. The NRC range low compared to Brooks and Mizuno. This is shown in the chart below.

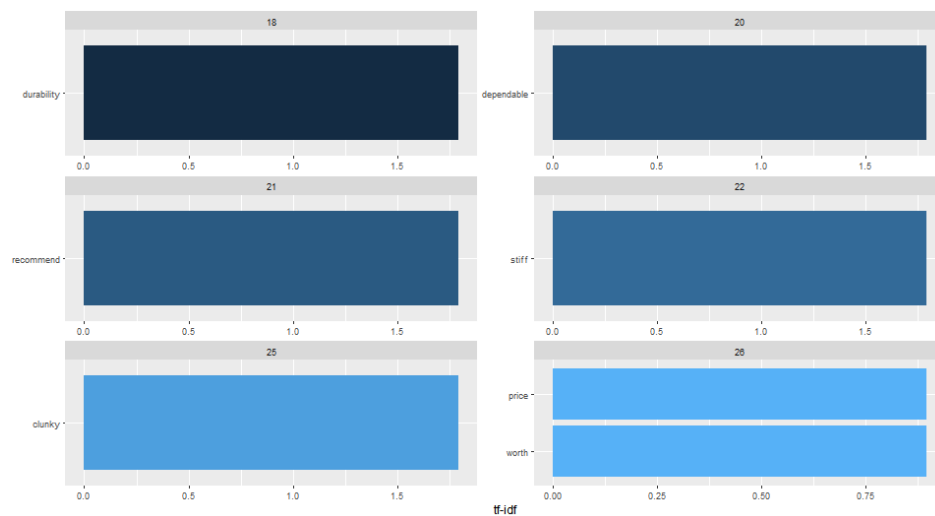


In the chart below it shows the different positive and negative words associated with the Asics Gel Kayano shoe. The negative words stiff and clunky were interesting. Stiff is like the negative

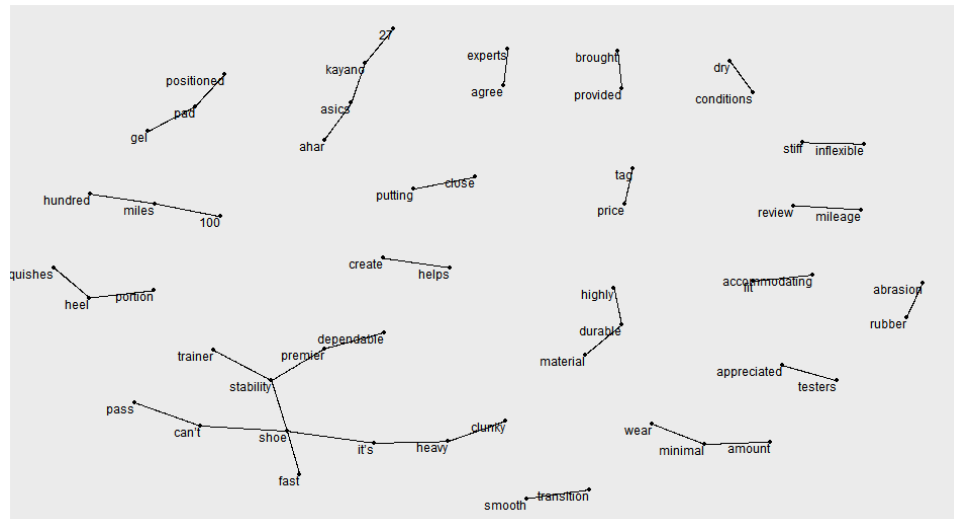
word rigid used for Brooks. Some of the positive words used were support, dependable, and durable.



The terms durability and clunky were the most used. Durability was used 33 times while clunky was used 28 times. Even though these two terms were used the most, the tf-idf was very high. The tf-idf was 1.791 which means these words are frequent and not unique. The unique words were price and worth which had a tf-idf of 0.895. The chart below shows the tf-idf for the different words.



The bigram chart below shows the different 2 token terms used together. The words that caught my eye were appreciated testers, hundred miles, experts agree, and stability trainer. If a runner who is looking for a long-distance shoe, then the Asics Gel Kayano may be a good fit. Also, experts agree and appreciated testers stood out because it shows credibility for the brand and shoe.



From the different analysis done, I would recommend not purchasing the shoes Brooks Adrenaline and Mizuno Wave Rider. The reasons why I would recommend against purchasing is because of the impact of the negative words such as strike and rigid. The negative words were used a high amount of time. The negative words were used more frequently for these 2 shoes compared to others. Also, the bigrams chart showed words that stuck out. These words include heel slippage, foot rolls, rigid design, and extra drop. Rigid was a negative word used frequently for Brooks Adrenaline. Since strike was a negative word used for Mizuno then this may be why the heel slippage may be occurring.

Depending on what a person is looking for I would recommend purchasing the Nike Pegasus or Asics Gel Kayano shoe. I would recommend a person who is new to running and looking for a versatile shoe to purchase the Nike Pegasus. Through the sentiment and tf-idf analysis, price and

durable were used. A new runner may be on a budget and wants a cheaper durable shoe. Also, the bigram chart showed that beginner runner and daily trainer were used. The Nike Pegasus would be a good fit for them. I would recommend a person who runs often or is a marathon runner to purchase the Asics Gel Kayano shoe. Through the sentiment and tf-idf analysis, durability was a term used often. Worth was not a frequent term used. Also, the bigrams chart shows words used such as hundred miles and experts agree.

In the end I would recommend a beginner runner to purchase the Nike Pegasus Shoe and a regular runner to purchase the Asics Gel Kayano shoe. I would not recommend purchasing the Brooks Adrenaline and Mizuno Wave Rider shoes. This recommendation was given after completing the different analysis and visualizations. These analyses include sentiment, term frequency, tf-idf, and n-grams.

References

Eisinger, A. (2021). *The Nike Air Zoom Pegasus 38 is the perfect shoe for all types of runners.*

SELF. <https://www.self.com/review/nike-pegasus-38-review>.

Ellis, C. (2021). *Brooks Adrenaline GTS 21 review.* TechRadar.

<https://www.techradar.com/in/reviews/brooks-adrenaline-gts-21>.

Jones, T. J. (2020). *Asics Gel Kayano 27 Review.* Running Shoes Guru.

<https://www.runningshoesguru.com/2020/06/asics-gel-kayano-27-review/>.

Langelier, A., Haines, R. (2021). *Mizuno Wave Rider 25 performance review "believe in the*

run. Believe in the Run. <https://www.believeintherun.com/mizuno-wave-rider-25-performance-review/>.

Law, B. (2021). *Nike Zoom Pegasus 38 review.* Running Shoes Guru.

<https://www.runningshoesguru.com/2021/05/nike-zoom-pegasus-38-review/>.

Law, B. (2021). *Mizuno Wave Rider 25 review.* Running Shoes Guru.

<https://www.runningshoesguru.com/2021/08/mizuno-wave-rider-25-review/>.

Matsumoto, E. (2020). *Shoe review: Asics Gel-Kayano 27.* Fleet Feet.

<https://www.fleetfeet.com/blog/shoe-review-asics-gel-kayano-27>.

Metzler, B. (2020). *BROOKS ADRENALINE GTS 21 REVIEW.*

JackRabbit.<https://www.jackrabbit.com/info/blog/brooks-adrenaline-gts-21-review.html>.

Petruny, M. (2021). *Nike gives the 38th Air Zoom Pegasus just the quick tune-up it needed.*

Runner's World. <https://www.runnersworld.com/gear/a37282601/nike-air-zoom-pegasus-38-review/>.

Ronto, P. (2020). *Asics Gel Kayano 27 - review 2021 - facts, deals (\$89).* Athletic shoe reviews.

<https://runrepeat.com/asics-gel-kayano-27>.

Unknown. (2021). *Gel-Kayano*. ASICS. <https://www.asics.com/us/en-us/gel-kayano/c/aa50101000/>.

Unknown. (2021). *How to choose the right shoes for you*. Runners Need. <https://www.runnersneed.com/expert-advice/gear-guides/choosing-the-right-running-shoes.html>.

Unknown. (2021). *Running Shoes Reviews*. Running Shoes Guru. <https://www.runningshoesguru.com/reviews/nike/pegasus/>.

Unknown. (2021). *The new adrenaline GTS 22*. Brooks Running. https://www.brooksrunning.com/en_us/brooks-adrenaline-gts-running-shoes/.

Unknown. (2021). *Wave rider 24*. Mizuno USA Official Website. <https://www.mizunousa.com/wave-rider-24>.

Unknown. (2021). *Brooks Adrenaline GTS 21 review*. Running Shoes Guru. <https://www.runningshoesguru.com/2021/01/brooks-adrenaline-gts-21-review/>.

Unknown. (2021). *Brooks Adrenaline GTS 21: Reviews and full analysis*. Runner's Lab. <https://runnerslab.com/review/brooks-adrenaline-gts-21/>.

Appendix

R Code:

```
.libPaths("C:/Rlibs")

#Mizuno Reviews Text
library(dplyr)
library(tidytext)

#Mizuno Reviews
text1 <- c("Enerzy is what Mizuno needed to stay relevant in the game. The Wave Rider 25 is energetic with the energy return and bounce that hits just right without being too soft or too jarring. The midsole and Wave Plate are in total sync with each other to make every stride smooth. Speaking of that plate, it provides just a little bit of stability for those who don't quite pronate to the extreme.")
text2 <- c("I'm impressed that Mizuno kept the slipper-like step-in feel that has decent lockdown. When I first tried these on, I was reminded of what made the previous versions good for their era. However, these aren't the Riders of the early 2000s, as Mizuno does a good job of blending the new with the classic. If you like Mizuno, you will like this shoe, and it'll probably last forever.")
text3 <- c("Mizuno's X10 rubber outsole is tough as a bowl of nails, without any milk. I put it through plenty of reasonable surfaces around town, yet it keeps coming back like it's brand new.")
text4 <- c("On top of the outsole, Mizuno spread its Enerzy foam from the heel to the rest of the shoe for a full-length midsole. I'm a big fan of the improved softness combined with this more subtle version of the Wave Plate. The Wave Rider 25 isn't as much of a stability shoe, so the Wave Plate isn't nearly as extreme as the one on the Wave Inspire 17. This shoe feels much more neutral rolling through your stride.")
text5 <- c("Moving to the upper, I have no complaints about the recycled mesh. As mentioned earlier, it's green in more ways than one which is a huge plus for me. A well-padded heel counter keeps my foot in place and eliminates any heel slippage. The gusseted tongue also contributes to the secure fit, but let's not mention it too much (say "gusseted tongue" three times and Thomas appears, just like Beetlejuice).")
text6 <- c("This is a backloaded shoe, with much more stack in the heel (hello, 12 mm drop). Heel strikers will like this shoe, but the midfooters of the world, like myself, may feel a little left out. That 12 mm drop is noticeable, especially if you're used to 4-8 mm drop shoes.")
text7 <- c("The Mizuno truthers out there won't like this take: I don't like the 12 mm drop. I didn't love it in the Wave Inspire 17, and it doesn't work here either. It pushes me to heel strike and is a bit too steep for my liking. If you can teach me to love the extra drop height, I'm all ears.")

mizuno_rating <- rbind(text1, text2, text3, text4, text5, text6, text7)
colnames(mizuno_rating)[1] <- "text"
mizuno_rating <- as.data.frame(mizuno_rating)

id <- c(1,2,3,4,5,6,7)
```

```

id <- matrix(id, ncol=1)

mizuno_rating <- cbind(id,mizuno_rating)

tidy_mizuno <- mizuno_rating %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  count(word) %>%
  arrange(desc(n))

#Nike Pegasus Review
text8 <- c("When I tried it on for the first time, the updated upper felt way softer and more
comfortable than the previous version. It fit like a glove.")
text9 <- c("On that first run, the ride felt really comfortable with plenty of long-distance
cushioning but it didn't feel like the versatile Pegasus models of days gone by.")
text10 <- c("The Pegasus 38 is more suited to easy runs and struggles with faster-paced runs
because of how much softer it is. This workhorse has lost its wings.")
text11 <- c("The main reason you'd choose the Pegasus 38 over any other daily trainer in its
price class is durability. It feels like a really well-built, durable trainer with an outsole that could
easily do over 1000 kilometres.")
text12 <- c("It's also a great option for beginner runners who only want to buy one pair of shoes
that has a comfortable fit and will last a very, very long time.")
text13 <- c("It's on those tough days that I'm especially thankful for shoes like the Nike Air
Zoom Pegasus 38. The workhorse trainer—which was first introduced in 1983—continues to be
a shoe that any runner, at any level, can use and love.")
text14 <- c("I always find that Nike runs a bit narrow and a bit small. ")
text15 <- c("In many ways the Nike Air Zoom Pegasus 38 sneakers really are the Goldilocks of
running shoes: Just the right amount of cushion and responsiveness, plus a low-profile silhouette
and several stylish color options that make these an easy choice to wear while running errands or
with any athleisure look.")
text16 <- c("Though they're a bit heavier than one might expect, I didn't really notice this until I
was on slightly longer runs. Perhaps best of all, they provided an adequate amount of support so
that my knees and hips felt good from start to finish.")

nike_rating <- rbind(text8, text9, text10, text11, text12,text13,text14,text15,text16)
colnames(nike_rating)[1] <- "text"
nike_rating <- as.data.frame(nike_rating)

id <- c(8,9,10,11,12,13,14,15,16)
id <- matrix(id, ncol=1)

nike_rating <- cbind(id,nike_rating)

tidy_nike <- nike_rating %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%

```



```
count(word) %>%
  arrange(desc(n))
```

#Asics Gel Kayano 27 Review

```
text17 <- c("The heel portion is firm and responsive while the forefoot is plushy and springy.")
text18 <- c("Durability is a strength of the Kayano 27. They show minimal wear even after
putting close to 100 miles on them. These are built to last through many miles at many different
paces.")
```

```
text19 <- c("The material used for the outsole is AHAR (ASICS High Abrasion Rubber), a
highly durable material that provides traction in both wet and dry conditions. After a hundred
miles, a minimal amount of wear showed on the outsole.")
```

```
text20 <- c("Overall, the ASICS Kayano 27 is a dependable premier stability trainer. The ride
they provided brought me back to them again and again after my review mileage.")
```

```
text21 <- c("If you are a runner looking for a premier stability shoe that delivers every time you
lace them up I would recommend the ASICS Kayano 27.")
```

```
text22 <- c("These aspects can cause the back portion of the trainers to be inflexible, stiff and
heavy.")
```

```
text23 <- c("Two of the most noticeable things about the Kayano, though, are its step-in feel and
accommodating fit.")
```

```
text24 <- c("The GEL pad, positioned in the heel, squishes when you land to soak up shock and
disperses it evenly throughout the shoe. It also helps create a smooth transition from heel to toe,
which testers appreciated.")
```

```
text25 <- c("All experts agree this shoe can't pass as a fast shoe. It's heavy, clunky, has a lot of
support and, therefore, can't help you pick up the pace. ")
```

```
text26 <- c("While the price tag is high, those who commented on it said Kayano 27 is worth the
money. ")
```

```
asics_rating <- rbind(text17, text18, text19, text20, text21, text22, text23, text24, text25, text26)
```

```
colnames(asics_rating)[1] <- "text"
```

```
asics_rating <- as.data.frame(asics_rating)
```

```
id <- c(17, 18, 19, 20, 21, 22, 23, 24, 25, 26)
```

```
id <- matrix(id, ncol=1)
```

```
asics_rating <- cbind(id, asics_rating)
```

```
tidy_asics <- asics_rating %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  count(word) %>%
  arrange(desc(n))
```

#Brooks Adrenaline GTS 21 Review

```
text27 <- c("I knew from experience Brooks trainers take a little longer to break-in. ")
```

```
text28 <- c("They provided a natural and comfortable ride from the first stride. By the end of the
run, I was looking forward to more miles in them.")
```

```

text29 <- c("It was a reliable ride whether it was 2 miles or 12 miles. The midsole was built to
last for hundreds of miles.")
text30 <- c("Although a great shoe, and one I'll continue to use, the shoe is not perfect. There are
two main negatives that linger in my mind. The biggest issue for me was the 12mm drop.")
text31 <- c("The other main drawback for me was the break-in time. Most running shoes are
ready to go straight out of the box, yet these took more than 30 miles to get a good feel for
them.")
text32 <- c("Priced at 130 dollars, the Brooks Adrenaline is worth every penny. They provided
me with countless comfortable miles because of the execution of all aspects.")
text33 <- c("As expected, the ride feels smooth and very stable. This is definitely a firm shoe, but
not unpleasantly so. The rigid design ensures good transfer of energy as your foot rolls, while
preventing unwanted twisting.")
text34 <- c("While this shoe is lighter, smoother and more flexible than previous editions, the
Adrenaline GTS doesn't have the liveliness to handle faster track and fartlek workouts as well as
some performance-oriented stability models.")
text35 <- c("The cushioning makes it a great shoe for longer rides and daily use. The upper is
breathable and supportive with an updated engineered mesh. ")
text36 <- c("It's not recommended for sprints or faster workouts. Some users also suggested that
a lower heel drop-off would be preferable.")

brooks_rating <- rbind(text27,text28,text29,text30,text31,text32,text33,text34,text35,text36)
colnames(brooks_rating)[1] <- "text"
brooks_rating <- as.data.frame(brooks_rating)

id <- c(27,28,29,30,31,32,33,34,35,36)
id <- matrix(id, ncol=1)

brooks_rating <- cbind(id,brooks_rating)

tidy_brooks <- brooks_rating %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  count(word) %>%
  arrange(desc(n))

####SENTIMENT ANALYSIS

####BROOKS####
#Getting sentiment analysis for Brooks
brooks_afinn <- tidy_brooks %>%
  inner_join(get_sentiments("afinn"))%>%
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")

brooks_bing_and_nrc <- bind_rows(
  tidy_brooks%>%

```

```

    inner_join(get_sentiments("bing"))%>%
    mutate(method = "Bing et al."),
tidy_brooks %>%
    inner_join(get_sentiments("nrc")) %>%
        filter(sentiment %in% c("positive", "negative"))) %>%
    mutate(method = "NRC")) %>%
count(method, sentiment) %>%
spread(sentiment, n, fill=0) %>%
mutate(sentiment = positive-negative)

library(ggplot2)
bind_rows(brooks_afinn, brooks_bing_and_nrc) %>%
  ggplot(aes(method, sentiment, fill=method))+
  geom_col(show.legend=FALSE)+
  facet_wrap(~method, ncol =1, scales= "free_y")

#Most common positive and negative words for brooks
brooks_bing_counts <- tidy_brooks %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T) %>%
  ungroup()

brooks_bing_counts

brooks_bing_counts %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(word, n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y")+
  labs(y="Contribution to sentiment", x=NULL)+
  coord_flip()

####ASICS#####
#Getting sentiment analysis for Asics
asics_afinn <- tidy_asics %>%
  inner_join(get_sentiments("afinn"))%>%
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")

asics_bing_and_nrc <- bind_rows(
  tidy_asics%>%
    inner_join(get_sentiments("bing"))%>%
    mutate(method = "Bing et al."),

```

```

tidy_asics %>%
  inner_join(get_sentiments("nrc") %>%
    filter(sentiment %in% c("positive", "negative"))) %>%
  mutate(method = "NRC") %>%
  count(method, sentiment) %>%
  spread(sentiment, n, fill=0) %>%
  mutate(sentiment = positive-negative)

library(ggplot2)
bind_rows(asics_afinn, asics_bing_and_nrc) %>%
  ggplot(aes(method, sentiment, fill=method))+
  geom_col(show.legend=FALSE)+
  facet_wrap(~method, ncol=1, scales="free_y")

#Most common positive and negative words for asics
asics_bing_counts <- tidy_asics %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T) %>%
  ungroup()

asics_bing_counts

asics_bing_counts %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(word, n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y")+
  labs(y="Contribution to sentiment", x=NULL)+
  coord_flip()

####NIKE####
#Getting sentiment analysis for Nike
nike_afinn <- tidy_nike %>%
  inner_join(get_sentiments("afinn"))%>%
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")

nike_bing_and_nrc <- bind_rows(
  tidy_nike%>%
    inner_join(get_sentiments("bing"))%>%
    mutate(method = "Bing et al."),
  tidy_brooks %>%
    inner_join(get_sentiments("nrc") %>%

```

```

      filter(sentiment %in% c("positive", "negative")))) %>%
    mutate(method = "NRC")) %>%
  count(method, sentiment) %>%
  spread(sentiment, n, fill=0) %>%
  mutate(sentiment = positive-negative)

library(ggplot2)
bind_rows(nike_afinn, nike_bing_and_nrc) %>%
  ggplot(aes(method, sentiment, fill=method))+
  geom_col(show.legend=FALSE)+
  facet_wrap(~method, ncol=1, scales= "free_y")

#Most common positive and negative words for nike
nike_bing_counts <- tidy_nike %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T) %>%
  ungroup()

nike_bing_counts

nike_bing_counts %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(word, n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y")+
  labs(y="Contribution to sentiment", x=NULL)+
  coord_flip()

####MIZUNO####
#Getting sentiment analysis for mizuno
mizuno_afinn <- tidy_mizuno %>%
  inner_join(get_sentiments("afinn"))%>%
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")

mizuno_bing_and_nrc <- bind_rows(
  tidy_mizuno%>%
    inner_join(get_sentiments("bing"))%>%
    mutate(method = "Bing et al."),
  tidy_mizuno %>%
    inner_join(get_sentiments("nrc")) %>%
      filter(sentiment %in% c("positive", "negative")))) %>%
  mutate(method = "NRC")) %>%

```

```

count(method, sentiment) %>%
spread(sentiment, n, fill=0) %>%
mutate(sentiment = positive-negative)

library(ggplot2)
bind_rows(mizuno_afinn, mizuno_bing_and_nrc) %>%
  ggplot(aes(method, sentiment, fill=method))+
  geom_col(show.legend=FALSE)+
  facet_wrap(~method, ncol =1, scales= "free_y")

#Most common positive and negative words for mizuno
mizuno_bing_counts <- tidy_mizuno %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T) %>%
  ungroup()

mizuno_bing_counts

mizuno_bing_counts %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(word, n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y")+
  labs(y="Contribution to sentiment", x=NULL)+
  coord_flip()

##ZIPFs Law and TF-IDF###
###MIZUNO####
mizuno <- mizuno_rating %>%
  unnest_tokens(word, text) %>%
  count(id, word, sort=TRUE) %>%
  ungroup()

mizuno_total_words <- mizuno %>%
  group_by(id) %>%
  summarize(total=sum(n))

mizuno_words <- left_join(mizuno, mizuno_total_words)%>%
  filter(word %in% c("smooth", "secure", "strike", "jarring", "steep", "love", "stability"))

print(mizuno_words)

mizuno_freq_by_rank <- mizuno_words %>%

```

```

group_by(id) %>%
mutate(rank = row_number(),
       `term frequency` = n/total)
mizuno_freq_by_rank

mizuno_freq_by_rank %>%
  ggplot(aes(rank, `term frequency`, color=word))+
  geom_abline(intercept=-0.62, slope= -1.1, color='gray50', linetype=2)+
  geom_line(size= 1.1, alpha = 0.8, show.legend = FALSE)+
  scale_x_log10()+
  scale_y_log10()

#TF-IDF
mizuno_brand_words <- mizuno_words %>%
  bind_tf_idf(word, id,n)

mizuno_brand_words

mizuno_brand_words %>%
  arrange(desc(tf_idf))

mizuno_uniqueness <- mizuno_brand_words %>%
  arrange(desc(tf_idf))

#Graph for Term Frequency
mizuno_brand_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(id) %>%
  top_n(15) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=id))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~id, ncol=2, scales="free")+
  coord_flip()

##NIKE###
nike <- nike_rating %>%
  unnest_tokens(word, text) %>%
  count(id, word, sort=TRUE) %>%
  ungroup()

nike_total_words <- nike %>%
  group_by(id) %>%
  summarize(total=sum(n))

```

```

nike_words <- left_join(nike, nike_total_words)%>%
  filter(word %in% c("easy", "faster", "struggles", "soft", "price", "comfortable", "durable"))

print(nike_words)

nike_freq_by_rank <- nike_words %>%
  group_by(id) %>%
  mutate(rank = row_number(),
         `term frequency` = n/total)
nike_freq_by_rank

nike_freq_by_rank %>%
  ggplot(aes(rank, `term frequency`, color=word))+
  geom_abline(intercept=-0.62, slope= -1.1, color='gray50', linetype=2)+
  geom_line(size= 1.1, alpha = 0.8, show.legend = FALSE)+
  scale_x_log10()+
  scale_y_log10()

#TF-IDF
nike_brand_words <- nike_words %>%
  bind_tf_idf(word, id,n)

nike_brand_words

nike_brand_words %>%
  arrange(desc(tf_idf))

nike_uniqueness <- nike_brand_words %>%
  arrange(desc(tf_idf))

#Graph for Term Frequency
nike_brand_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(id) %>%
  top_n(15) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=id))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~id, ncol=2, scales="free")+
  coord_flip()

##Asics###
asics <- asics_rating %>%

```



```

unnest_tokens(word, text) %>%
count(id, word, sort=TRUE) %>%
ungroup()

asics_total_words <- asics %>%
  group_by(id) %>%
  summarize(total=sum(n))

asics_words <- left_join(asics, asics_total_words)%>%
  filter(word %in% c("durability", "dependable", "worth", "recommend", "stiff", "price",
"clunky"))

print(asics_words)

asics_freq_by_rank <- asics_words %>%
  group_by(id) %>%
  mutate(rank = row_number(),
         `term frequency` = n/total)
asics_freq_by_rank

asics_freq_by_rank %>%
  ggplot(aes(rank, `term frequency`, color=word))+
  geom_abline(intercept=-0.62, slope= -1.1, color='gray50', linetype=2)+
  geom_line(size= 1.1, alpha = 0.8, show.legend = FALSE)+
  scale_x_log10()+
  scale_y_log10()

#TF-IDF
asics_brand_words <- asics_words %>%
  bind_tf_idf(word, id,n)

asics_brand_words

asics_brand_words %>%
  arrange(desc(tf_idf))

asics_uniqueness <- asics_brand_words %>%
  arrange(desc(tf_idf))

#Graph for Term Frequency
asics_brand_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(id) %>%
  top_n(15) %>%
  ungroup %>%

```

```

ggplot(aes(word, tf_idf, fill=id))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~id, ncol=2, scales="free")+
  coord_flip()

###Brooks###
brooks <- brooks_rating %>%
  unnest_tokens(word, text) %>%
  count(id, word, sort=TRUE) %>%
  ungroup()

brooks_total_words <- brooks %>%
  group_by(id) %>%
  summarize(total=sum(n))

brooks_words <- left_join(brooks, brooks_total_words)%>%
  filter(word %in% c("comfortable", "stable", "worth", "perfect", "supportive", "rigid", "issue",
"break"))

print(brooks_words)

brooks_freq_by_rank <- brooks_words %>%
  group_by(id) %>%
  mutate(rank = row_number(),
  `term frequency` = n/total)
brooks_freq_by_rank

brooks_freq_by_rank %>%
  ggplot(aes(rank, `term frequency`, color=word))+
  geom_abline(intercept=-0.62, slope= -1.1, color='gray50', linetype=2)+
  geom_line(size= 1.1, alpha = 0.8, show.legend = FALSE)+
  scale_x_log10()+
  scale_y_log10()

#TF-IDF
brooks_brand_words <- brooks_words %>%
  bind_tf_idf(word, id,n)

brooks_brand_words

brooks_brand_words %>%
  arrange(desc(tf_idf))

brooks_uniqueness <- brooks_brand_words %>%
  arrange(desc(tf_idf))

```

```

#Graph for Term Frequency
brooks_brand_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(id) %>%
  top_n(15) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=id))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~id, ncol=2, scales="free")+
  coord_flip()
##Bigrams##
###MIZUNO Bigrams
mizuno_bigram <- mizuno_rating %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)

mizuno_bigram

mizuno_bigram%>%
  count(bigram, sort = TRUE)

library(tidyr)
mizuno_bigrams_separated <- mizuno_bigram%>%
  separate(bigram, c("word1", "word2"), sep = " ")

mizuno_bigrams_filtered <- mizuno_bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

mizuno_bigram_counts <- mizuno_bigrams_filtered %>%
  count(word1, word2, sort = TRUE)

mizuno_bigram_counts

###BROOKS

brooks_bigram <- brooks_rating %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)

brooks_bigram

brooks_bigram%>%
  count(bigram, sort = TRUE)

```

```
library(tidyr)
brooks_bigrams_separated <- brooks_bigram%>%
  separate(bigram, c("word1", "word2"), sep = " ")

brooks_bigrams_filtered <- brooks_bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

#creating the new bigram, "no-stop-words":
brooks_bigram_counts <- brooks_bigrams_filtered %>%
  count(word1, word2, sort = TRUE)
```

```
brooks_bigram_counts
```

```
###ASICS
asics_bigram <- asics_rating %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)
```

```
asics_bigram
```

```
asics_bigram%>%
  count(bigram, sort = TRUE)
```

```
library(tidyr)
```

```
asics_bigrams_separated <- asics_bigram%>%
  separate(bigram, c("word1", "word2"), sep = " ")

asics_bigrams_filtered <- asics_bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

asics_bigram_counts <- asics_bigrams_filtered %>%
  count(word1, word2, sort = TRUE)
```

```
asics_bigram_counts
```

```
###NIKE
```

```
nike_bigram <- nike_rating %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)
```

```
nike_bigram
```

```
nike_bigram%>%
  count(bigram, sort = TRUE)
```

```
library(tidyr)
nike_bigrams_separated <- nike_bigram%>%
  separate(bigram, c("word1", "word2"), sep = " ")

nike_bigrams_filtered <- nike_bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

nike_bigram_counts <- nike_bigrams_filtered %>%
  count(word1, word2, sort = TRUE)

nike_bigram_counts

###CHARTS FOR BIGRAMS
##MIZUNO
library(igraph)
mizuno_bigram_graph <- mizuno_bigram_counts %>%
  filter(n>0) %>%
  graph_from_data_frame()

mizuno_bigram_graph

#install.packages("ggraph")
library(ggraph)
ggraph(mizuno_bigram_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)

###Brooks
library(igraph)
brooks_bigram_graph <- brooks_bigram_counts %>%
  filter(n>0) %>%
  graph_from_data_frame()

brooks_bigram_graph

library(ggraph)
ggraph(brooks_bigram_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)

##Asics
library(igraph)
```

```
asics_bigram_graph <- asics_bigram_counts %>%
  filter(n>0) %>%
  graph_from_data_frame()
```

```
asics_bigram_graph
```

```
library(gggraph)
gggraph(asics_bigram_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust=1, hjust=1)
```

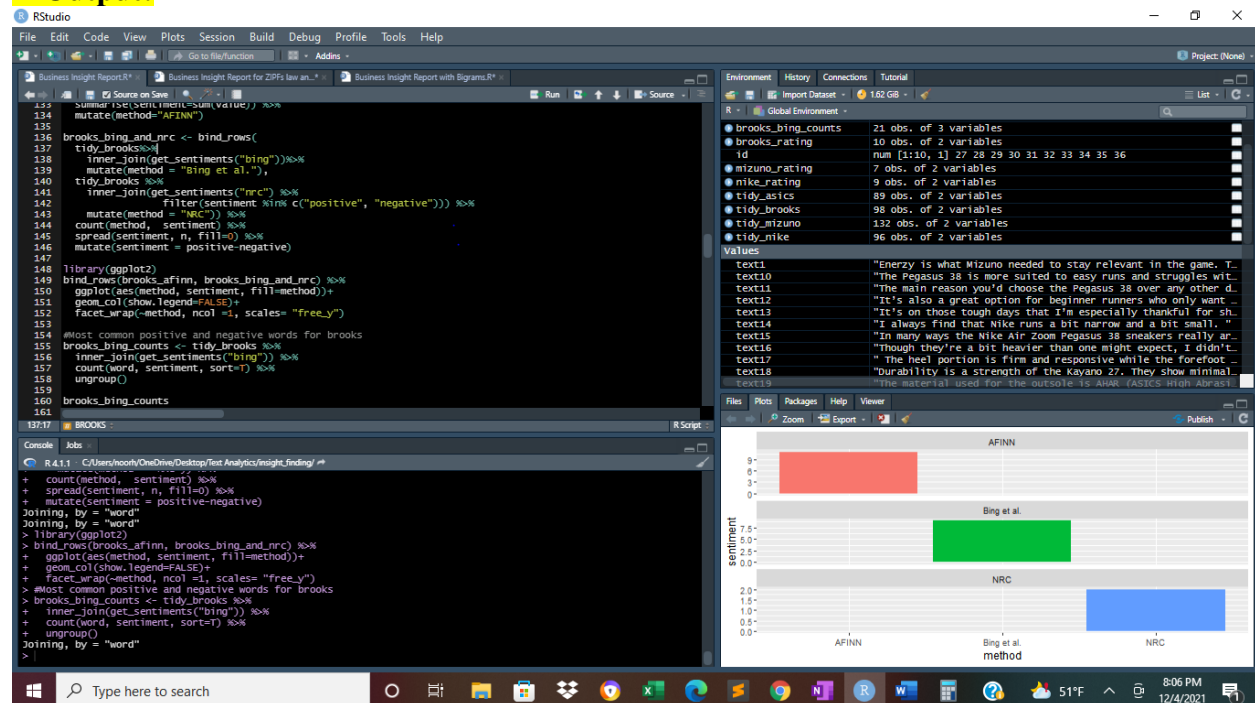
```
##Nike
```

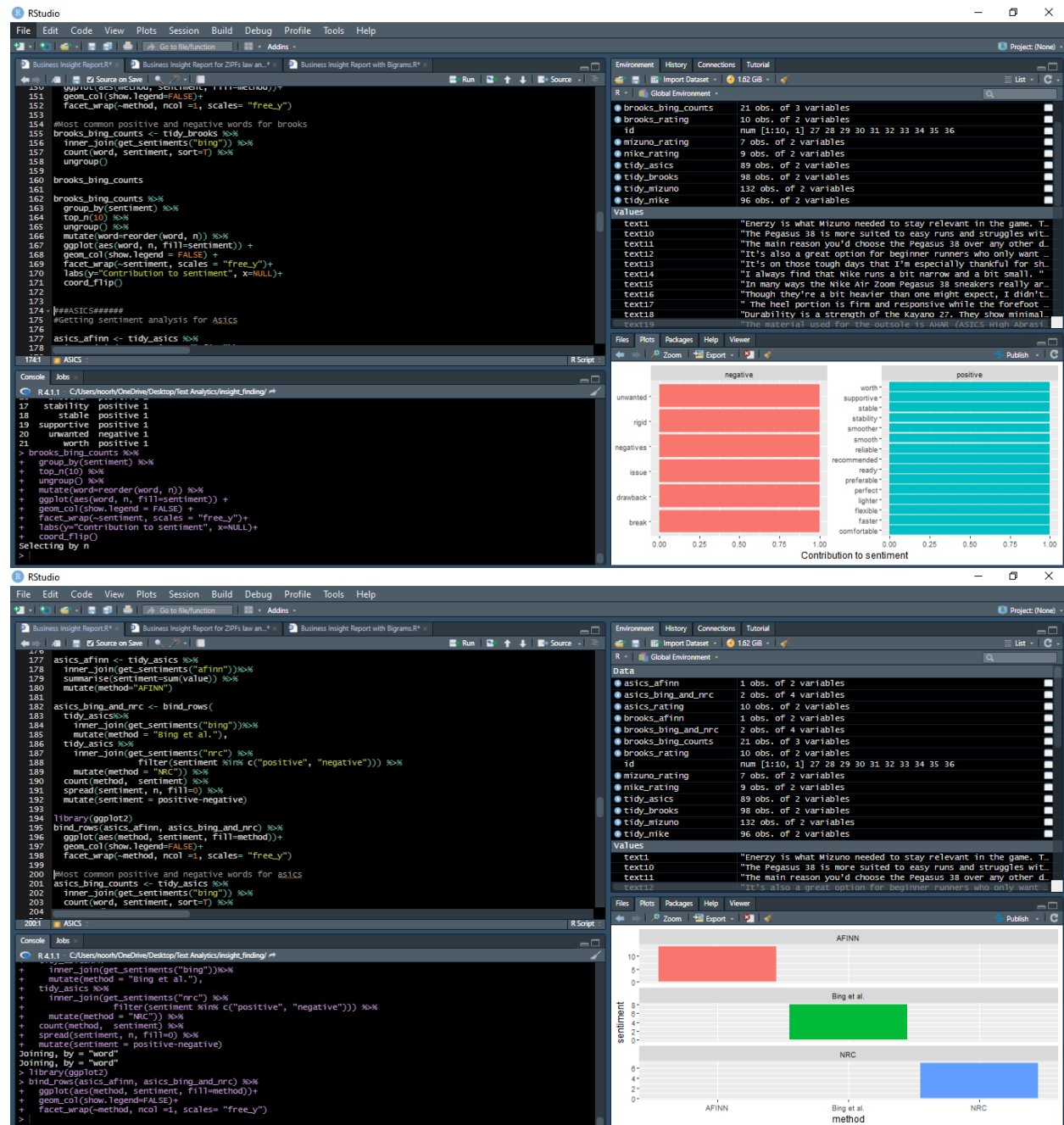
```
library(igraph)
nike_bigram_graph <- nike_bigram_counts %>%
  filter(n>0) %>%
  graph_from_data_frame()
```

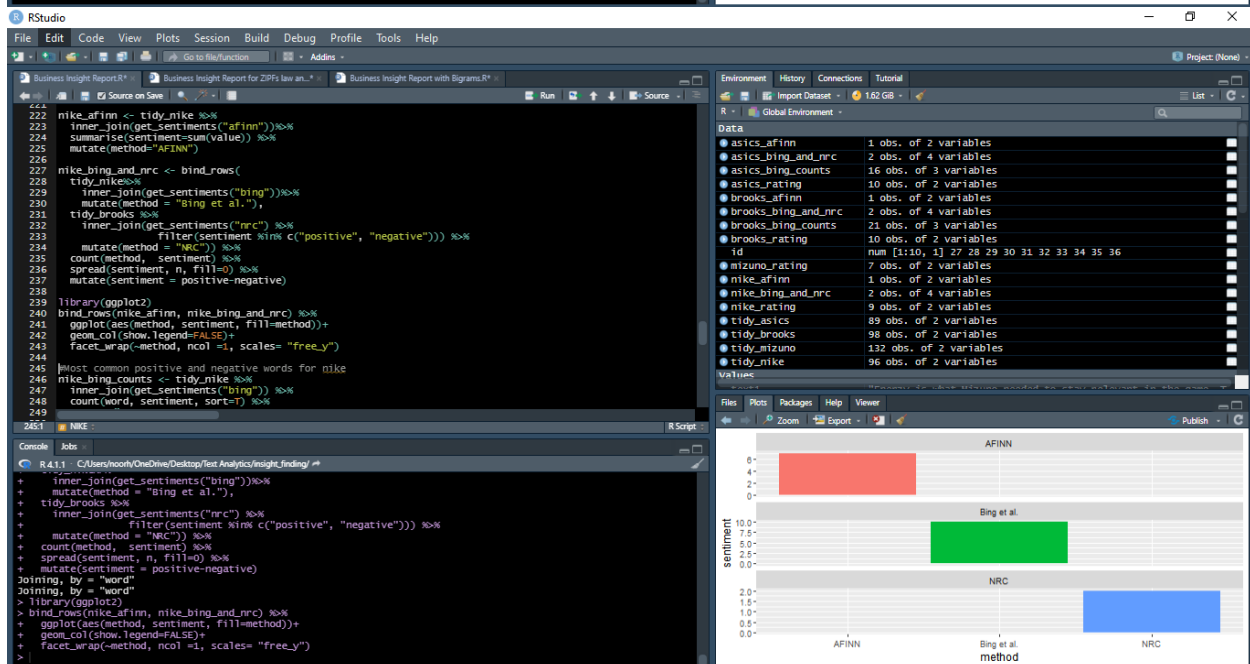
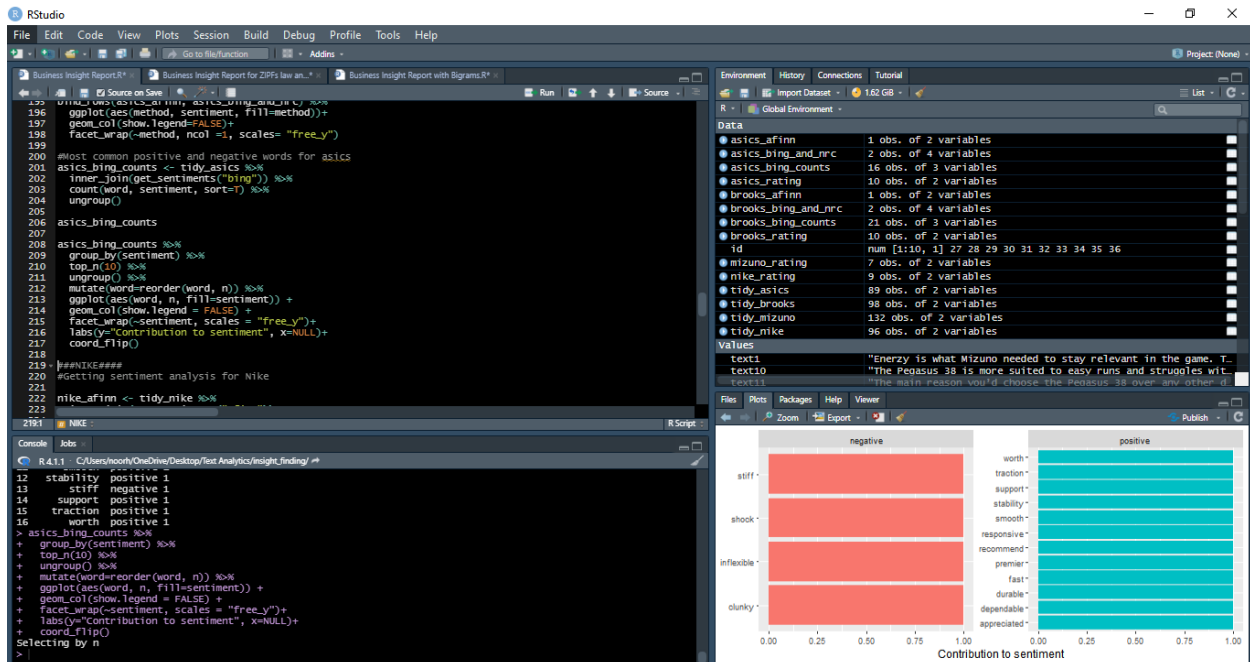
```
nike_bigram_graph
```

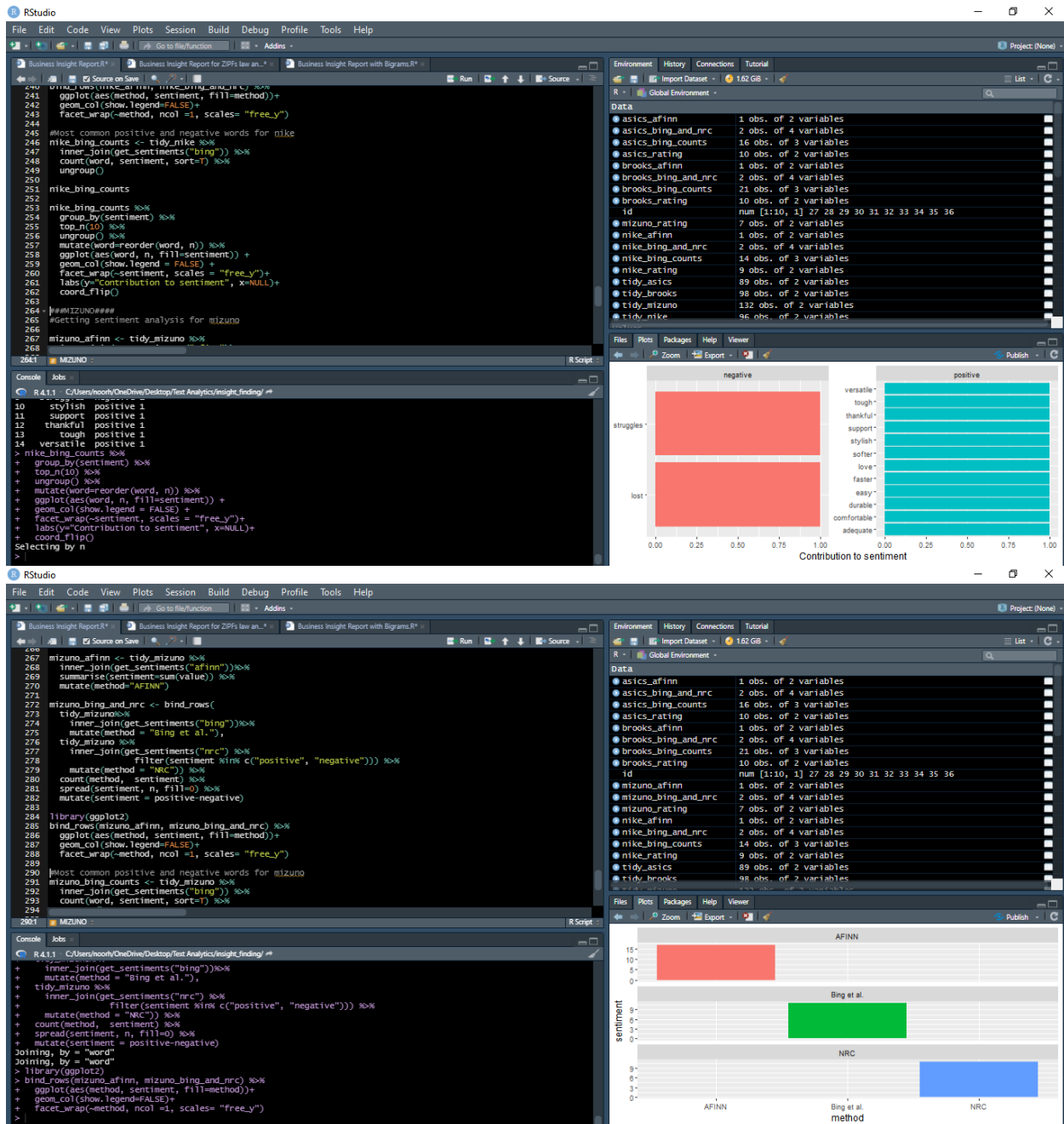
```
library(gggraph)
gggraph(nike_bigram_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust=1, hjust=1)
```

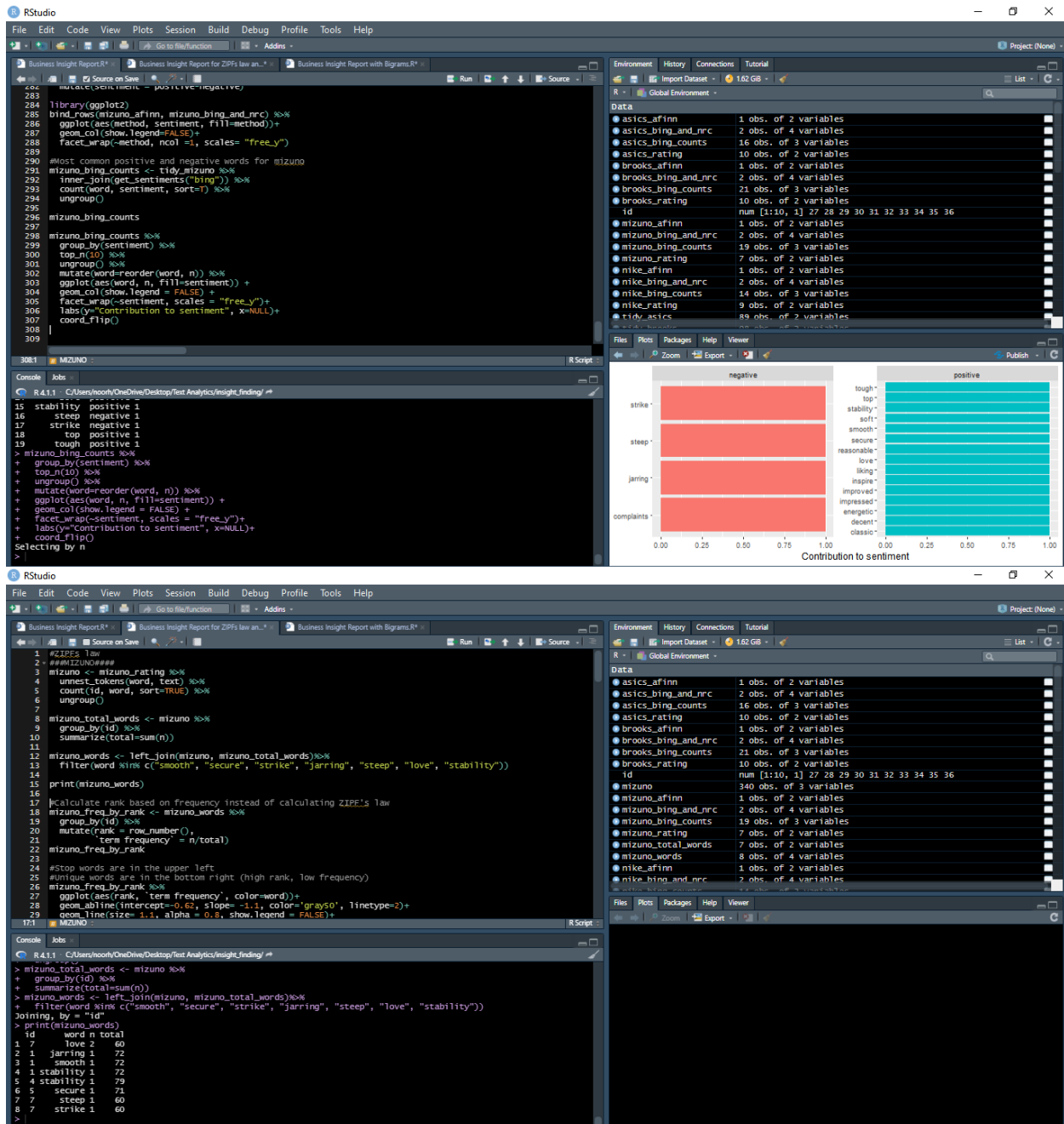
R Output:











RStudio

```

1 #ZIPF's law
2 ##MIZUNO###
3 mizuno <- mizuno_rating %>%
4   unnest_tokens(word, text) %>%
5   count(id, word, sort=TRUE) %>%
6   ungroup()
7
8 mizuno_total_words <- mizuno %>%
9   group_by(id) %>%
10  summarize(total=sum(n))
11
12 mizuno_words <- left_join(mizuno, mizuno_total_words)%>%
13   filter(word %in% c("smooth", "secure", "strike", "jarring", "steep", "love", "stability"))
14
15 print(mizuno_words)
16
17 #calculate rank based on frequency instead of calculating ZIPF's law
18 mizuno_freq_by_rank <- mizuno_words %>%
19   group_by(id) %>%
20   mutate(rank = row_number(),
21          'term frequency' = n/total)
22 mizuno_freq_by_rank
23
24 #stop words are in the upper left
25 #unique words are in the bottom right (high rank, low frequency)
26 mizuno_freq_by_rank %>%
27   ggplot(aes(rank, 'term frequency', color=word))+
28   geom_abline(intercept=-0.62, slope=-1.1, color='gray50', linetype=2)+
29   geom_line(size=1.1, alpha=0.8, show.legend=FALSE)+

```

Console

```

R411 C:\Users\hoofy\OneDrive\Desktop\Text Analytics\insight_finding/ #
> group_by(id) %>%
+ mutate(rank = row_number(),
+        'term frequency' = n/total)
> mizuno_freq_by_rank
# A tibble: 8 x 6
# Groups:   id [4]
  id word      n total rank 'term frequency'
<dbl> <chr> <int> <int> <int> <dbl>
1 2 love      2 60 1 0.0333
2 1 jarring    1 72 2 0.0139
3 1 smooth    1 72 2 0.0139
4 1 stability 1 72 1 0.0139
5 4 stability 1 79 1 0.0127
6 5 secure    1 71 1 0.0141
7 7 steep     1 60 2 0.0167
8 7 strike    1 60 3 0.0167

```

Environment

Variable	Size
asics_afinn	1 obs. of 2 variables
asics_bing_and_nrc	2 obs. of 4 variables
asics_bing_counts	16 obs. of 3 variables
asics_rating	10 obs. of 2 variables
brooks_afinn	1 obs. of 2 variables
brooks_bing_and_nrc	2 obs. of 4 variables
brooks_bing_counts	21 obs. of 3 variables
brooks_rating	10 obs. of 2 variables
id	num [1:10, 1] 27 28 29 30 31 32 33 34 35 36
mizuno	340 obs. of 3 variables
mizuno_afinn	1 obs. of 2 variables
mizuno_bing_and_nrc	2 obs. of 4 variables
mizuno_bing_counts	19 obs. of 3 variables
mizuno_freq_by_rank	8 obs. of 6 variables
mizuno_rating	7 obs. of 2 variables
mizuno_total_words	7 obs. of 2 variables
mizuno_words	8 obs. of 4 variables
nike_afinn	1 obs. of 2 variables

RStudio

```

14 print(mizuno_words)
15
16 #calculate rank based on frequency instead of calculating ZIPF's law
17 mizuno_freq_by_rank <- mizuno_words %>%
18   group_by(id) %>%
19   mutate(rank = row_number(),
20          'term frequency' = n/total)
21 mizuno_freq_by_rank
22
23 #stop words are in the upper left
24 #unique words are in the bottom right (high rank, low frequency)
25 mizuno_freq_by_rank %>%
26   ggplot(aes(rank, 'term frequency', color=word))+
27   geom_abline(intercept=-0.62, slope=-1.1, color='gray50', linetype=2)+
28   geom_line(size=1.1, alpha=0.8, show.legend=FALSE)+
29   scale_x_log10()+
30   scale_y_log10()
31
32 #TF-IDF
33 mizuno_brand_words <- mizuno_words %>%
34   bind_tf_idf(word, id, n)
35
36 mizuno_brand_words
37
38 mizuno_brand_words %>%
39   arrange(desc(tf_idf))
40
41 mizuno_uniqueness <- mizuno_brand_words %>%

```

Console

```

R411 C:\Users\hoofy\OneDrive\Desktop\Text Analytics\insight_finding/ #
+ geom_line(size=1.1, alpha=0.8, show.legend=FALSE)+
+ scale_x_log10()+
+ scale_y_log10()
> #TF-IDF
> mizuno_brand_words <- mizuno_words %>%
+ bind_tf_idf(word, id, n)
> mizuno_brand_words
# A tibble: 8 x 5
  id word      n total tf_idf
<dbl> <chr> <int> <int> <dbl>
1 2 love      2 60 0.5000000 1.3862944
2 1 jarring    1 72 0.3333333 1.3862944
3 1 smooth    1 72 0.3333333 1.3862944
4 1 stability 1 72 0.3333333 0.6931472
5 4 stability 1 79 1.0000000 0.6931472
6 5 secure    1 71 1.0000000 1.3862944
7 7 steep     1 60 0.2500000 1.3862944
8 7 strike    1 60 0.2500000 1.3862944

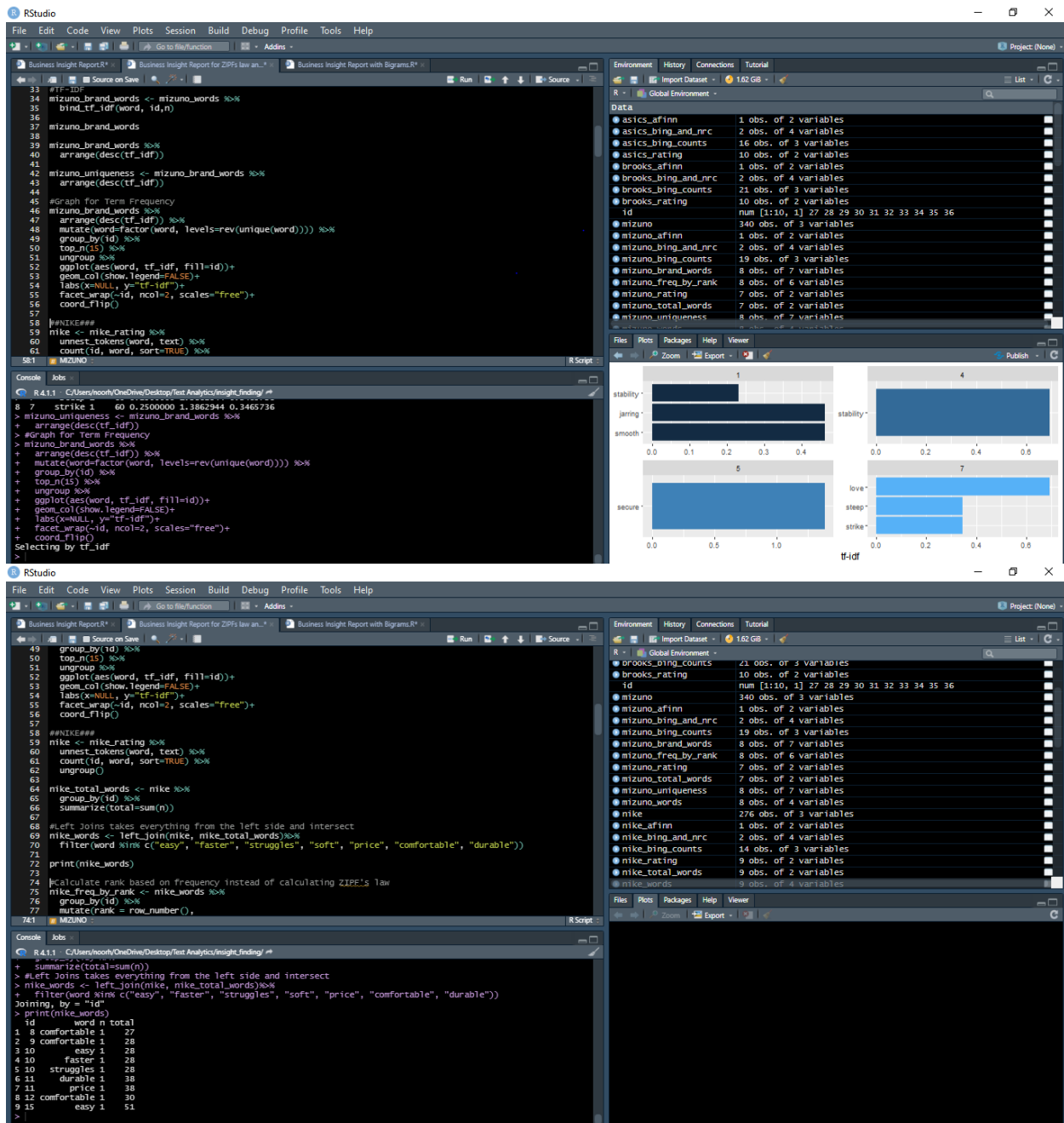
```

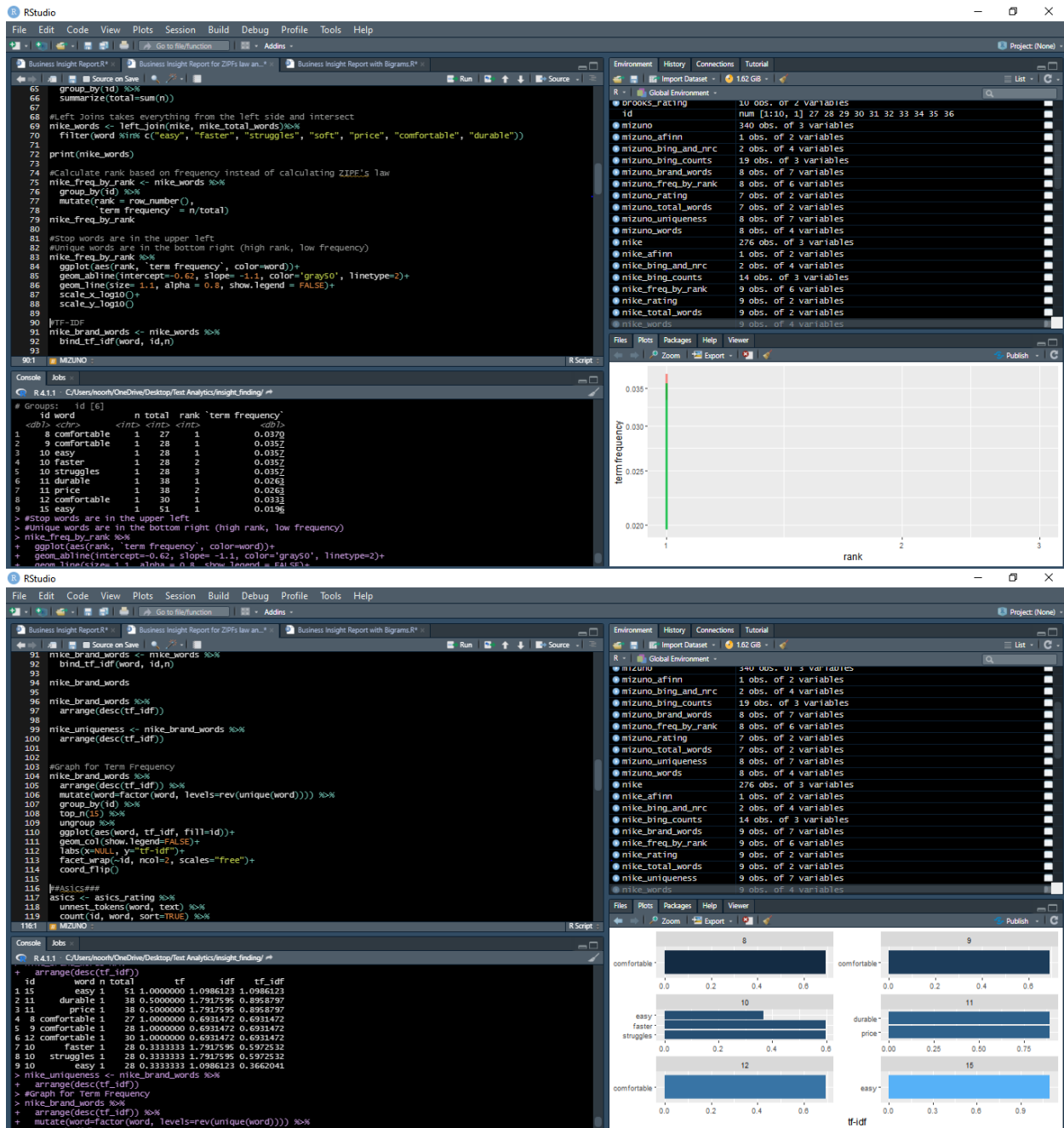
Environment

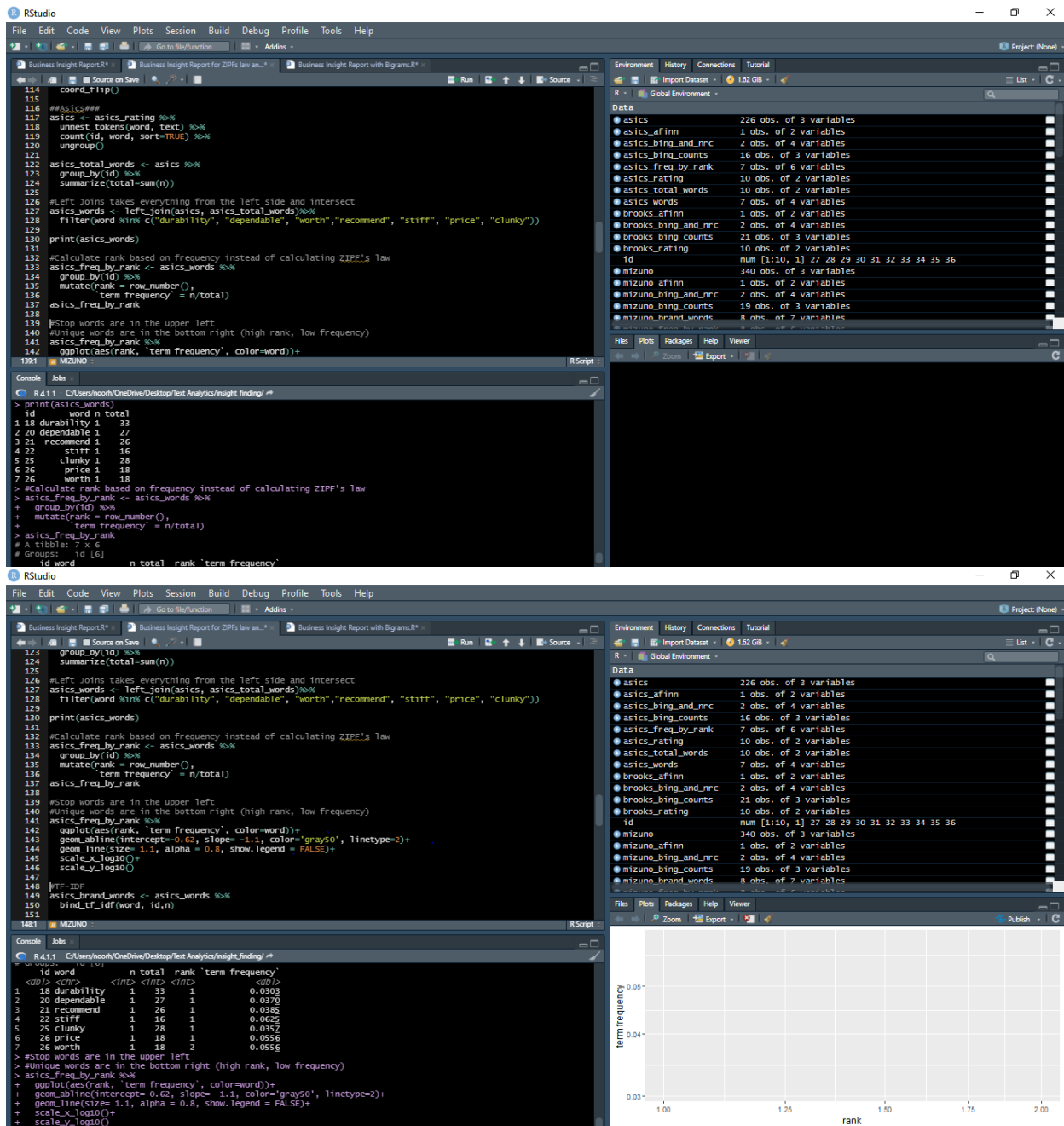
Variable	Size
asics_afinn	1 obs. of 2 variables
asics_bing_and_nrc	2 obs. of 4 variables
asics_bing_counts	16 obs. of 3 variables
asics_rating	10 obs. of 2 variables
brooks_afinn	1 obs. of 2 variables
brooks_bing_and_nrc	2 obs. of 4 variables
brooks_bing_counts	21 obs. of 3 variables
brooks_rating	10 obs. of 2 variables
id	num [1:10, 1] 27 28 29 30 31 32 33 34 35 36
mizuno	340 obs. of 3 variables
mizuno_afinn	1 obs. of 2 variables
mizuno_bing_and_nrc	2 obs. of 4 variables
mizuno_bing_counts	19 obs. of 3 variables
mizuno_brand_words	8 obs. of 7 variables
mizuno_freq_by_rank	8 obs. of 6 variables
mizuno_rating	7 obs. of 2 variables
mizuno_total_words	7 obs. of 2 variables
mizuno_uniqueness	8 obs. of 7 variables

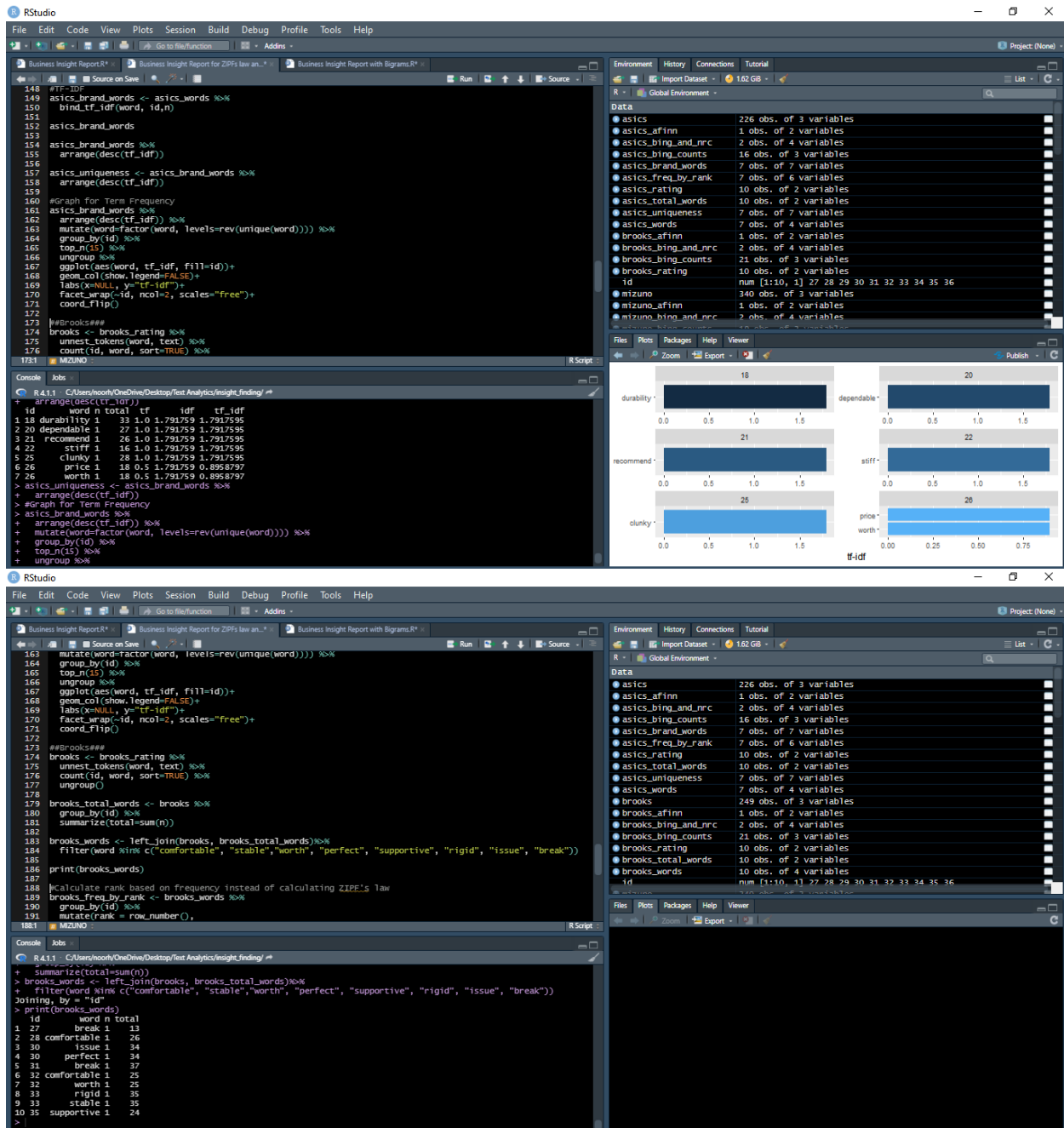
term frequency

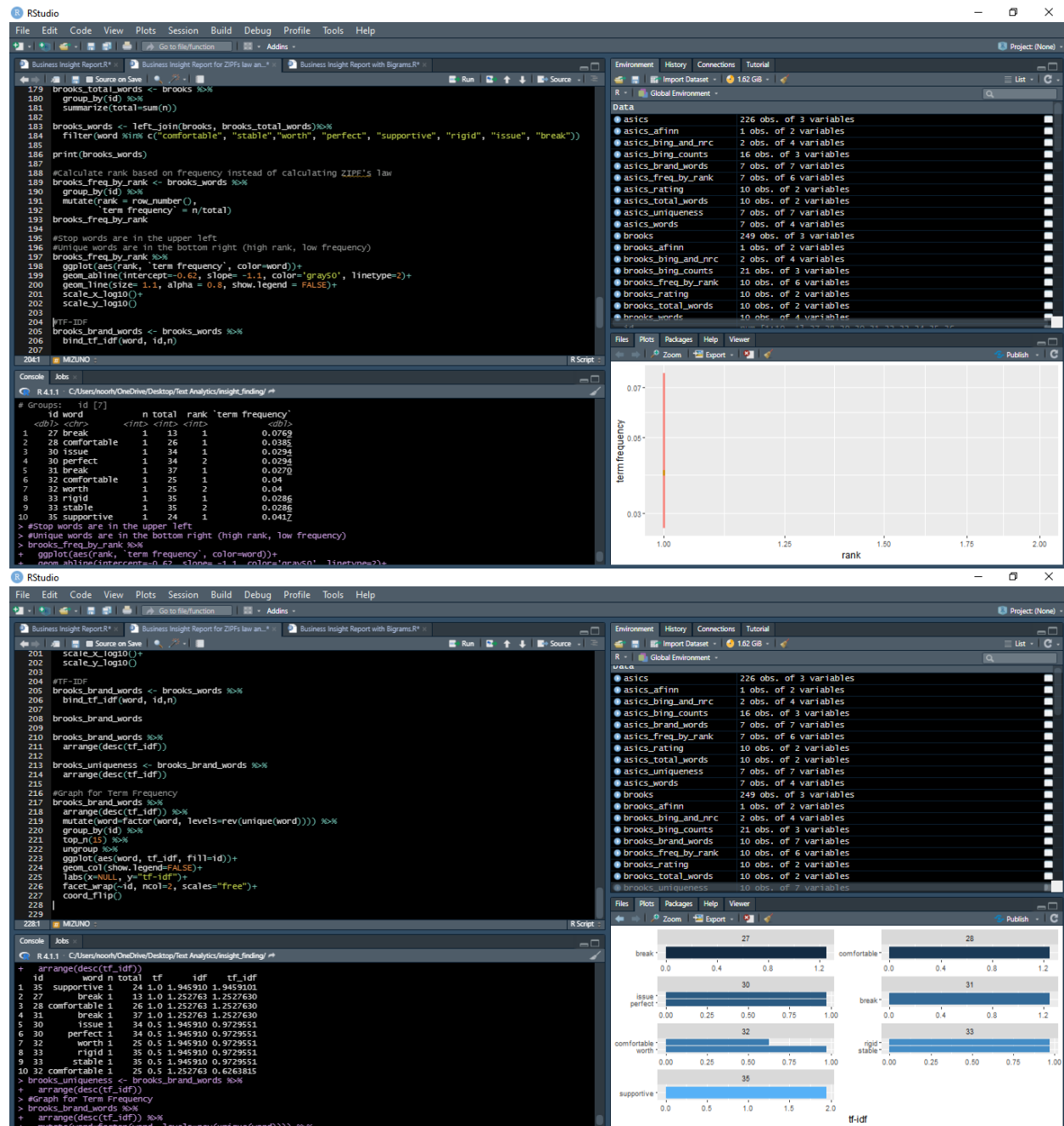
rank











Script Editor:

```

1 ##MIZUNO Bigrams
2 mizuno_bigram <- mizuno_rating %>%
3   unnest_tokens(bigram, text, token = "ngrams", n=2)
4
5 mizuno_bigram ##we want to see the bigrams (words that appear together, "pairs")
6
7 mizuno_bigram%>%
8   count(bigram, sort = TRUE)
9
10 #to remove stop words from the bigram data
11 library(tidyverse)
12 #separating the bigram and tokenizing
13 mizuno_bigrams_separated <- mizuno_bigram%>%
14   separate(bigram, c("word1", "word2"), sep = " ")
15
16 mizuno_bigrams_filtered <- mizuno_bigrams_separated %>%
17   filter(word1 %in% stop_words$word) %>%
18   filter(word2 %in% stop_words$word)
19
20 #creating the new bigram, "no-stop-words":
21 mizuno_bigram_counts <- mizuno_bigrams_filtered %>%
22   count(word1, word2, sort = TRUE)
23
24 mizuno_bigram_counts
25
26 ##BROOKS
27
28 brooks_bigram <- brooks_rating %>%
29   unnest_tokens(bigram, text, token = "ngrams", n=2)

```

Console:

```

R41.1 C:\Users\hoofh\OneDrive\Desktop\Text Analytics\insight_finding / #
text7...415 7 for my
text7...416 7 my liking
text7...417 7 liking if
text7...418 7 if you
text7...419 7 you can
text7...420 7 can teach
text7...421 7 teach me
text7...422 7 me to
text7...423 7 to love
text7...424 7 love the
text7...425 7 the extra
text7...426 7 extra drop
text7...427 7 drop height
text7...428 7 height i'm
text7...429 7 i'm all
text7...430 7 all ears

```

Environment:

Object	Class	Attributes
asics_brand_words	data.frame	7 obs. of 7 variables
asics_freq_by_rank	data.frame	7 obs. of 6 variables
asics_rating	data.frame	10 obs. of 2 variables
asics_total_words	data.frame	10 obs. of 2 variables
asics_uniqueness	data.frame	7 obs. of 7 variables
asics_words	data.frame	7 obs. of 4 variables
brooks	data.frame	249 obs. of 3 variables
brooks_afinn	data.frame	1 obs. of 2 variables
brooks_bing_counts	data.frame	2 obs. of 4 variables
brooks_bing_and_nrc	data.frame	21 obs. of 3 variables
brooks_brand_words	data.frame	10 obs. of 7 variables
brooks_freq_by_rank	data.frame	10 obs. of 6 variables
brooks_rating	data.frame	10 obs. of 2 variables
brooks_total_words	data.frame	10 obs. of 2 variables
brooks_uniqueness	data.frame	10 obs. of 7 variables
brooks_words	data.frame	10 obs. of 4 variables
id	data.frame	num [1:10, 1] 27 28 29 30 31 32 33 34 35 36
mizuno	data.frame	340 obs. of 3 variables
mizuno_afinn	data.frame	1 obs. of 2 variables
mizuno_bigram	data.frame	430 obs. of 2 variables

Script Editor:

```

1 ##MIZUNO Bigrams
2 mizuno_bigram <- mizuno_rating %>%
3   unnest_tokens(bigram, text, token = "ngrams", n=2)
4
5 mizuno_bigram ##we want to see the bigrams (words that appear together, "pairs")
6
7 mizuno_bigram%>%
8   count(bigram, sort = TRUE)
9
10 #to remove stop words from the bigram data
11 library(tidyverse)
12 #separating the bigram and tokenizing
13 mizuno_bigrams_separated <- mizuno_bigram%>%
14   separate(bigram, c("word1", "word2"), sep = " ")
15
16 mizuno_bigrams_filtered <- mizuno_bigrams_separated %>%
17   filter(word1 %in% stop_words$word) %>%
18   filter(word2 %in% stop_words$word)
19
20 #creating the new bigram, "no-stop-words":
21 mizuno_bigram_counts <- mizuno_bigrams_filtered %>%
22   count(word1, word2, sort = TRUE)
23
24 mizuno_bigram_counts
25
26 ##BROOKS
27
28 brooks_bigram <- brooks_rating %>%
29   unnest_tokens(bigram, text, token = "ngrams", n=2)

```

Console:

```

R41.1 C:\Users\hoofh\OneDrive\Desktop\Text Analytics\insight_finding / #
40 plate isn't 1
41 previous versions 1
42 reasonable surfaces 1
43 recycled mesh 1
44 rubber outsole 1
45 secure fit 1
46 shoe feels 1
47 smooth speaking 1
48 softness combined 1
49 stability shoe 1
50 stay relevant 1
51 stride smooth 1
52 subtle version 1
53 thomas appears 1
54 total sync 1
55 xio rubber 1

```

Environment:

Object	Class	Attributes
asics_brand_words	data.frame	10 obs. of 7 variables
asics_freq_by_rank	data.frame	7 obs. of 6 variables
asics_rating	data.frame	7 obs. of 4 variables
asics_total_words	data.frame	249 obs. of 3 variables
asics_uniqueness	data.frame	1 obs. of 2 variables
asics_words	data.frame	2 obs. of 4 variables
brooks	data.frame	21 obs. of 3 variables
brooks_afinn	data.frame	10 obs. of 7 variables
brooks_bing_counts	data.frame	10 obs. of 2 variables
brooks_bing_and_nrc	data.frame	10 obs. of 6 variables
brooks_brand_words	data.frame	10 obs. of 2 variables
brooks_freq_by_rank	data.frame	10 obs. of 7 variables
brooks_rating	data.frame	10 obs. of 4 variables
brooks_total_words	data.frame	10 obs. of 7 variables
brooks_uniqueness	data.frame	10 obs. of 4 variables
brooks_words	data.frame	num [1:10, 1] 27 28 29 30 31 32 33 34 35 36
id	data.frame	340 obs. of 3 variables
mizuno	data.frame	1 obs. of 2 variables
mizuno_afinn	data.frame	430 obs. of 2 variables
mizuno_bigram	data.frame	55 obs. of 3 variables
mizuno_bigram_counts	data.frame	67 obs. of 3 variables
mizuno_bigram_filter	data.frame	430 obs. of 3 variables

The image displays two screenshots of the RStudio interface, showing the process of analyzing shoe reviews using the tidytext package.

Top Screenshot:

- Script Editor:** Contains R code for processing the 'mizuno' dataset. It includes steps for removing stop words, separating bigrams, and creating a new bigram dataset.
- Environment:** Lists the objects in the global environment, including 'asics_uniqueness', 'asics_words', 'brooks', 'brooks_afinn', 'brooks_bigram', 'brooks_bing_and_nrc', 'brooks_bing_counts', 'brooks_brand_words', 'brooks_freq_by_rank', 'brooks_rating', 'brooks_total_words', 'brooks_uniqueness', 'brooks_words', 'id', 'mizuno', 'mizuno_afinn', 'mizuno_bigram', 'mizuno_bigram_counts', 'mizuno_bigram_filter', and 'mizuno_bigrams_separa'.
- Console:** Shows the output of the code, displaying the first few rows of the 'mizuno' dataset, including columns for 'id', 'text', and 'brand'.

Bottom Screenshot:

- Script Editor:** Continues the R code, showing the final steps of the analysis, including the creation of the 'mizuno' dataset and the calculation of sentiment scores.
- Environment:** Lists the objects in the global environment, including 'asics_uniqueness', 'asics_words', 'brooks', 'brooks_afinn', 'brooks_bigram', 'brooks_bing_and_nrc', 'brooks_bing_counts', 'brooks_brand_words', 'brooks_freq_by_rank', 'brooks_rating', 'brooks_total_words', 'brooks_uniqueness', 'brooks_words', 'id', 'mizuno', 'mizuno_afinn', 'mizuno_bigram', 'mizuno_bigram_counts', 'mizuno_bigram_filter', and 'mizuno_bigrams_separa'.
- Console:** Shows the output of the code, displaying the first few rows of the 'mizuno' dataset, including columns for 'id', 'text', and 'brand'.

The image displays two screenshots of the RStudio interface, showing the process of analyzing shoe reviews using bigrams.

Top Screenshot:

- Code Editor:** Contains R code for processing Brooks bigram data. The code includes:


```
library(tidyverse)
# separating the bigram and tokenizing
brooks_bigrams_separated <- brooks_bigram %>%
  separate(bigram, c("word1", "word2"), sep = " ")
# filtering out stop words
brooks_bigrams_filtered <- brooks_bigrams_separated %>%
  filter(word1 %in% stop_words$word) %>%
  filter(word2 %in% stop_words$word)
# creating the new bigram, "no-stop-words":
brooks_bigram_counts <- brooks_bigrams_filtered %>%
  count(word1, word2, sort = TRUE)
brooks_bigram_counts
```
- Console:** Shows the output of the code, displaying a list of bigrams and their counts, such as "up 1 1", "up shock 1", "up the 1", etc.
- Environment:** Lists the objects created in the global environment, including:
 - asics_bigram: 244 obs. of 2 variables
 - asics_bing_and_nrc: 2 obs. of 4 variables
 - asics_bing_counts: 16 obs. of 3 variables
 - asics_brand_words: 7 obs. of 7 variables
 - asics_freq_by_rank: 7 obs. of 6 variables
 - asics_rating: 10 obs. of 2 variables
 - asics_total_words: 10 obs. of 2 variables
 - asics_uniqueness: 7 obs. of 4 variables
 - asics_words: 7 obs. of 4 variables
 - brooks: 249 obs. of 3 variables
 - brooks_afinn: 1 obs. of 2 variables
 - brooks_bigram: 262 obs. of 2 variables
 - brooks_bigram_counts: 41 obs. of 3 variables
 - brooks_bigrams_filtered: 41 obs. of 3 variables
 - brooks_bigrams_separated: 262 obs. of 3 variables
 - brooks_bing_and_nrc: 2 obs. of 4 variables
 - brooks_bing_counts: 21 obs. of 3 variables
 - brooks_brand_words: 10 obs. of 7 variables
 - brooks_freq_by_rank: 10 obs. of 6 variables

Bottom Screenshot:

- Code Editor:** Contains R code for processing Asics bigram data. The code includes:


```
##ASICS
asics_bigram <- asics_rating %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)
asics_bigram
asics_bigram %>%
  count(bigram, sort = TRUE)
library(tidyverse)
asics_bigrams_separated <- asics_bigram %>%
  separate(bigram, c("word1", "word2"), sep = " ")
# filtering out stop words
asics_bigrams_filtered <- asics_bigrams_separated %>%
  filter(word1 %in% stop_words$word) %>%
  filter(word2 %in% stop_words$word)
# creating the new bigram, "no-stop-words":
asics_bigram_counts <- asics_bigrams_filtered %>%
  count(word1, word2, sort = TRUE)
asics_bigram_counts
```
- Console:** Shows the output of the code, displaying a list of bigrams and their counts, such as "hundred miles 1", "inflexible stiff 1", "it's heavy 1", etc.
- Environment:** Lists the objects created in the global environment, including:
 - asics_bigram_counts: 35 obs. of 3 variables
 - asics_bigrams_filtered: 40 obs. of 3 variables
 - asics_bigrams_separated: 244 obs. of 3 variables
 - asics_bing_and_nrc: 2 obs. of 4 variables
 - asics_bing_counts: 16 obs. of 3 variables
 - asics_brand_words: 7 obs. of 7 variables
 - asics_freq_by_rank: 7 obs. of 6 variables
 - asics_rating: 10 obs. of 2 variables
 - asics_total_words: 10 obs. of 2 variables
 - asics_uniqueness: 7 obs. of 7 variables
 - asics_words: 7 obs. of 4 variables
 - brooks: 249 obs. of 3 variables
 - brooks_afinn: 1 obs. of 2 variables
 - brooks_bigram: 262 obs. of 2 variables
 - brooks_bigram_counts: 41 obs. of 3 variables
 - brooks_bigrams_filtered: 41 obs. of 3 variables
 - brooks_bigrams_separated: 262 obs. of 3 variables
 - brooks_bing_and_nrc: 2 obs. of 4 variables
 - brooks_bing_counts: 21 obs. of 3 variables

The image displays two screenshots of the RStudio interface, showing the process of creating a bigram from text data.

Top Screenshot:

- Source Editor:** Contains R code for creating a bigram from text data. The code includes filtering stop words, creating a bigram, and counting the bigram counts.
- Environment:** Lists the objects in the global environment, including `brooks_uniqueness`, `brooks_words`, `id`, `mizuno`, `mizuno_afinn`, `mizuno_bigram`, `mizuno_bigram_counts`, `mizuno_bigrams_filter`, `mizuno_bigrams_separa`, `mizuno_bing_and_nrc`, `mizuno_bing_counts`, `mizuno_brand_words`, `mizuno_freq_by_rank`, `mizuno_rating`, `mizuno_total_words`, `mizuno_uniqueness`, `mizuno_words`, `nike`, `nike_afinn`, and `nike_bigram`.
- Console:** Shows the output of the code, displaying the bigram counts for the text "an adequate amount of support so that my knees and hips felt good from start to finish".

Bottom Screenshot:

- Source Editor:** The code is updated to include a filter for "no-stop-words".
- Environment:** The environment remains the same as in the top screenshot.
- Console:** Shows the output of the code, displaying the bigram counts for the text "ways the 1 wear while 1 well built 1 when i 1 which was 1 while running 1 who only 1 will last 1 with an 1 with any 1 with faster 1 with plenty 1 workhorse has 1 you'd choose 1".

