# Cloud Business Insight Report from NLP algorithms

Prepared By: Lili Li

Global MBAN, Hult International Business School

Text Analytics and Natural Language Processing (NLP) - DAT-5317 - FMBAN2

Professor: Thomas Kurnicki

December 05, 2021

# Contents

# Cloud Business Insight Report from NLP algorithms

The three biggest cloud business players are AWS from Amazon, Azure from Microsoft, and GCP from Google. This article extracts some common and unique features related to the cloud computing business for these three players. The annual report reflects how the company self-evaluates its whole business, and cloud tweets show how the public sees these players' cloud services.

In annual reports, the cloud business is mentioned as a part of the whole corporate strategy. And the frequency of cloud keywords can show how important cloud business is in their current development stage and its role in long-term strategy. Some keywords mentioned by each company could be some differentiators for developing cloud business in the mid-term or long-term.

In cloud tweets related to these three companies, we saw more insights about which areas individuals or general people mention about the clouding players, which helps grab more depth analysis.

Our recommendation for investing in cloud business from public listed companies is Microsoft. Microsoft cloud brand, Azure, is a strong brand image for the general public. Current business areas range from government to operation, from public to private. Customer diversity is the highest among major cloud players. "Microsoft Azure benefits from its software-as-a-service footprint, most of the revenue is derived from Office 365, Dynamics, and a bevy of other cloud services that are software-based over infrastructure" (https://www.zdnet.com/article/the-top-cloud-providers-of-2021-aws-microsoft-azure-google-cloud-hybrid-saas/). Amazon AWS is also a strong brand name and enjoys the leading position in cloud business now, but it more benefits from eCommerce business, which is not diverse enough compared with Microsoft Azure. Currently, Google cloud GCP is not competitive compared to AWS and Azure.

Summary Table from Annual Report

| Companies | Cloud business | Features | Main comments | Diversity of word | Unique business units | General |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| Google | GCP | | Network | More | Youtube + ad + maps | Being innovative and sustainable development for community and planet |
| Microsoft | Azure | Edge computing | Dynamic software, Server | More | Windows + azure + github + linkedin | More contents being innovative and sustainable development for community and planet |
| Amazon | AWS | Edge computing | Cash provided | Less | AWS + ecommerce | Less about human being welfare but more concentrated on its own business |

Summary Table from Cloud Tweets

| Companies | Cloud business | Brand name | Positive sentiment | Strategy priority | Business areas | Other features | Uniqueness |
|---|---|---|---|---|---|---|---|
| Google | GCP | Strong | Weak | Access | Crypto, bitcoin mining | Famous universities | “cryptocurrency”, “mining” |
| Microsoft | Azure | Strong | Strong | | Government, public policy, healthtech, managerial, financial, operational, accounting | Certifications | “government cloud”, “public policy” |
| Amazon | AWS | Strong | Medium | Access | Fleet management, user experience and customer experience | Partnering with public sector | “reinvest”, “uniware”, “partner” |

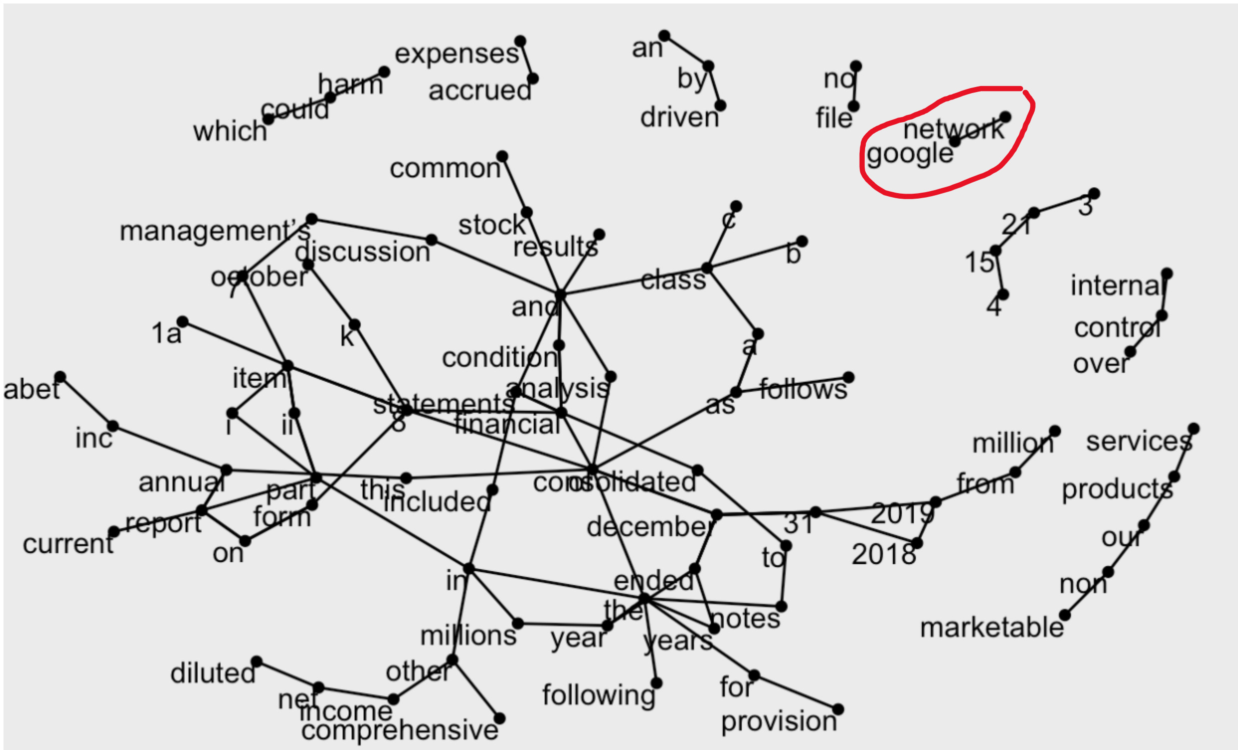
## 1.Annual report NLP analysis

# Business insights:

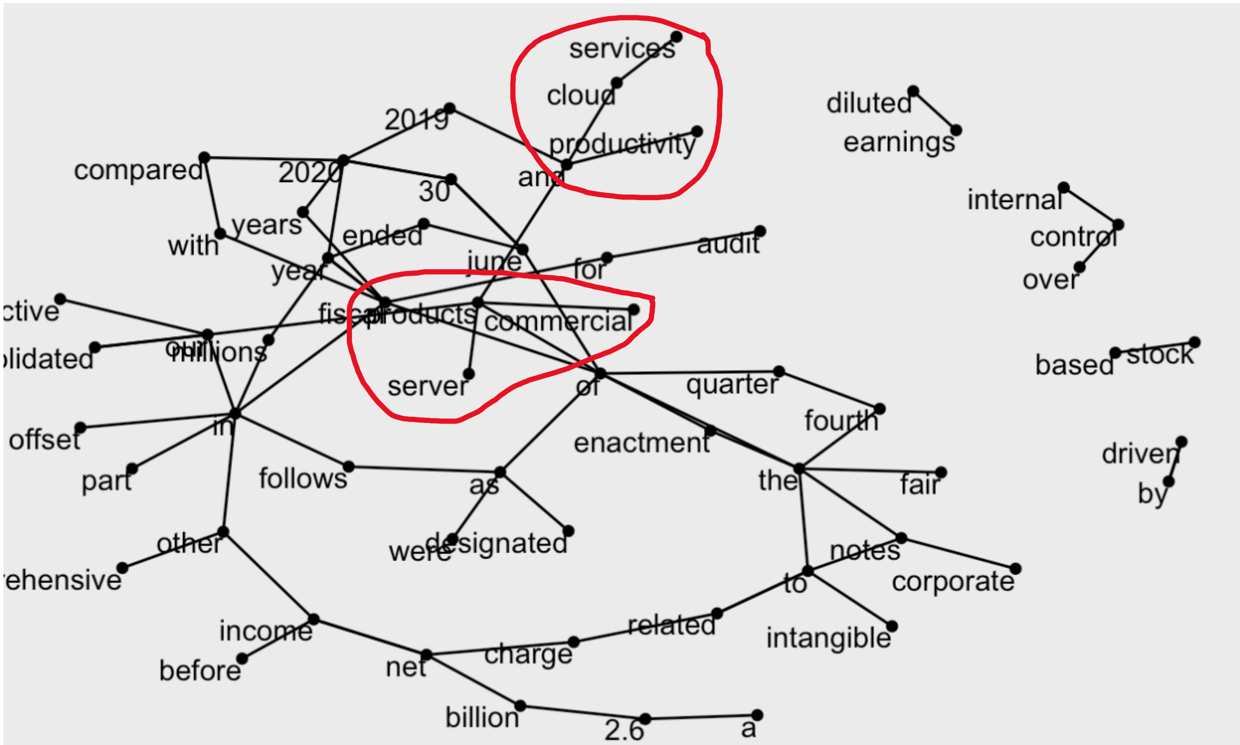
Microsoft and Amazon's annual reports talk about edge computing, the key differentiator among the major cloud service providers.

For amazon it tries more on data archives, which means it is generating lots of market share in the cloud computing industry. It also shows its current leading position in the cloud business, a cash-generating business.

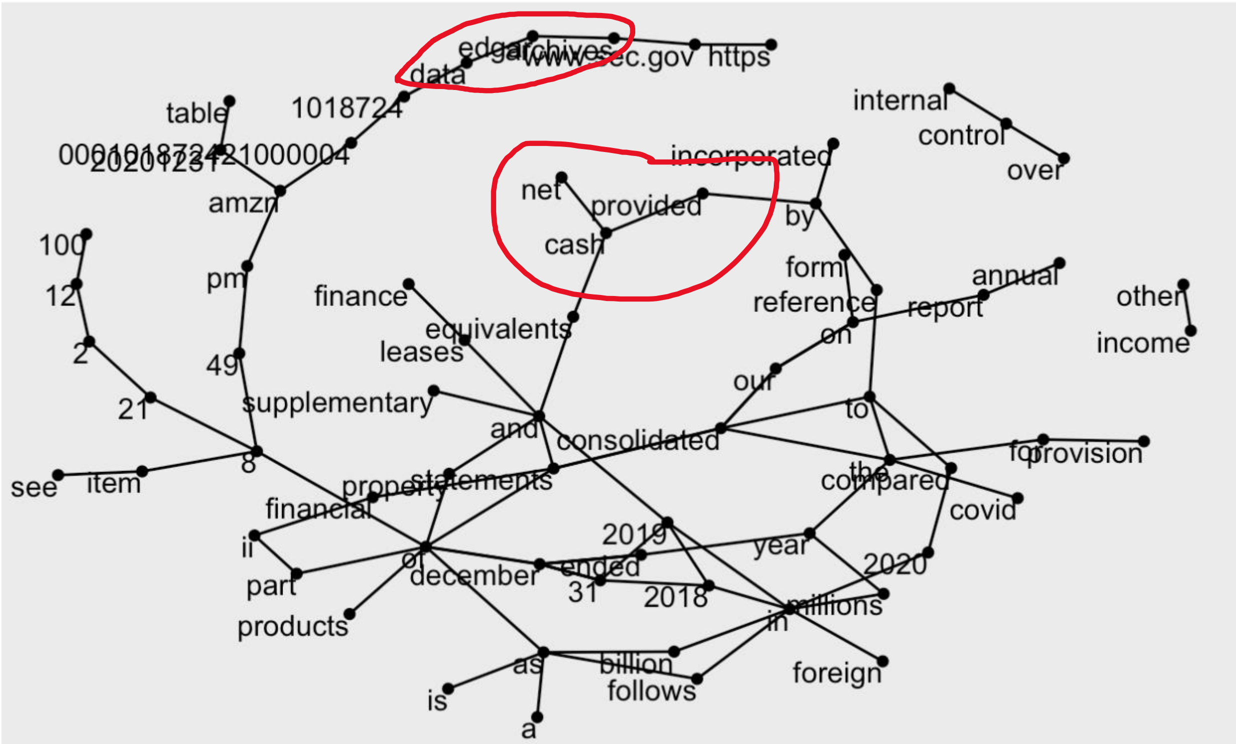
# Google




# Microsoft

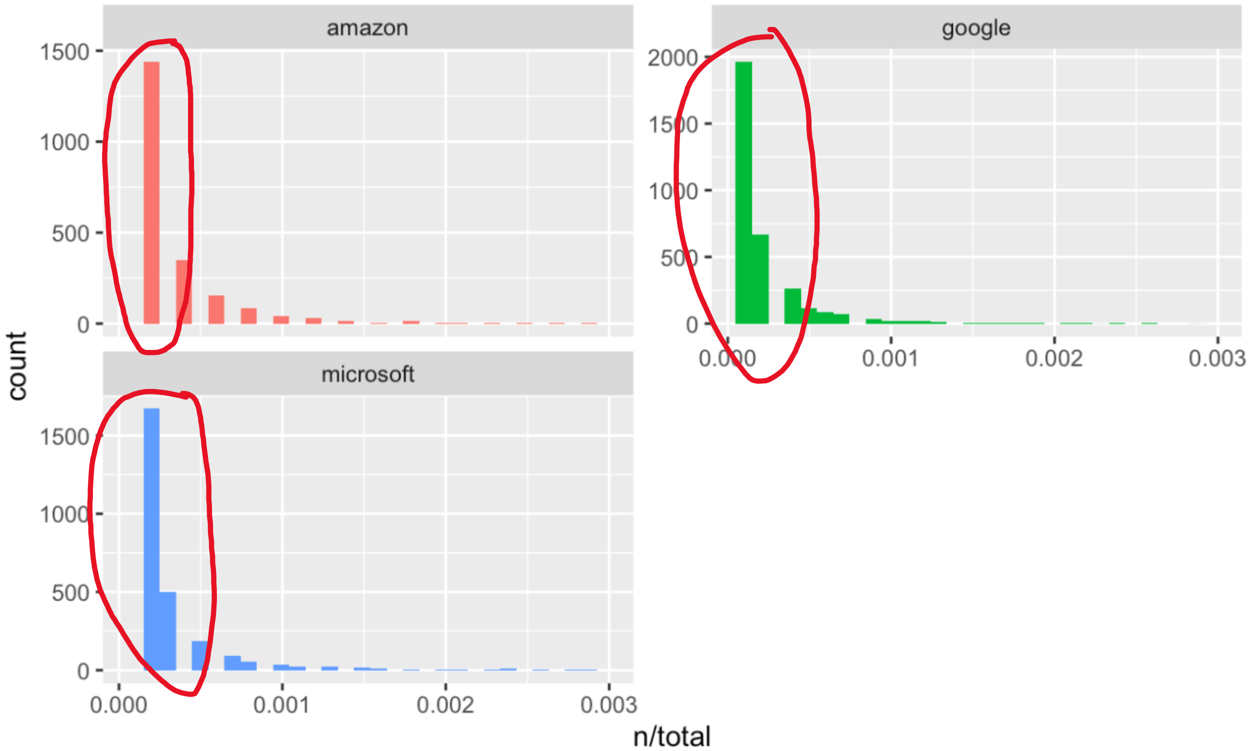

services
cloud
productivity
and
2019
2020
30
compared
with
years
year
ended
june
for
audit
fiscal
products
commercial
server
millions
our
offset
part
in
follows
as
were
designated
of
quarter
fourth
enactment
the
fair
notes
corporate
to
intangible
related
charge
net
billion
2.6
a
income
before
other
diluted
earnings
internal
control
over
based
stock
driven
by

# Amazon



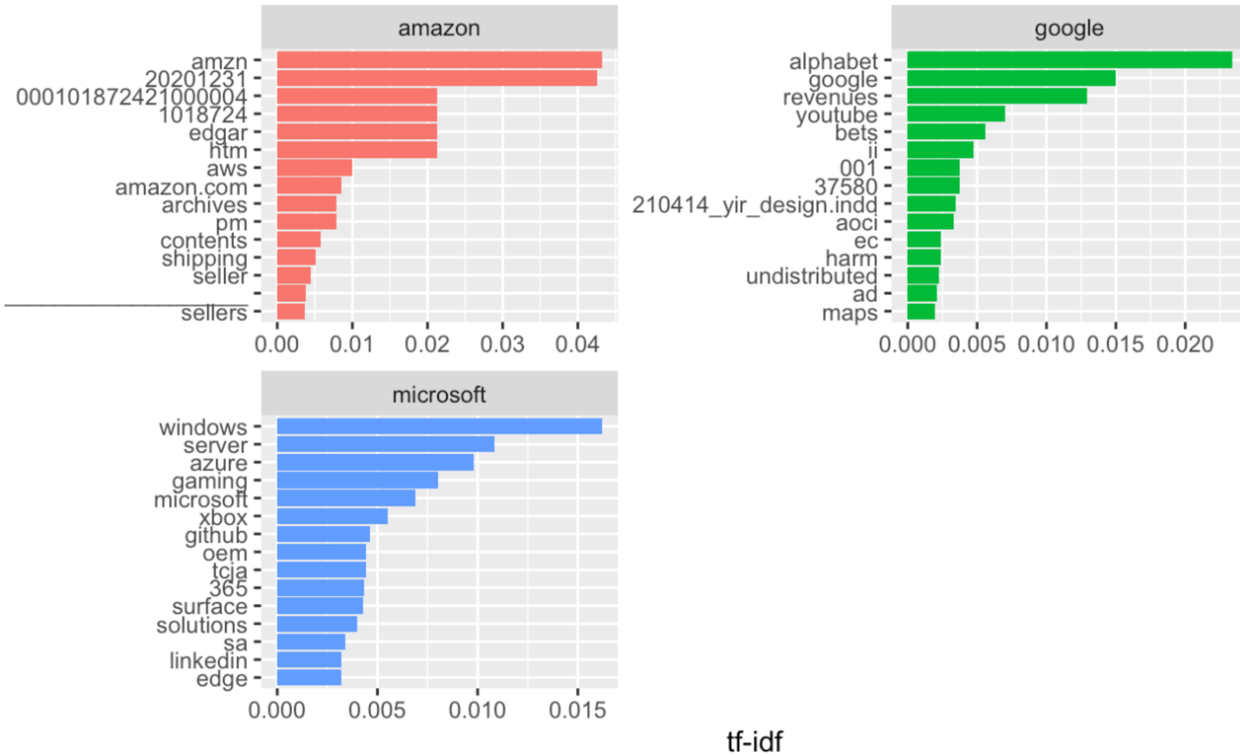

# Business insights:

Google and Microsoft annual reports have a more diverse character base than Amazon annual reports. Since the annual report somehow follows standard format for providing information for investors, Amazon uses fewer words than the other two companies.



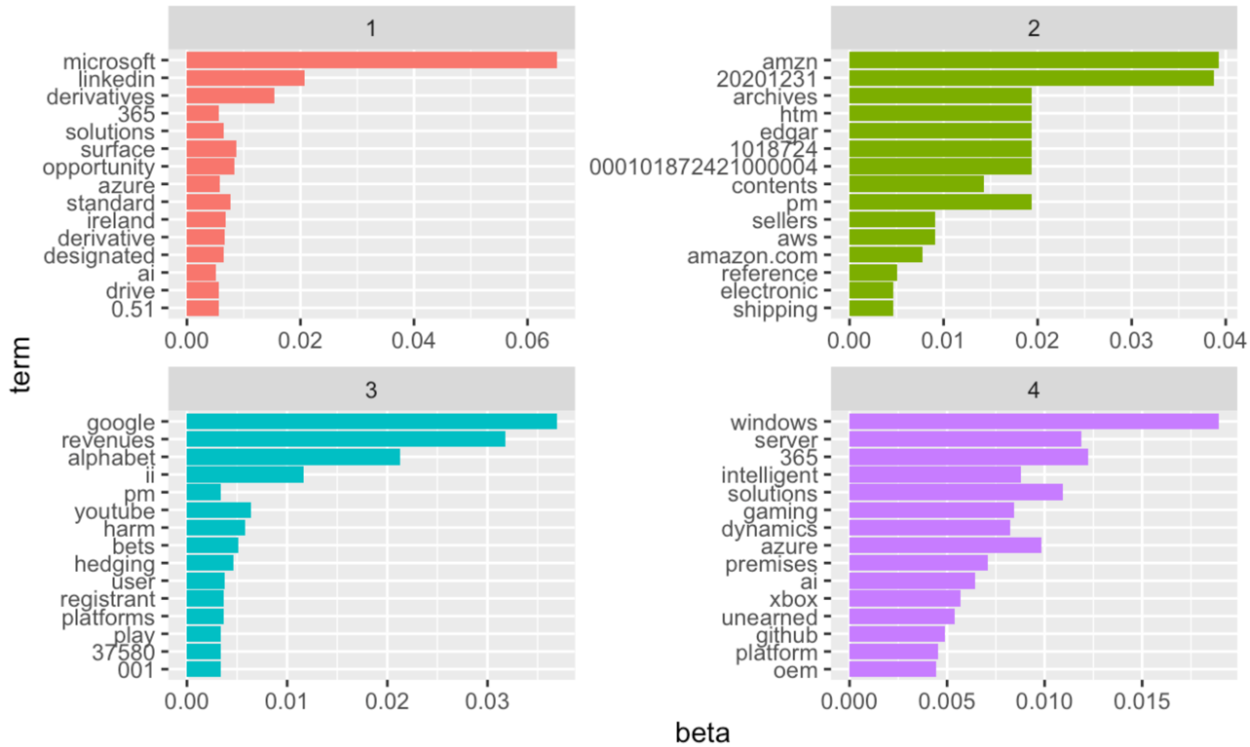

# Business insights:

TF-IDF shows the top 15 unique tokens for each company. For amazon, cloud business-related words such as "aws" and "archive", eCommerce related words like "shipping" and "sellers" are highly unique for describing and distincting Amazon. "Youtube", "ad" and "maps" are distinguish for Google. "Windows", "azure", "GitHub" and "LinkedIn" are very unique for Microsoft business. We can clearly tell the most popular or important business units for these three companies based on the IDF graph.




# Business insights:

Topic 1 describes Microsoft, topic 2 is for amazon, topic 3 says for google, topic 4 is more. NLP algorithms automatically separated the topic 4 as innovative business related to computing.




# Business insights:

Google and Microsoft's annual report has more innovative and sustainable development for the community and planet (future business). Amazon's annual report mentioned less about human welfare but more concentrated on its own business.Comparison for commonness in topic 3: Google and topic 4: future business

| | term | log_rate_google_cloud |
|---|---|---|
| 1 | calls | -2.59936564 |
| 2 | commence | -2.43706698 |
| 3 | headcount | -2.39745097 |
| 4 | workplace | -2.33091419 |
| 5 | materials | -2.19546121 |
| 6 | students | -1.94179127 |
| 7 | crisis | -1.79104531 |
| 8 | carbon | -1.73651191 |
| 9 | hedge | -1.64809671 |
| 10 | align | -1.64324331 |
| 11 | derivatives | -1.40870235 |
| 12 | play | -1.36450732 |
| 13 | audiences | -1.34596811 |
| 14 | user | -1.34501767 |
| 15 | 738 | -1.31484640 |
| 16 | sustainability | -1.15349844 |
| 17 | black | -1.10678523 |
| 18 | strong | -1.05828159 |
| 19 | erp | -1.04056706 |

| | term | log_rate_google_cloud |
|---|---|---|
| 1 | questions | 2.80783039 |
| 2 | standard | 2.79838956 |
| 3 | repurchased | 2.77543776 |
| 4 | live | 2.72392810 |
| 5 | transformation | 2.69939763 |
| 6 | hybrid | 2.62513247 |
| 7 | achieve | 2.56809971 |
| 8 | planet | 2.42205850 |
| 9 | productive | 2.39299324 |
| 10 | 2.2 | 2.34762462 |
| 11 | ai | 2.27813518 |
| 12 | alliances | 2.20783574 |
| 13 | consoles | 2.18573731 |
| 14 | learn | 2.14023841 |
| 15 | insights | 2.11717699 |
| 16 | closing | 1.95789582 |
| 17 | canadian | 1.93531385 |
| 18 | ecosystem | 1.76729197 |
| 19 | strive | 1.66853061 |
| 20 | role | 1.57086064 |
| 21 | experiences | 1.53930914 |
| 22 | certificates | 1.52924838 |
| 23 | entertainment | 1.49449095 |
| 24 | metrics | 1.49341656 |
| 25 | community | 1.45028838 |
| 26 | understand | 1.40148647 |
| 27 | oracle | 1.39756588 |
| 28 | platform | 1.32367569 |
| 29 | transforming | 1.31475161 |
| 30 | shipping | 1.26142100 |
| 31 | opportunity | 1.25974091 |
| 32 | guidelines | 1.20760053 |
| 33 | function | 1.20226181 |
| 34 | world's | 1.19514959 |
| 35 | accelerate | 1.13300954 |

Comparison for commonness in topic 2: Amazon and topic 4: future business

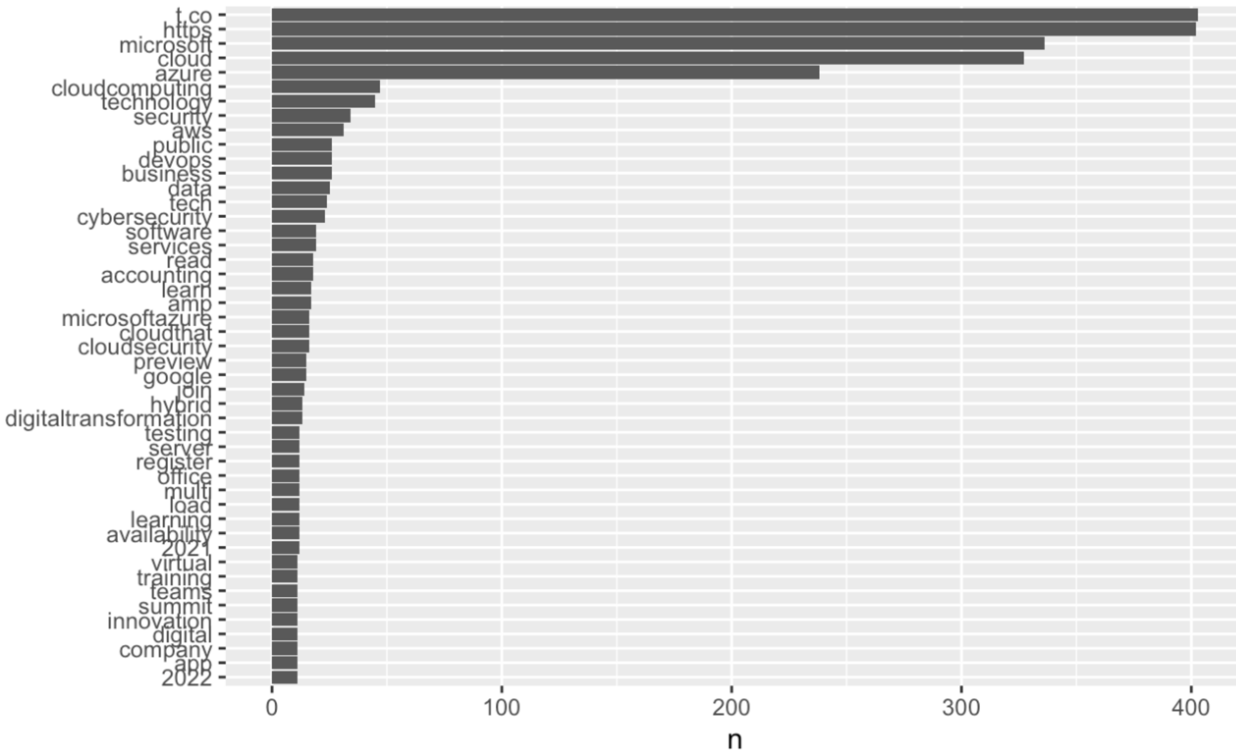
| | term | log_rate_google_cloud | log_rate_amazon_cloud |
|---|---|---|---|
| 1 | premises | 157.4152 | 5.19116527 |
| 2 | distinct | 158.4049 | 3.70294961 |
| 3 | methodology | 157.5468 | 3.36280152 |
| 4 | doubtful | 155.7188 | 2.57820806 |
| 5 | 2.13 | 157.7903 | 2.50602439 |
| 6 | virtual | 154.1599 | 2.29896091 |
| 7 | 157 | 156.8165 | 2.19843732 |
| 8 | unearned | 159.3126 | 1.62844184 |
| 9 | indices | 159.0425 | 1.58662741 |
| 10 | percent | 155.5369 | 1.00054777 |
| 11 | enterprises | 156.3588 | 0.61301602 |
| 12 | 15.8 | 155.5056 | 0.36510516 |
| 13 | combined | 156.0691 | 0.32376334 |
| 14 | inventories | 155.7347 | 0.24535770 |
| 15 | producing | 156.4671 | 0.03646300 |
| 16 | lines | 153.6613 | -0.03735544 |
| 17 | database | 154.8120 | -0.36645698 |
| 18 | preparing | 152.1937 | -0.50684915 |
| 19 | shareholders | 152.8448 | -1.07218412 |
| 20 | receivables | 154.8460 | -1.08589050 |
| 21 | subsequent | 154.7024 | -1.48098266 |
| 22 | vendor | 154.8228 | -1.65686791 |
| 23 | grade | 152.3866 | -2.04770811 |
| 24 | selection | 153.4819 | -2.33500389 |
| 25 | deductions | 153.2416 | -2.41443127 |
| 26 | warrant | 152.0251 | -2.55795182 |
| 27 | omnichannel | 151.2322 | -2.65897989 |
| 28 | 411 | 151.5329 | -2.68650478 |
| 29 | entry | 153.6667 | -2.75287606 |
| 30 | absolute | 152.6242 | -3.42569915 |
| 31 | electronic | 146.3717 | -4.50808889 |

## 2.Cloud tweets NLP analysis

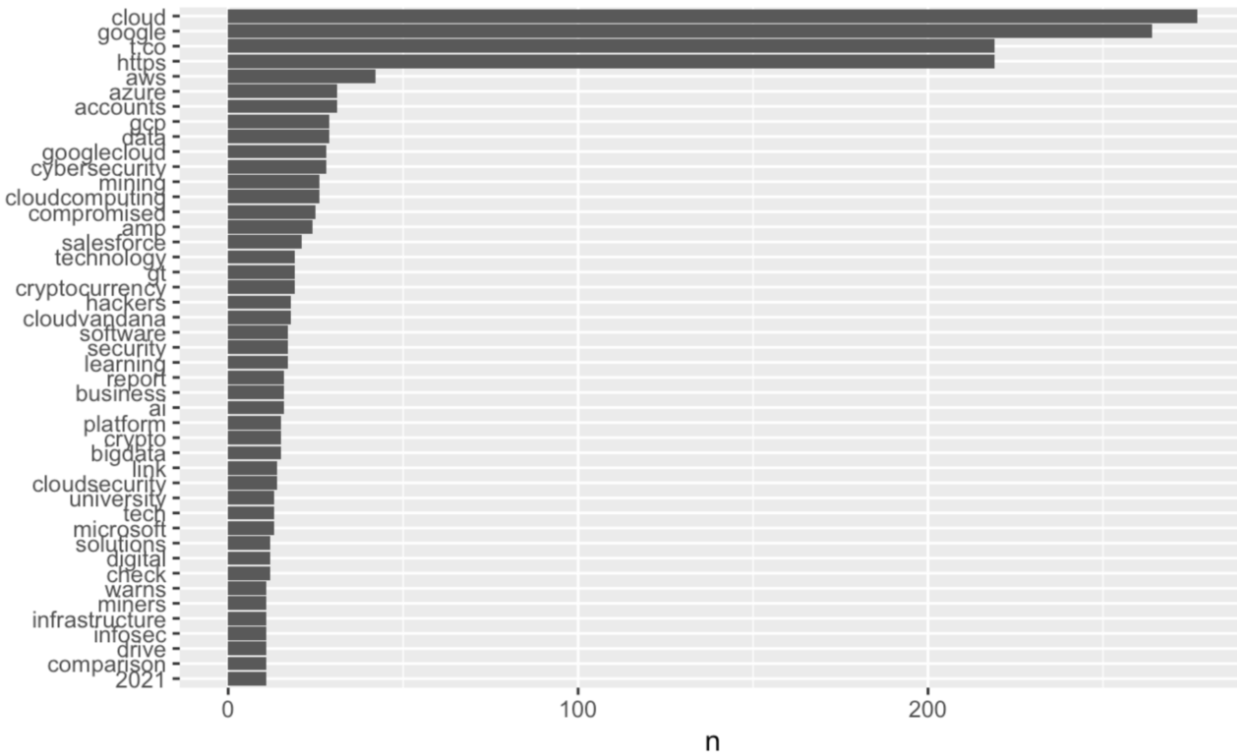
# Business insights:

"Azure" and "AWS" are frequently mentioned in the three companies' cloud tweets. But "GCP" gets mentioned a lot in cloud tweets related to Google.

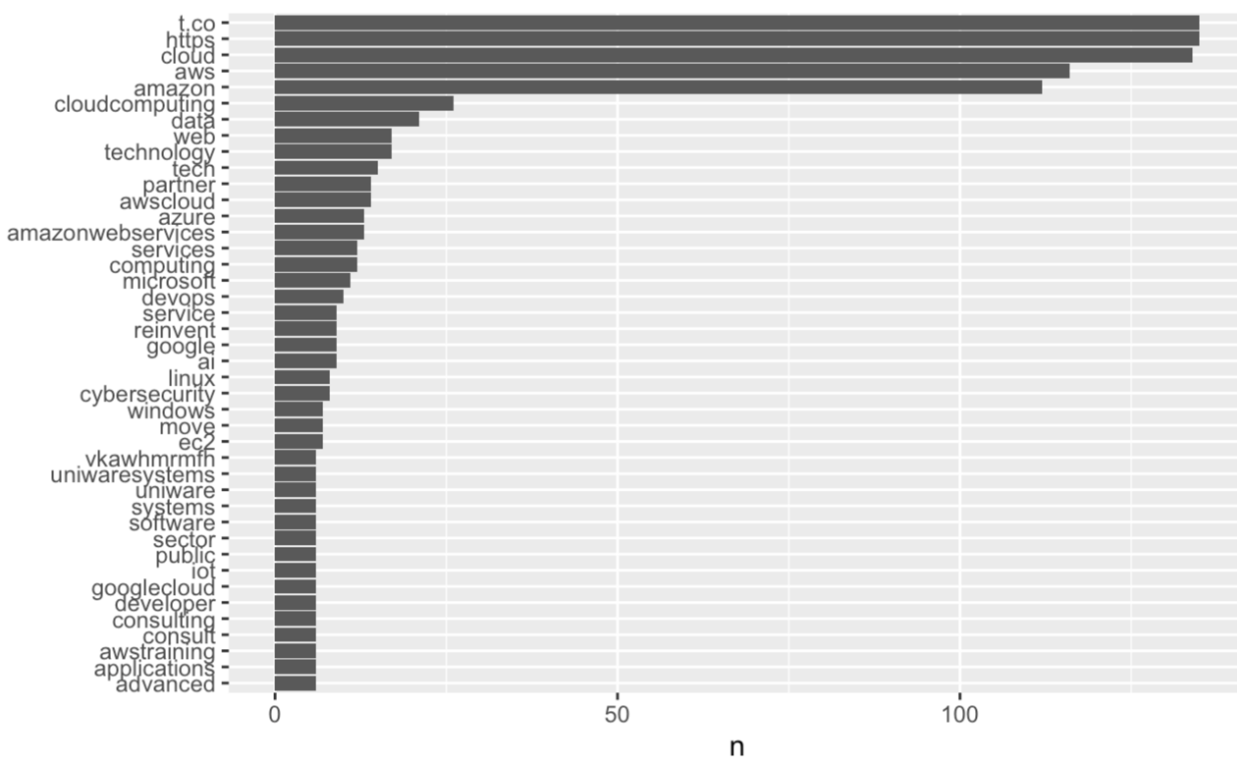
# Microsoft




# Google



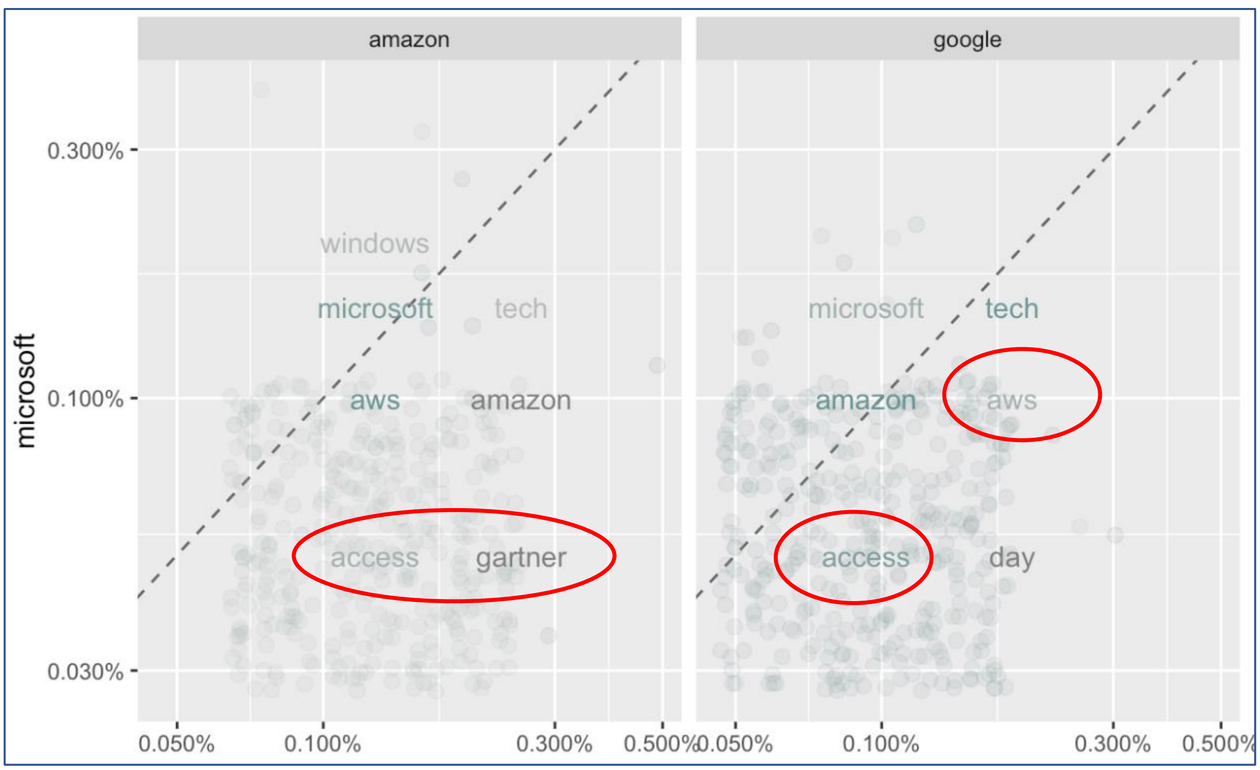

# Amazon




# Business insights

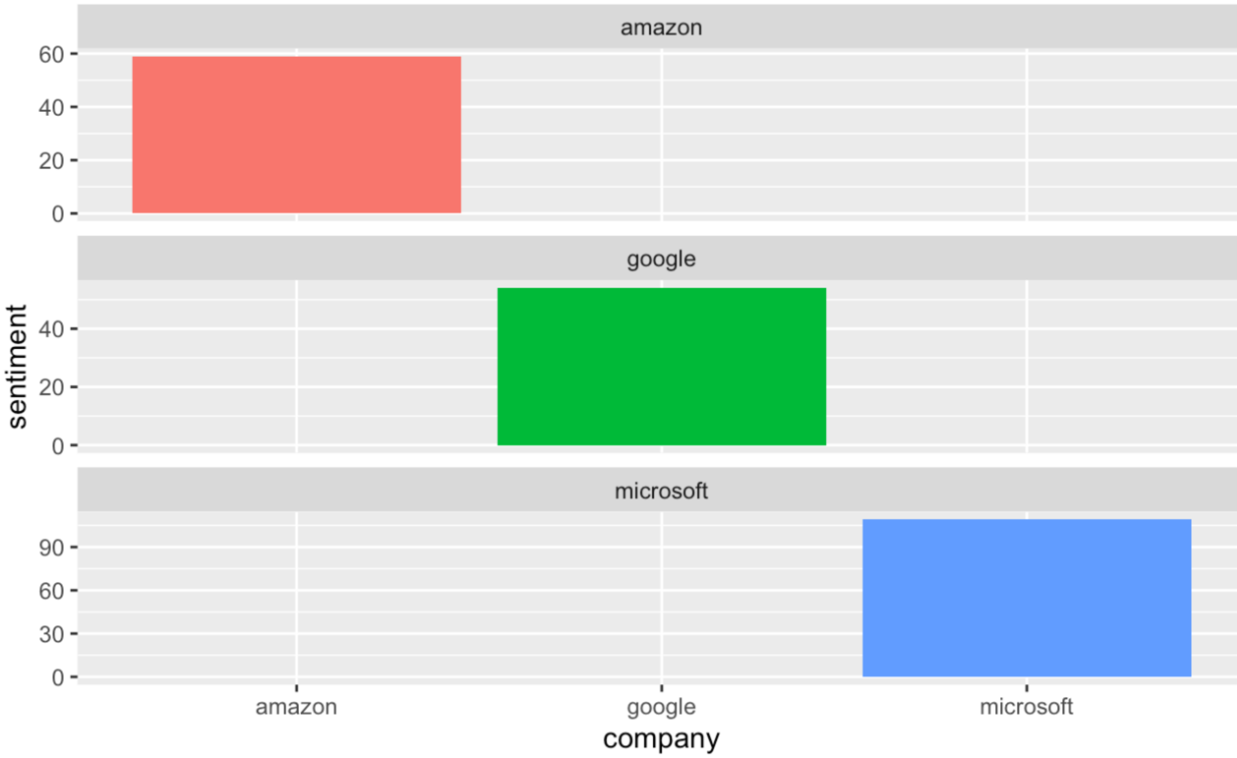
Compared with Microsoft, Amazon tweets talk more about "access" and "Gartner". The frequency of Gartner may come from the Gartner Magic Quadrants posted in August 2021. In the complete report, "AWS Named as a Leader for the 11th Consecutive Year in 2021 Gartner Magic Quadrant for Cloud Infrastructure & Platform Services (CIPS)." (AWS Named as a Leader for the 11th Consecutive Year in 2021 Gartner Magic Quadrant for Cloud Infrastructure & Platform Services (CIPS). (2021, August 2))

Compared with Microsoft, Google mentioned more about AWS, indicating that Google's benchmark for cloud business is AWS, the first market player in cloud. It may relate to similar growth strategy for google cloud as AWS. Because google tweets also mentioned "access" frequently. Considering that "Google Cloud has been hiring executives to sell into industries and has ramped its Anthos hybrid cloud effort to close its AWS and Azure sales gap." (Dignan, L. (2021, April 2))



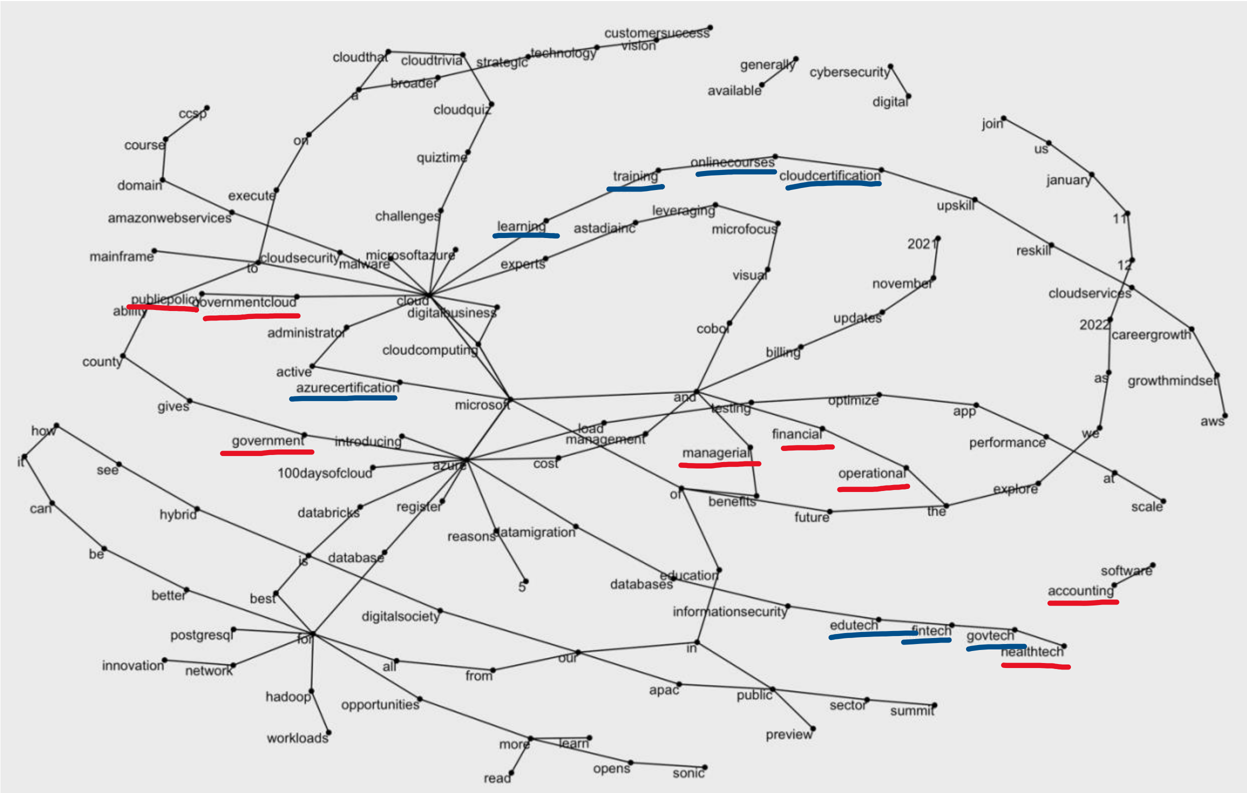

# Business insights:

After excluding other flavors in the nrc lexicon, Microsoft cloud-related tweets have the strongest positive sentiment, google has the weakest positive sentiment.



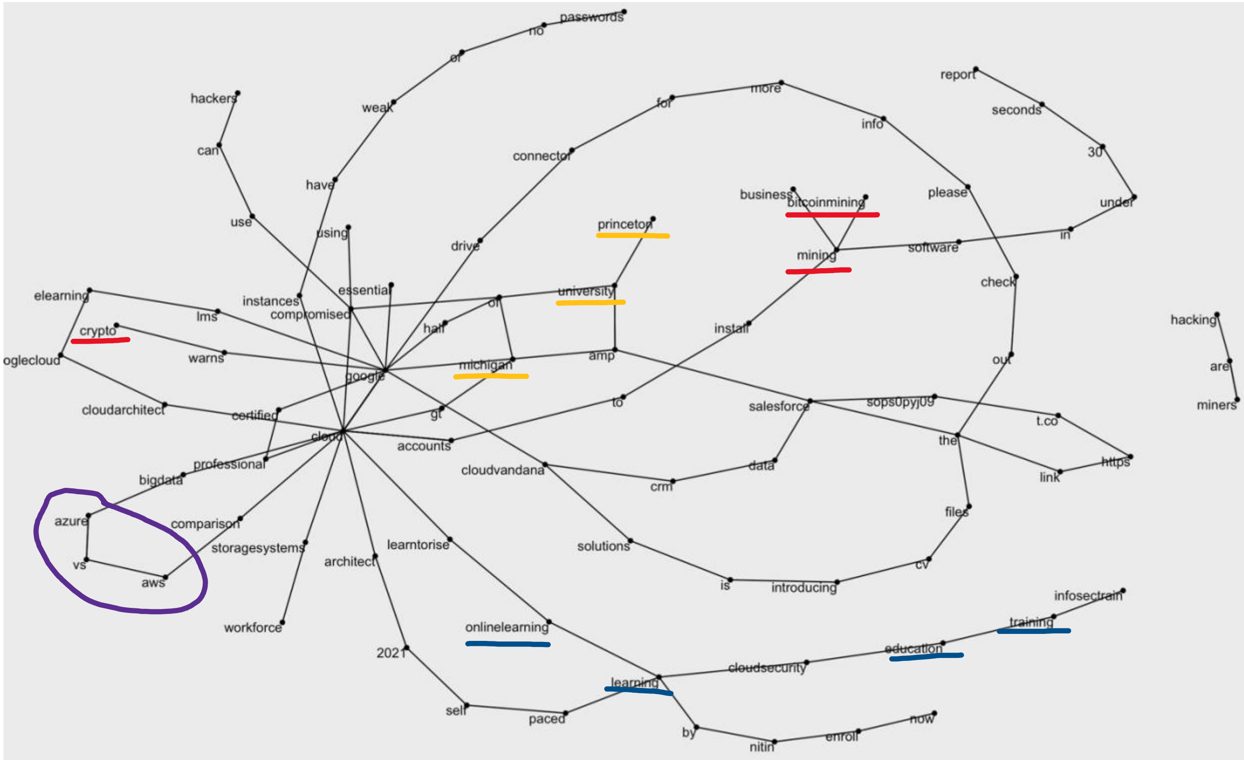

# Microsoft

Cloud tweets related to Microsoft mention business areas in government, public policy, healthtech, managerial, financial, operational, accounting. Compared with the other two competitors, Azure seems to have broader applications in business areas. Certifications from Azure also mentioned frequently on tweeter.



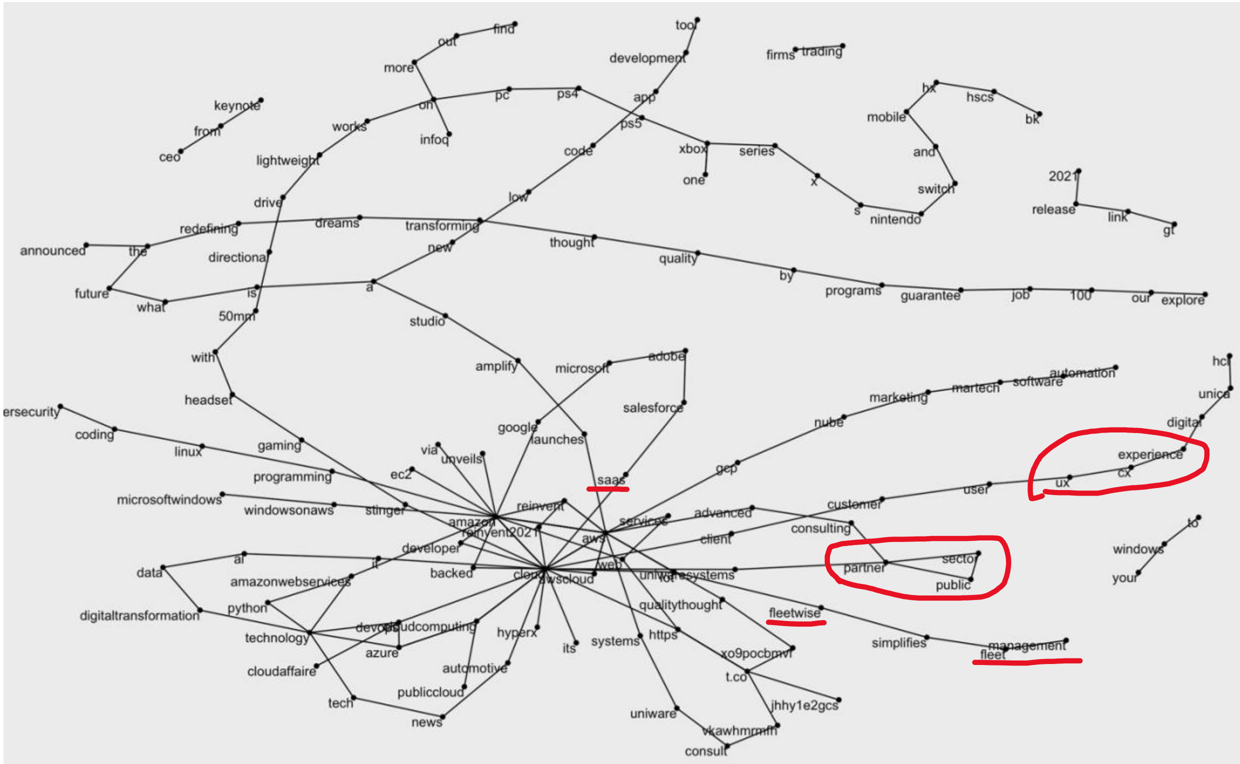
# Google

Interesting areas mentioned in tweets related to Google cloud is crypto, bitcoin mining. And some famous universities are mentioned frequently as well, such as Michigan, Princeton. This may be because Google Sheets which is part of Google cloud service are provided to university for free. According to recent news, "Google has released a new report stating that malicious cryptocurrency miners are using hacked Google Cloud accounts for mining purposes." (Papadopoulos, L. (2021, November 27)) People also discuss more about the comparison between Google cloud and Azure, AWS.



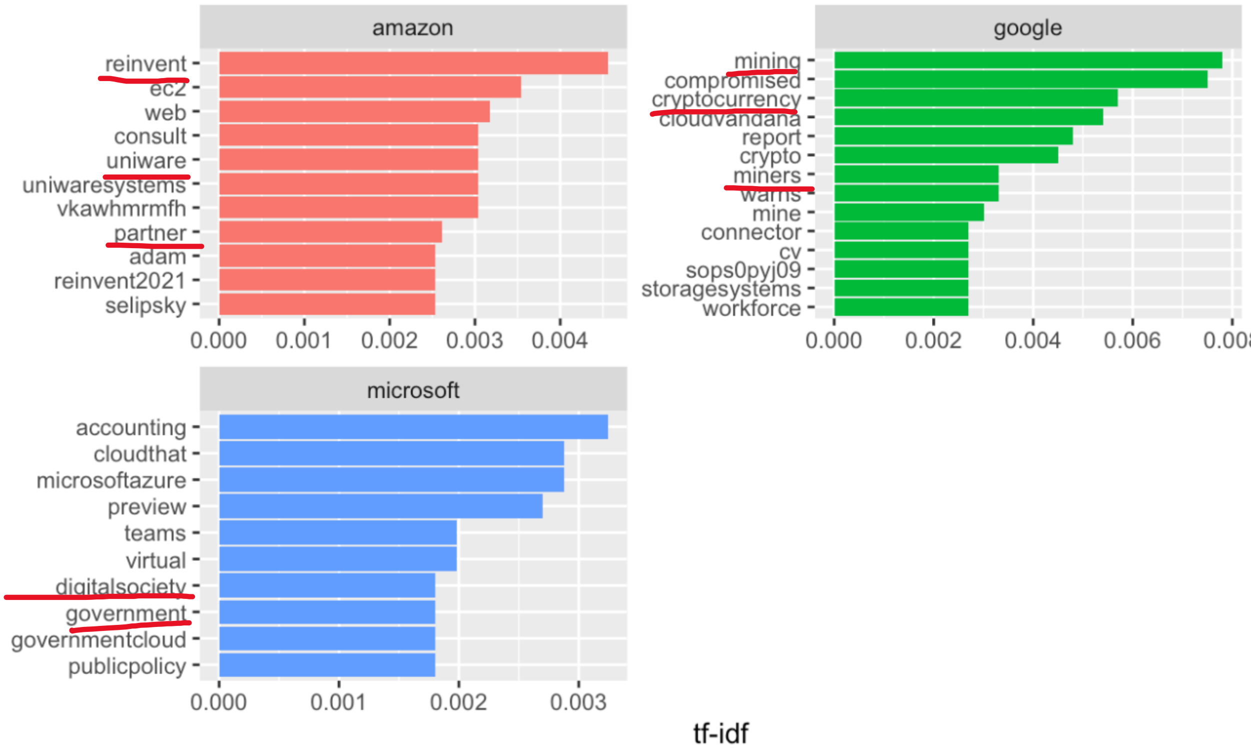

# Amazon

The business applications mentioned more with AWS are fleet management, user experience and customer experience. Partnering with public sector is also mentioned in tweets.




# Business insights:

The unique words for amazon cloud are "reinvest", "uniware", "partner". The uniqueness for Google cloud is "cryptocurrency", "mining". The unique words in Microsoft cloud are "government cloud", "public policy".



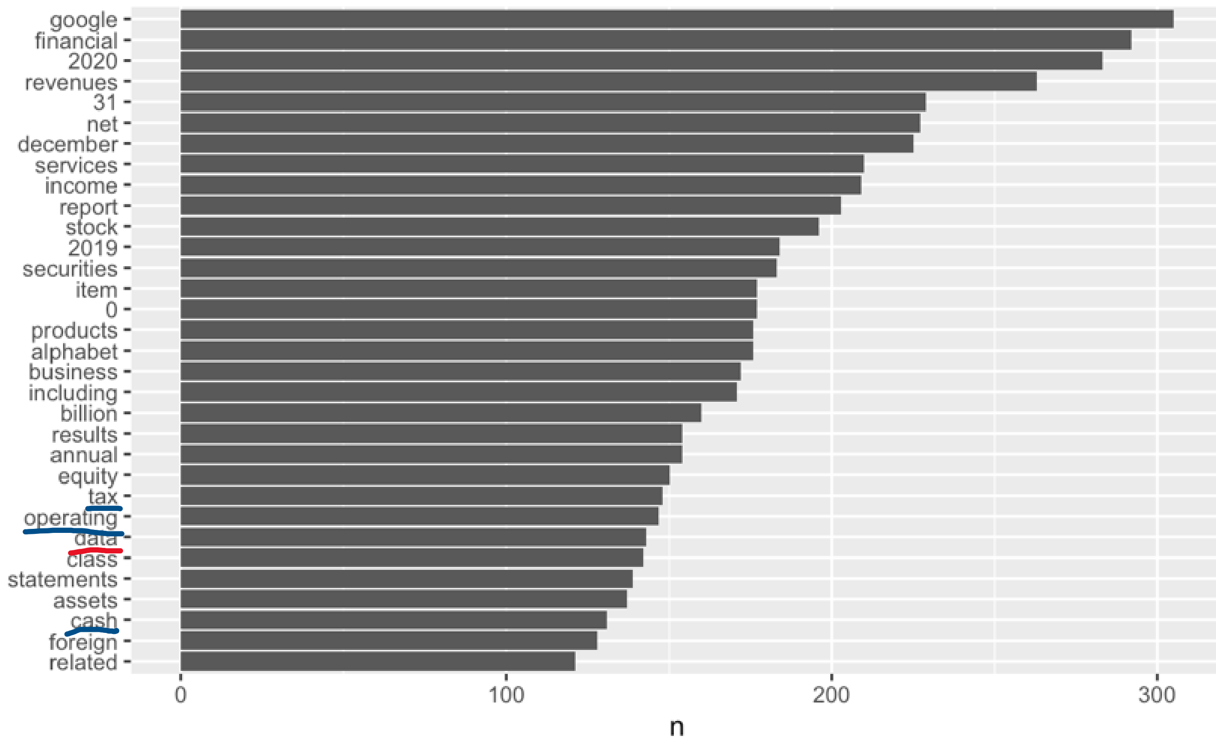
# References

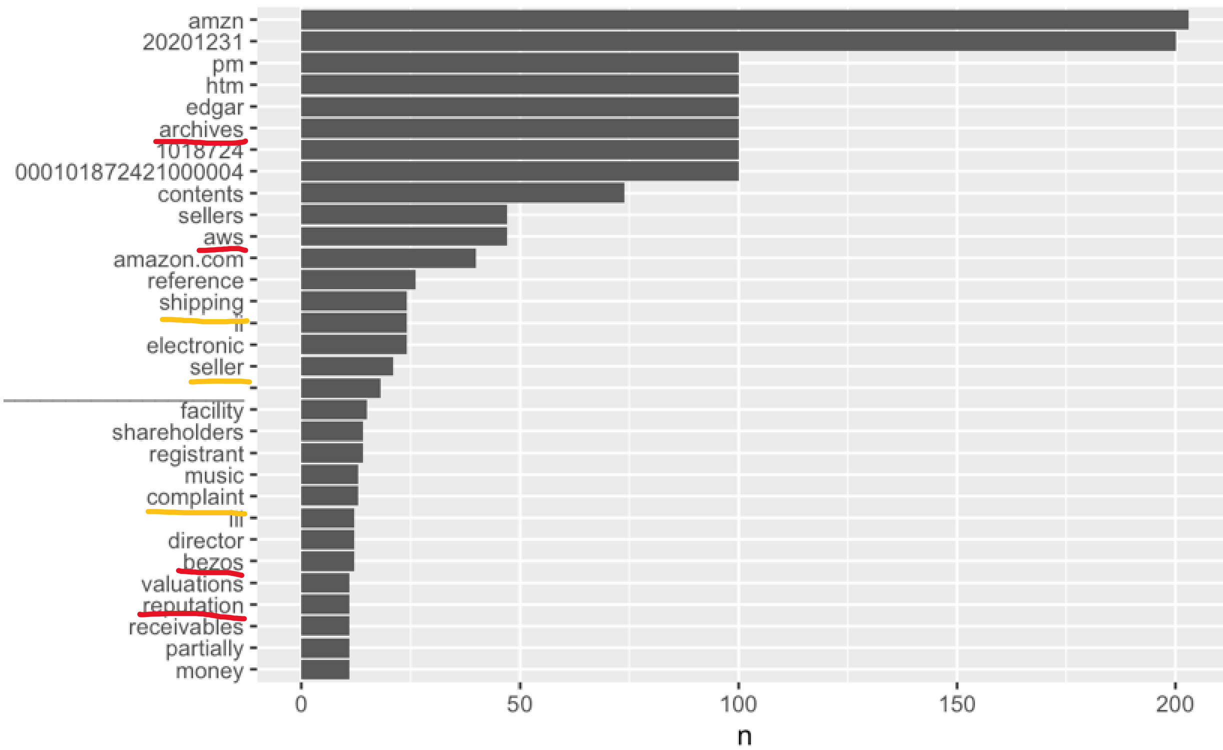

1. Papadopoulos, L. (2021, November 27). *Rogue Miners Are Using Google Cloud Servers to Mine Cryptocurrencies*. Interestingengineering. https://interestingengineering.com/rogue-miners-are-using-google-cloud-to-mine-crypto
2. Dignan, L. (2021, April 2). *Top cloud providers in 2021: AWS, Microsoft Azure, and Google Cloud, hybrid, SaaS players*. ZDNet. https://www.zdnet.com/article/the-top-cloud-providers-of-2021-aws-microsoft-azure-google-cloud-hybrid-saas/
3. *AWS Named as a Leader for the 11th Consecutive Year in 2021 Gartner Magic Quadrant for Cloud Infrastructure & Platform Services (CIPS)*. (2021, August 2). Amazon Web Services. https://aws.amazon.com/blogs/aws/aws-named-as-a-leader-for-the-11th-consecutive-year-in-2021-gartner-magic-quadrant-for-cloud-infrastructure-platform-services-cips/

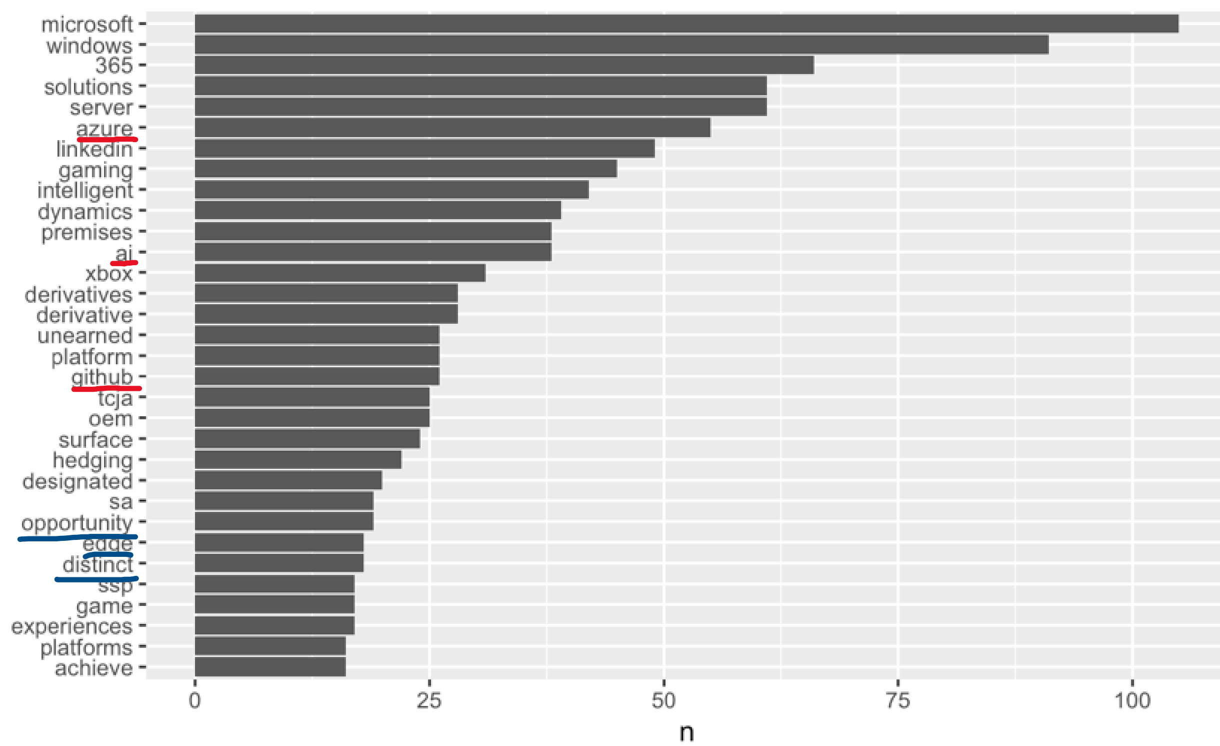
# Appendix. NLP Algorithms in R with graph output

```
##################################################################################
########################## Data from annual report ##############################
##################################################################################
# extracting text data from 2020 annual reports
library(pdftools) # we need this library to use pdf_text
setwd("/Users/lilialyssali/Downloads/Text Analytics/Financial Reports")
nm <- list.files(path="/Users/lilialyssali/Downloads/Text Analytics/Financial Reports")

google_data <- read_document(file=nm[1]) #This comes out as a vector, a list of strings
google_data_together <- paste(google_data, collapse = " ") # This will give us a concatenated
vector, one string
google_text <- do.call(rbind, lapply(nm[1], function(x) paste(read_document(file=x), collapse =
" "))) # each string for each file
google <- data.frame(google_text)
colnames(google) <- "text"

microsoft_data <- read_document(file=nm[3]) #This comes out as a vector, a list of strings
microsoft_data_together <- paste(microsoft_data, collapse = " ") # This will give us a
concatenated vector, one string
microsoft_text <- do.call(rbind, lapply(nm[3], function(x) paste(read_document(file=x), collapse
= " "))) # each string for each file
microsoft <- data.frame(microsoft_text)
colnames(microsoft) <- "text"

amazon_data <- read_document(file=nm[2]) #This comes out as a vector, a list of strings
amazon_data_together <- paste(amazon_data, collapse = " ") # This will give us a concatenated
vector, one string
amazon_text <- do.call(rbind, lapply(nm[2], function(x) paste(read_document(file=x), collapse =
" "))) # each string for each file
amazon <- data.frame(amazon_text)
colnames(amazon) <- "text"
# building tidy format
tidy_google <- google %>%
  unnest_tokens(word,text) %>%
  anti_join(stop_words) %>%
  count(word,sort = TRUE)

tidy_microsoft <- microsoft %>%
  unnest_tokens(word,text) %>%
  anti_join(stop_words) %>%
  count(word,sort=TRUE)

tidy_amazon <- amazon %>%
  unnest_tokens(word,text) %>%
  anti_join(stop_words) %>%
```

```
  count(word,sort = TRUE)

# exclude common words in three tidy datasets
inner_1 <- tidy_google %>% inner_join(tidy_microsoft, by = "word")
inner_2 <- inner_1 %>% inner_join(tidy_amazon,by="word")
tidy_google <- anti_join(tidy_google,inner_2,by="word")
tidy_amazon <- anti_join(tidy_amazon,inner_2,by="word")
tidy_microsoft <- anti_join(tidy_microsoft,inner_2,by="word")

###########################   term frequency histogram  ###########################
# google
google_freq_hist <- tidy_google %>%
  mutate(word=reorder(word, n)) %>%
  filter(n>120) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(google_freq_hist)
# microsoft
microsoft_freq_hist <- tidy_microsoft %>%
  mutate(word=reorder(word, n)) %>%
  filter(n>15) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(microsoft_freq_hist)
# amazon
amazon_freq_hist <- tidy_amazon %>%
  mutate(word=reorder(word, n)) %>%
  filter(n>100) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(amazon_freq_hist)
```

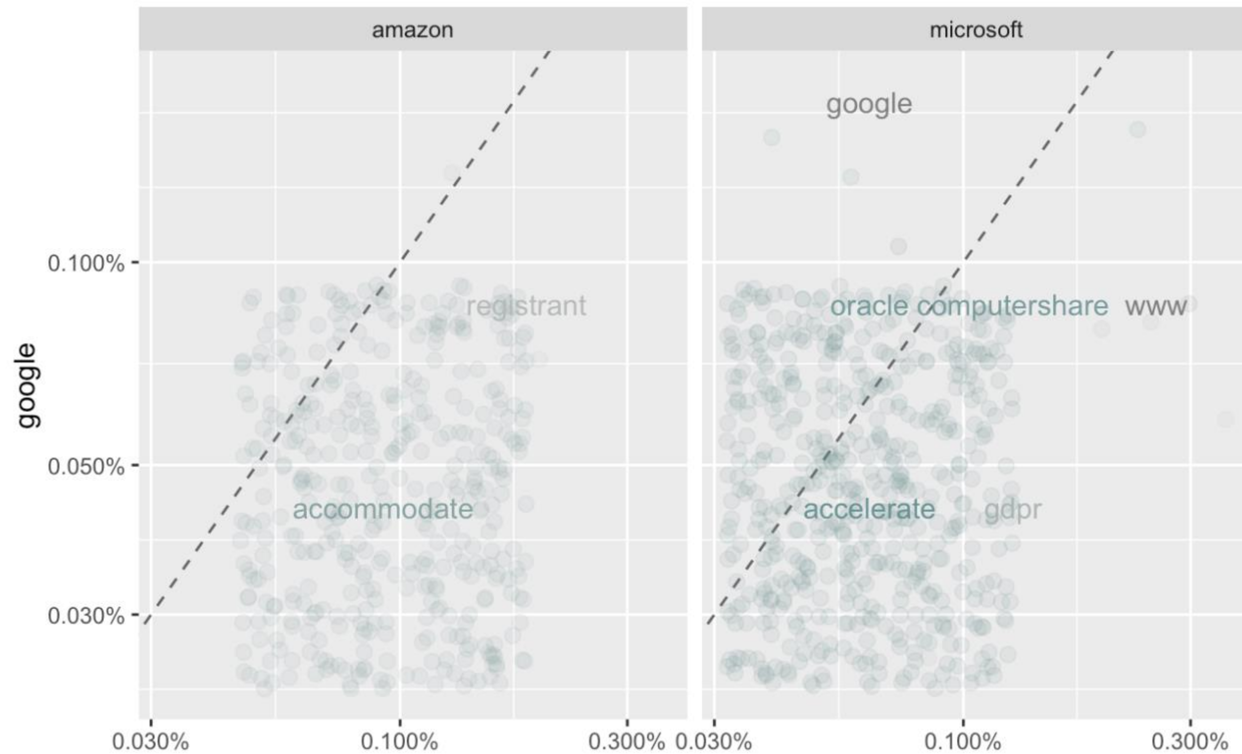

google
financial
2020
revenues
31
net
december
services
income
report
stock
2019
securities
item
0
products
alphabet
business
including
billion
results
annual
equity
tax
operating
data
class
statements
assets
cash
foreign
related
0
100
200
300
n




```
#################################   correlograms   ###############################
frequency <- bind_rows(mutate(tidy_google, author="google"),
                        mutate(tidy_microsoft, author= "microsoft"),
                        mutate(tidy_amazon, author="amazon")
)%>%#closing bind_rows
  mutate(word=str_extract(word, "[a-z']+")) %>%
```

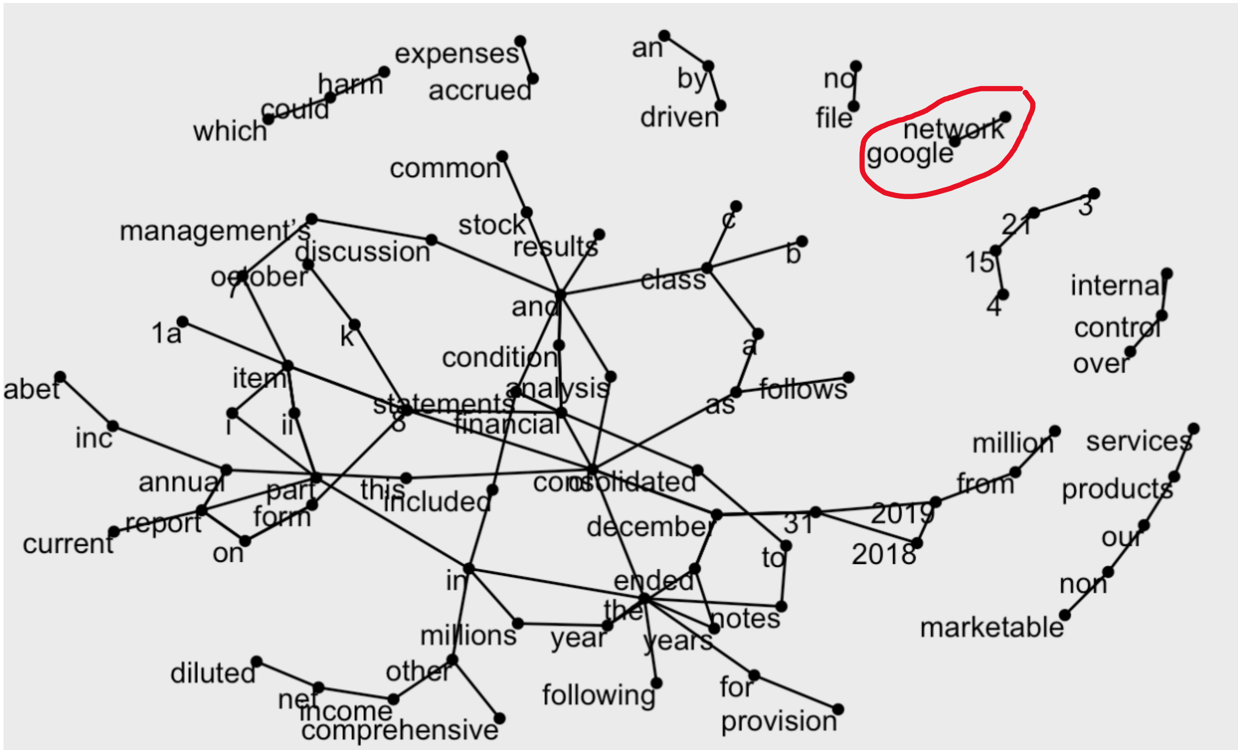
```
  filter(!nchar(word)==1) %>%
  filter(!nchar(word)==2) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n/sum(n))%>%
  select(-n) %>%
  spread(author, proportion) %>%
  gather(author, proportion, `microsoft`, `amazon`)

ggplot(frequency, aes(x=proportion, y=`google`,
                      color = abs(`google`- proportion)))+
  geom_abline(color="grey40", lty=2)+
  geom_jitter(alpha=.1, size=2.5, width=0.3, height=0.3)+
  geom_text(aes(label=word), check_overlap = TRUE, vjust=1.5) +
  scale_x_log10(labels = percent_format())+
  scale_y_log10(labels= percent_format())+
  scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+
  facet_wrap(~author, ncol=2)+
  theme(legend.position = "none")+
  labs(y= "google", x=NULL)

cor.test(data=frequency[frequency$author == "microsoft",],
         ~proportion + `google`)
cor.test(data=frequency[frequency$author == "amazon",],
         ~proportion + `google`)
```



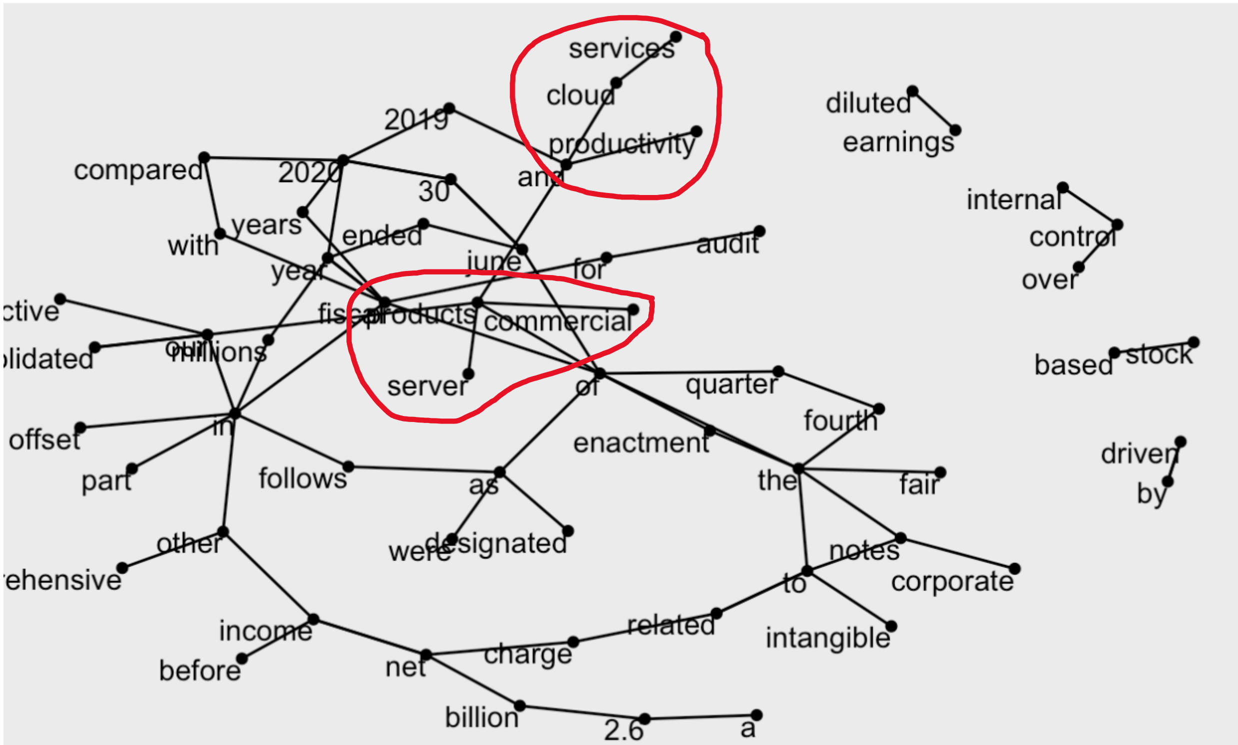

################################## Ngrams ######################################

```
# google
google_quadrograms <- google %>%
  unnest_tokens(quadrograms,text,token="ngrams",n=4) %>%
  separate(quadrograms,c("word1","word2","word3","word4"),sep = " ") %>%
  filter(!word1 %in% stop_words) %>%
  filter(!word2 %in% stop_words) %>%
  filter(!word3 %in% stop_words) %>%
  filter(!word4 %in% stop_words)

google_quadrograms_counts <- google_quadrograms %>%
  count(word1,word2,word3,word4,sort = TRUE)

google_quadrogram_graph <- google_quadrograms_counts %>%
  filter(n>10) %>%
  graph_from_data_frame()

ggraph(google_quadrogram_graph,layout="fr") +
  geom_edge_link() +
  geom_node_point() +
  geom_node_text(aes(label=name),vjust=1,hjust=1)
# microsoft
microsoft_quadrograms <- microsoft %>%
  unnest_tokens(quadrograms,text,token="ngrams",n=4) %>%
  separate(quadrograms,c("word1","word2","word3","word4"),sep = " ") %>%
```

```
  filter(!word1 %in% stop_words) %>%
  filter(!word2 %in% stop_words) %>%
  filter(!word3 %in% stop_words) %>%
  filter(!word4 %in% stop_words)

microsoft_quadrograms_counts <- microsoft_quadrograms %>%
  count(word1,word2,word3,word4,sort = TRUE)

microsoft_quadrogram_graph <- microsoft_quadrograms_counts %>%
  filter(n>8) %>%
  graph_from_data_frame()

ggraph(microsoft_quadrogram_graph,layout="fr") +
  geom_edge_link() +
  geom_node_point() +
  geom_node_text(aes(label=name),vjust=1,hjust=1)
# amazon
amazon_quadrograms <- amazon %>%
  unnest_tokens(quadrograms,text,token="ngrams",n=4) %>%
  separate(quadrograms,c("word1","word2","word3","word4"),sep = " ") %>%
  filter(!word1 %in% stop_words) %>%
  filter(!word2 %in% stop_words) %>%
  filter(!word3 %in% stop_words) %>%
  filter(!word4 %in% stop_words)

amazon_quadrograms_counts <- amazon_quadrograms %>%
  count(word1,word2,word3,word4,sort = TRUE)

amazon_quadrogram_graph <- amazon_quadrograms_counts %>%
  filter(n>10) %>%
  graph_from_data_frame()

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

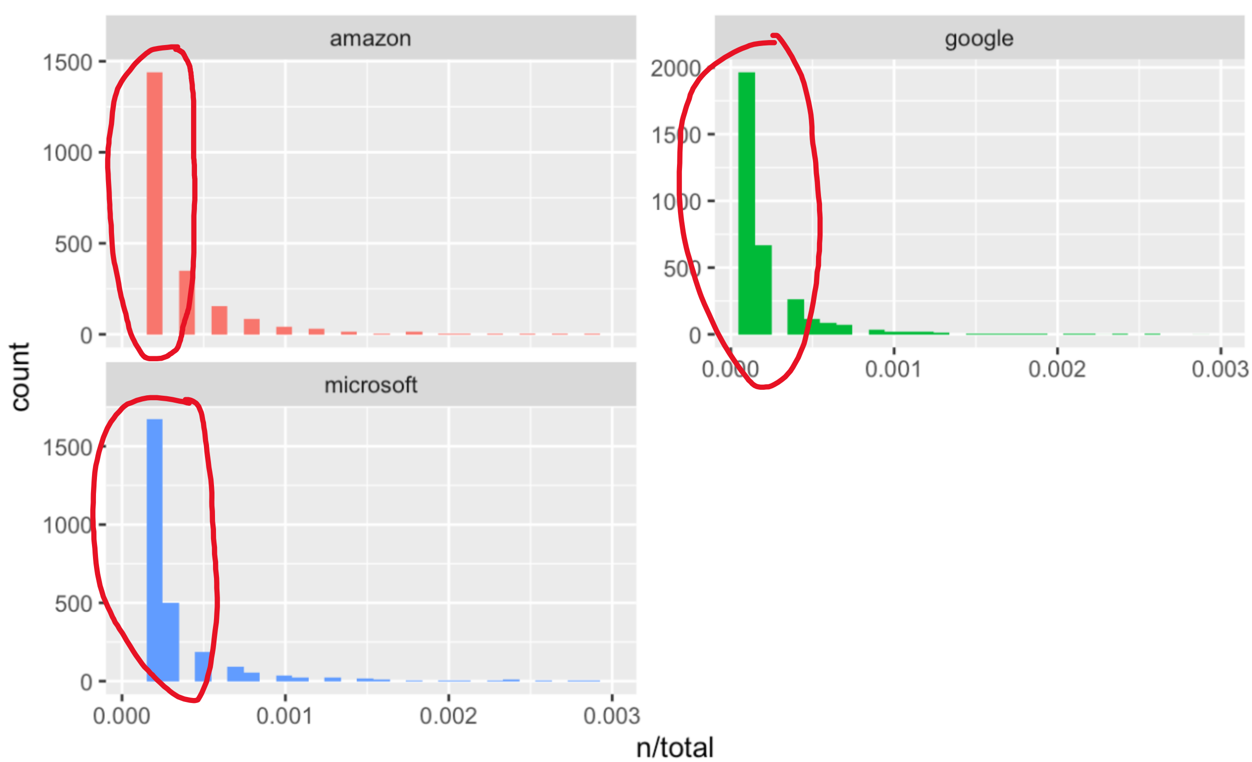


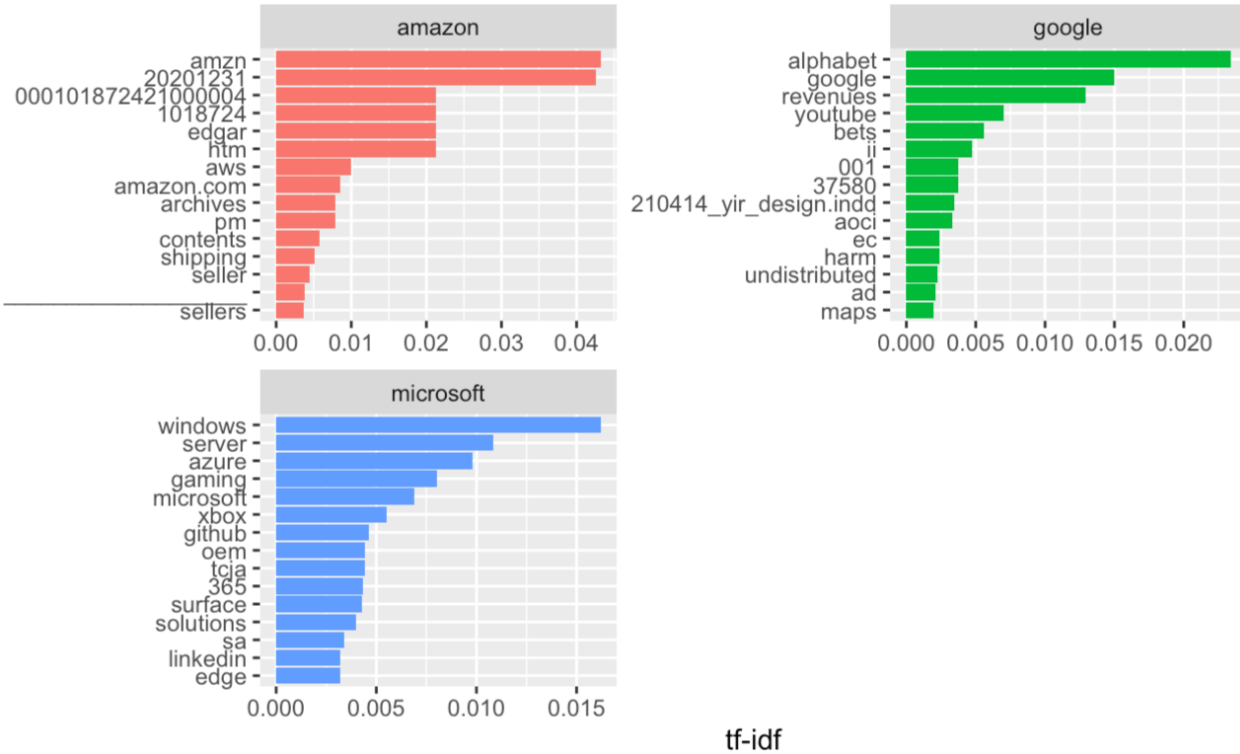
expenses
accrued
harm
could
which
an
by
driven
no
file
network
google
common
stock
management's
discussion
results
class
c
b
october
7
and
1a
k
condition
a
item
analysis
as
follows
abet
statements
i
ii
8
financial
inc
annual
part
this
included
consolidated
million
services
from
products
report
form
december
31
2019
current
on
to
2018
our
in
ended
non
the
notes
millions
year
years
marketable
diluted
other
net
income
following
for
comprehensive
provision
21
3
15
4
internal
control
over




```
############################  tf-idf  ##########################
#we're grouping by the author this time
all_tokens <- bind_rows(mutate(tidy_google, author="google"),
                        mutate(tidy_microsoft, author= "microsoft"),
                        mutate(tidy_amazon, author="amazon")
) # after removing all the common words

total_words <- all_tokens %>%
  group_by(author) %>%
  summarize(total=sum(n))

words <- left_join(all_tokens, total_words)

print(words)

ggplot(words, aes(n/total, fill = author))+
  geom_histogram(show.legend=FALSE)+
  xlim(NA, 0.005) +
  facet_wrap(~author, ncol=2, scales="free_y")  ##### left side stands for big business potential
#what do the tails represent?
#answer: extremely common words!
# we are really interested in the not so common words.
```

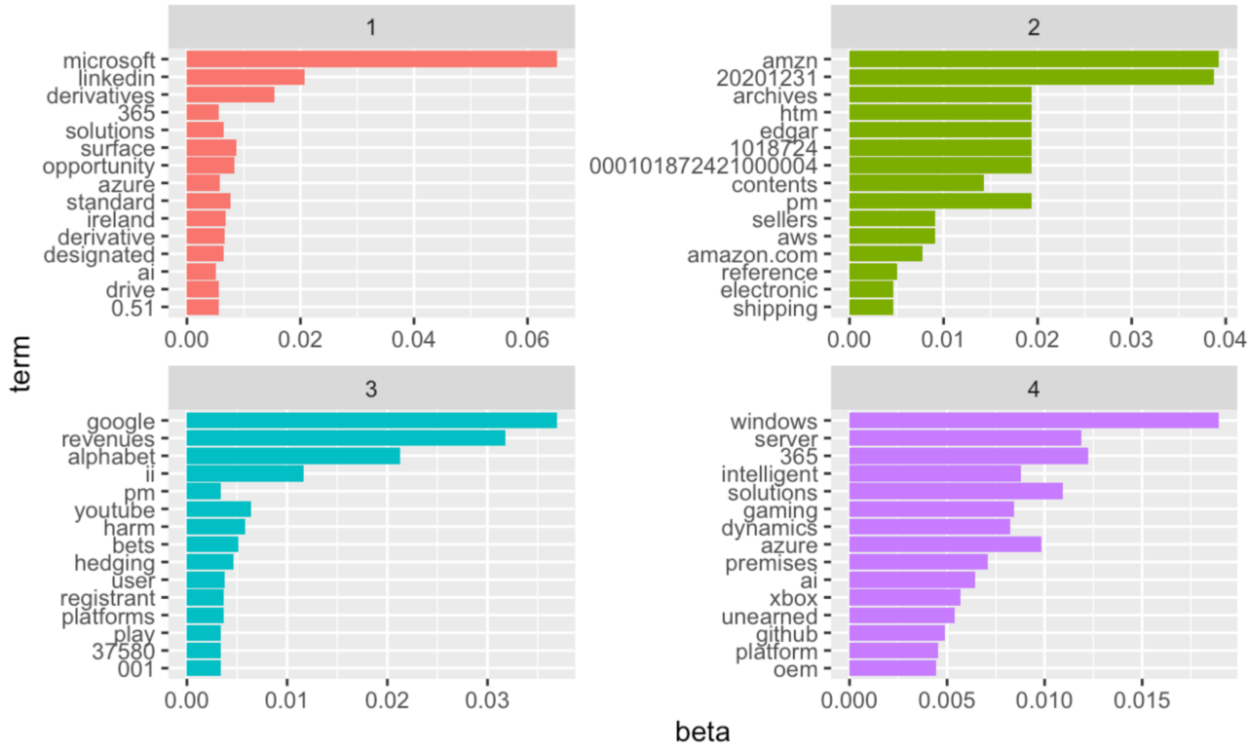
```
#################################   TF_IDF   #################################
company_words <- words %>%
  bind_tf_idf(word, author, n)

company_words # we get all the zeors because we are looking at stop words ... too common

arranged_idf <- company_words %>%
  arrange(desc(tf_idf))
#what can we say about these words?

##############
# looking at the graphical apprach:
company_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(author) %>%
  top_n(15) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=author))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~author, ncol=2, scales="free")+
  coord_flip()
```






```
##################################################################################
################# Latent Dirichlet Allocation algorithm ##############################
##################################################################################
dtm <- words %>% cast_dtm(author,word,n)
ap_lda <- LDA(dtm, k=4, control=list(seed=123))
ap_lda

#now we are looking for the per topic per word probabilities aka. beta
#beta - what is the probability that "this term" will be generated by "this topic"
library(tidytext)
ap_topics <- tidy(ap_lda, matrix="beta")
ap_topics
library(ggplot2)
library(dplyr)
library(tidyr)

top_terms <- ap_topics %>%
  group_by(topic) %>%
  top_n(15, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
top_terms
```

```
#lets plot the term frequencies by topic
top_terms %>%
  mutate(term=reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~topic, scales = "free") +
  coord_flip()

#lets calculate the relative difference between the betas for words in topic 1
#and words in topic 2

beta_spread <- ap_topics %>%
  mutate(topic=paste0("topic", topic)) %>%
  spread(topic, beta) %>%
  filter(topic1>.001 | topic2 >.001) %>%
  mutate(log_rate_google_cloud = log2(topic4/topic3),
         log_rate_amazon_cloud= log2(topic4/topic2)) # practice for more than 2 topics

google_cloud_insights <-
beta_spread[,c("term","log_rate_google_cloud","log_rate_amazon_cloud")] %>%
  filter(log_rate_google_cloud>-3,log_rate_google_cloud<3) %>%
  arrange(desc(log_rate_google_cloud))

amazon_cloud_insights <-
beta_spread[,c("term","log_rate_google_cloud","log_rate_amazon_cloud")] %>%
  filter(log_rate_amazon_cloud>-6,log_rate_amazon_cloud<6) %>%
  arrange(desc(log_rate_amazon_cloud))
```




| | term | log_rate_google_cloud |
|---|---|---|
| 1 | calls | -2.59936564 |
| 2 | commence | -2.43706698 |
| 3 | headcount | -2.39745097 |
| 4 | workplace | -2.33091419 |
| 5 | materials | -2.19546121 |
| 6 | students | -1.94179127 |
| 7 | crisis | -1.79104531 |
| 8 | carbon | -1.73651191 |
| 9 | hedge | -1.64809671 |
| 10 | align | -1.64324331 |
| 11 | derivatives | -1.40870235 |
| 12 | play | -1.36450732 |
| 13 | audiences | -1.34596811 |
| 14 | user | -1.34501767 |
| 15 | 738 | -1.31484640 |
| 16 | sustainability | -1.15349844 |
| 17 | black | -1.10678523 |
| 18 | strong | -1.05828159 |
| 19 | erp | -1.04056706 |

| | term | log_rate_google_cloud |
|---|---|---|
| 1 | questions | 2.80783039 |
| 2 | standard | 2.79838956 |
| 3 | repurchased | 2.77543776 |
| 4 | live | 2.72392810 |
| 5 | transformation | 2.69939763 |
| 6 | hybrid | 2.62513247 |
| 7 | achieve | 2.56809971 |
| 8 | planet | 2.42205850 |
| 9 | productive | 2.39299324 |
| 10 | 2.2 | 2.34762462 |
| 11 | ai | 2.27813518 |
| 12 | alliances | 2.20783574 |
| 13 | consoles | 2.18573731 |
| 14 | learn | 2.14023841 |
| 15 | insights | 2.11717699 |
| 16 | closing | 1.95789582 |
| 17 | canadian | 1.93531385 |
| 18 | ecosystem | 1.76729197 |
| 19 | strive | 1.66853061 |
| 20 | role | 1.57086064 |
| 21 | experiences | 1.53930914 |
| 22 | certificates | 1.52924838 |
| 23 | entertainment | 1.49449095 |
| 24 | metrics | 1.49341656 |
| 25 | community | 1.45028838 |
| 26 | understand | 1.40148647 |
| 27 | oracle | 1.39756588 |
| 28 | platform | 1.32367569 |
| 29 | transforming | 1.31475161 |
| 30 | shipping | 1.26142100 |
| 31 | opportunity | 1.25974091 |
| 32 | guidelines | 1.20760053 |
| 33 | function | 1.20226181 |
| 34 | world's | 1.19514959 |
| 35 | accelerate | 1.13300954 |

| | term | log_rate_google_cloud | log_rate_amazon_cloud |
|---|---|---|---|
| 1 | premises | 157.4152 | 5.19116527 |
| 2 | distinct | 158.4049 | 3.70294961 |
| 3 | methodology | 157.5468 | 3.36280152 |
| 4 | doubtful | 155.7188 | 2.57820806 |
| 5 | 2.13 | 157.7903 | 2.50602439 |
| 6 | virtual | 154.1599 | 2.29896091 |
| 7 | 157 | 156.8165 | 2.19843732 |
| 8 | unearned | 159.3126 | 1.62844184 |
| 9 | indices | 159.0425 | 1.58662741 |
| 10 | percent | 155.5369 | 1.00054777 |
| 11 | enterprises | 156.3588 | 0.61301602 |
| 12 | 15.8 | 155.5056 | 0.36510516 |
| 13 | combined | 156.0691 | 0.32376334 |
| 14 | inventories | 155.7347 | 0.24535770 |
| 15 | producing | 156.4671 | 0.03646300 |
| 16 | lines | 153.6613 | -0.03735544 |
| 17 | database | 154.8120 | -0.36645698 |
| 18 | preparing | 152.1937 | -0.50684915 |
| 19 | shareholders | 152.8448 | -1.07218412 |
| 20 | receivables | 154.8460 | -1.08589050 |
| 21 | subsequent | 154.7024 | -1.48098266 |
| 22 | vendor | 154.8228 | -1.65686791 |
| 23 | grade | 152.3866 | -2.04770811 |
| 24 | selection | 153.4819 | -2.33500389 |
| 25 | deductions | 153.2416 | -2.41443127 |
| 26 | warrant | 152.0251 | -2.55795182 |
| 27 | omnichannel | 151.2322 | -2.65897989 |
| 28 | 411 | 151.5329 | -2.68650478 |
| 29 | entry | 153.6667 | -2.75287606 |
| 30 | absolute | 152.6242 | -3.42569915 |
| 31 | electronic | 146.3717 | -4.50808889 |

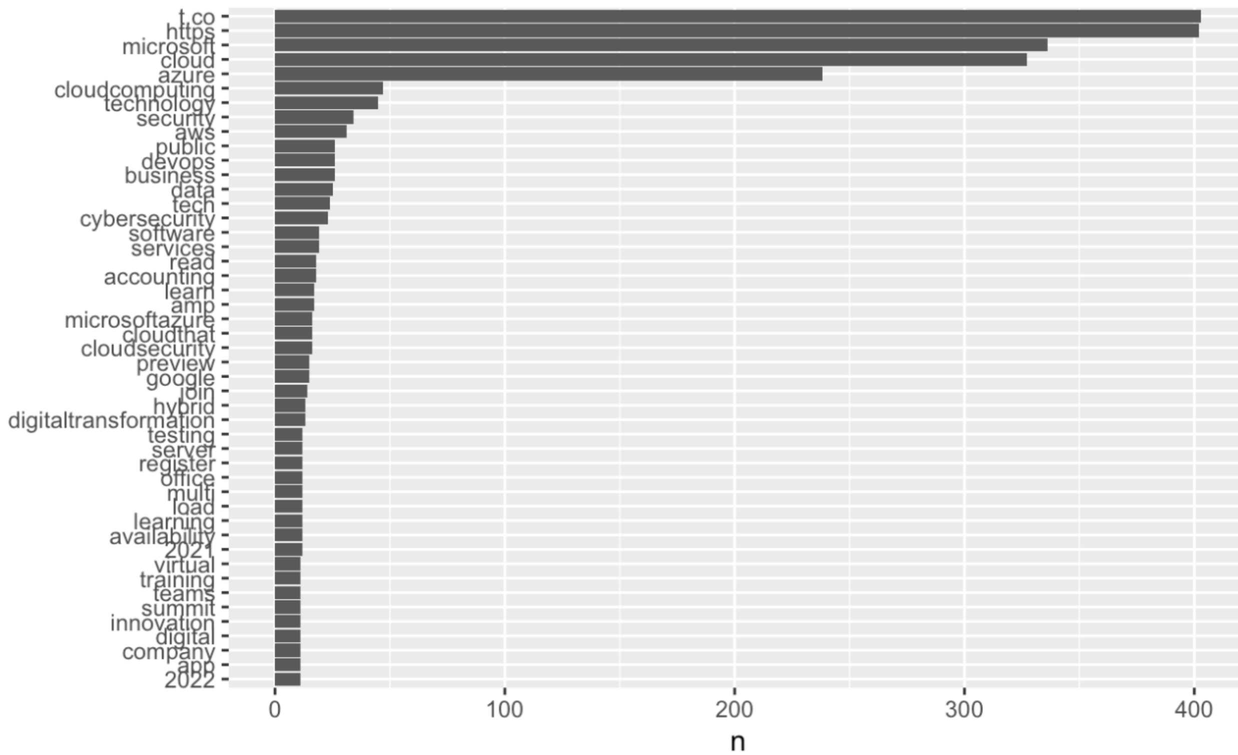
##########################################################################################
**########################  Open source data from Tweeter  #######################**
##########################################################################################

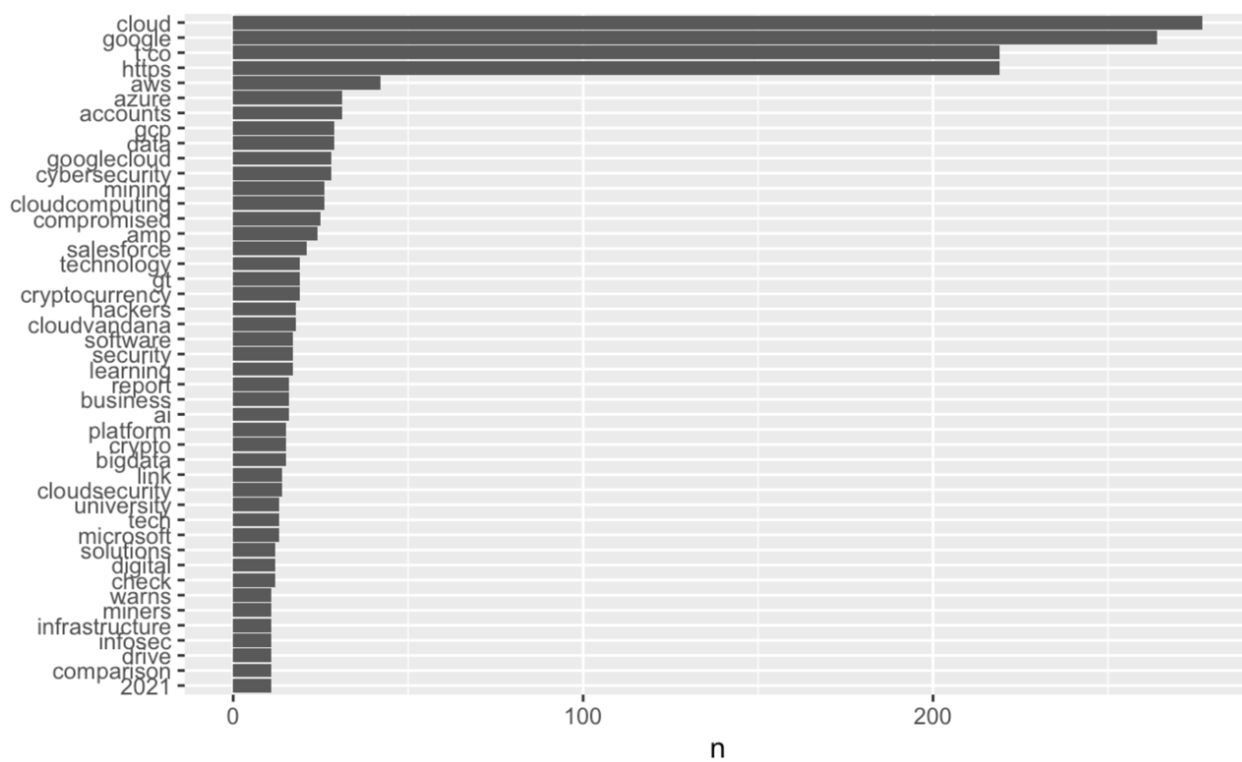
tweet_google <- search_tweets("#google + #cloud", n = 10000, include_rts = FALSE,lang="en")
tweet_amazon <- search_tweets("#amazon + #cloud", n = 20000, include_rts = FALSE,lang="en")
tweet_microsoft <- search_tweets("#microsoft + #cloud", n = 10000, include_rts = FALSE, lang="en")

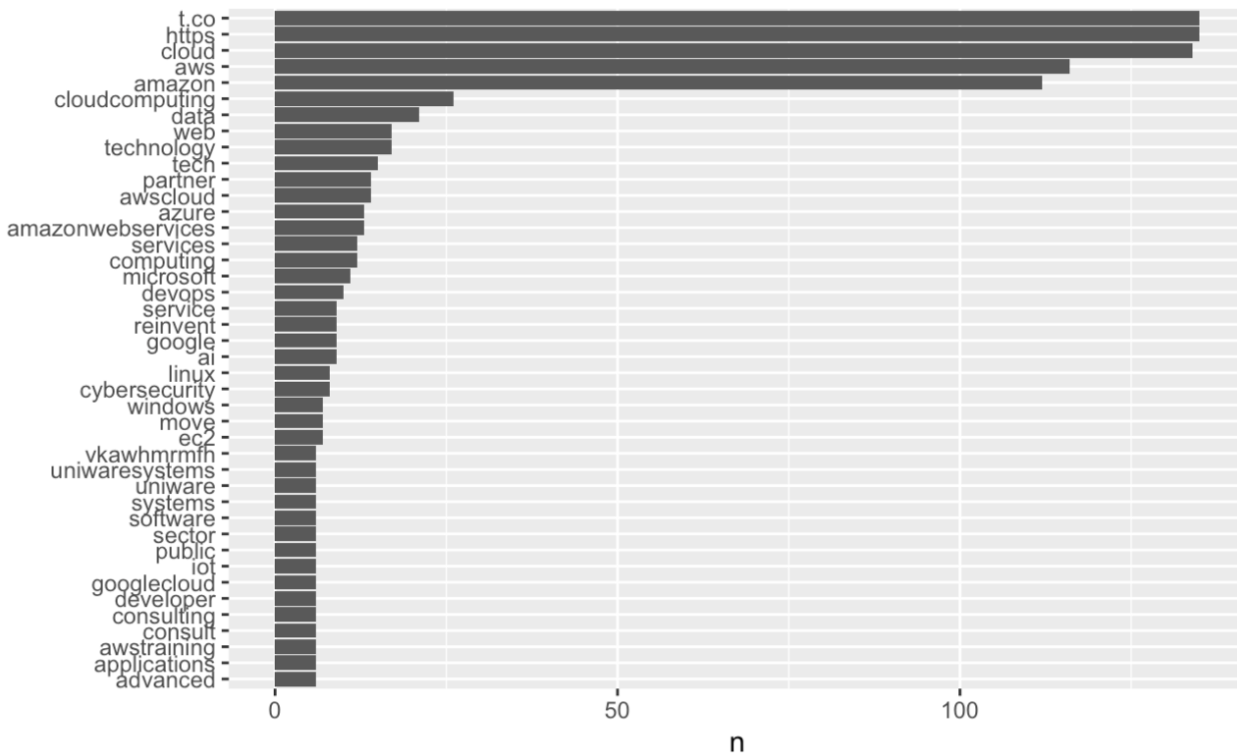
tweet_microsoft_text <- tweet_microsoft$text
tweet_microsoft_df <- data.frame(line=1:276,text=tweet_microsoft_text)

```
tweet_google_text <- tweet_google$text
tweet_google_df <- data.frame(line=1:173,text=tweet_google_text)

tweet_amazon_text <- tweet_amazon$text
tweet_amazon_df <- data.frame(line=1:93,text=tweet_amazon_text)

########################   Building tidy formate   ##########################
tidy_tweet_microsoft <- tweet_microsoft_df %>%
  unnest_tokens(word,text) %>%
  anti_join(stop_words) %>%
  count(word,sort=TRUE)
tidy_tweet_google <- tweet_google_df %>%
  unnest_tokens(word,text) %>%
  anti_join(stop_words) %>%
  count(word,sort=TRUE)
tidy_tweet_amazon <- tweet_amazon_df %>%
  unnest_tokens(word,text) %>%
  anti_join(stop_words) %>%
  count(word,sort=TRUE)

####################   Building term frequency histogram   ###################
microsoft_freq_hist <- tidy_tweet_microsoft %>%
  mutate(word=reorder(word, n)) %>%
  filter(n>10) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(microsoft_freq_hist)

google_freq_hist <- tidy_tweet_google %>%
  mutate(word=reorder(word, n)) %>%
  filter(n>10) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(google_freq_hist)

amazon_freq_hist <- tidy_tweet_amazon %>%
  mutate(word=reorder(word, n)) %>%
  filter(n>5) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
```

```
  coord_flip()
print(amazon_freq_hist)
```

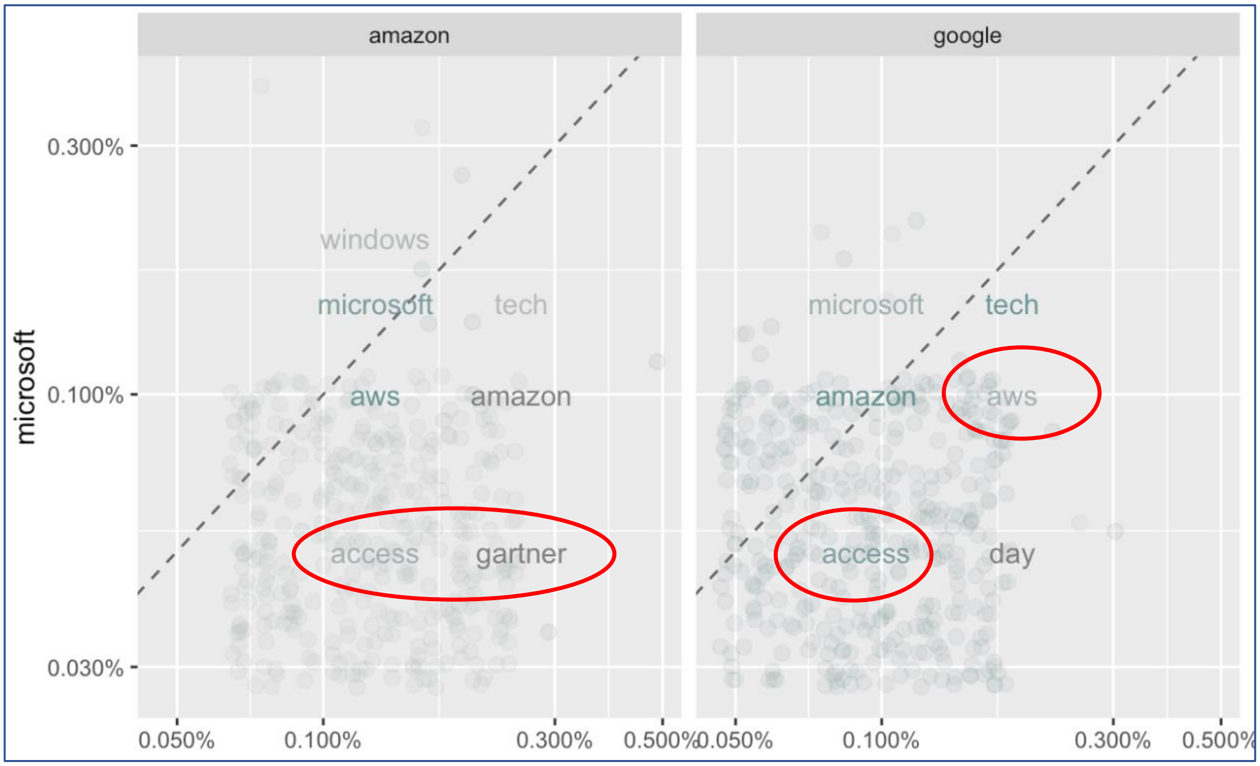




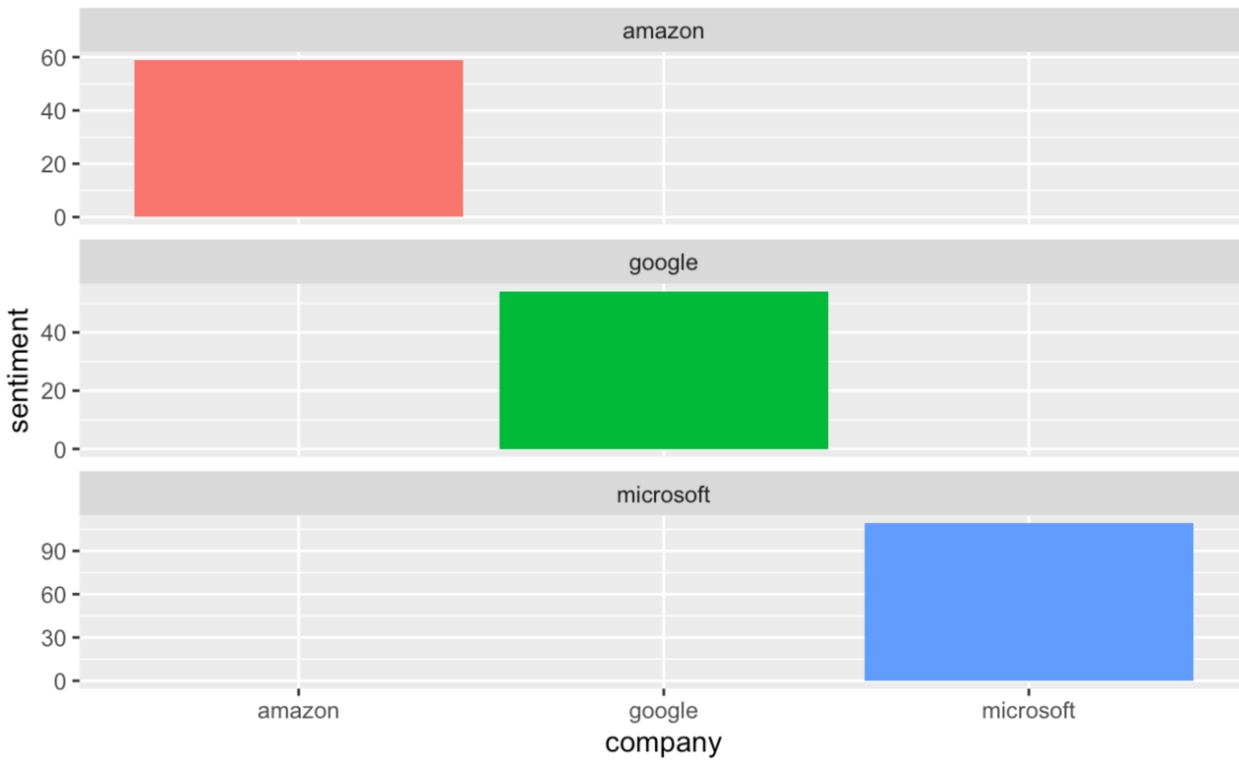

```
##############################   correlograms   ##############################
frequency <- bind_rows(mutate(tidy_tweet_microsoft, author="microsoft"),
                       mutate(tidy_tweet_google, author= "google"),
                       mutate(tidy_tweet_amazon, author="amazon")
)%>%#closing bind_rows
  mutate(word=str_extract(word, "[a-z']+")) %>%
  filter(!nchar(word)==1) %>%
  filter(!nchar(word)==2) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n/sum(n))%>%
  select(-n) %>%
  spread(author, proportion) %>%
  gather(author, proportion, `google`, `amazon`)

ggplot(frequency, aes(x=proportion, y=`microsoft`,
                      color = abs(`microsoft`- proportion)))+
  geom_abline(color="grey40", lty=2)+
  geom_jitter(alpha=.1, size=2.5, width=0.3, height=0.3)+
  geom_text(aes(label=word), check_overlap = TRUE, vjust=1.5) +
  scale_x_log10(labels = percent_format())+
  scale_y_log10(labels= percent_format())+
  scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+
  facet_wrap(~author, ncol=2)+
  theme(legend.position = "none")+
```

labs(y= "microsoft", x=NULL)




```
##################################################################
# sentiment analysis: comparison in three in nrc
##################################################################
microsoft_nrc <- tidy_tweet_microsoft %>%
  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) %>%
  mutate(company="microsoft")

google_nrc <- tidy_tweet_google %>%
  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) %>%
  mutate(company="google")

amazon_nrc <- tidy_tweet_amazon %>%
  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) %>%
  mutate(company="amazon")

bind_rows(microsoft_nrc,google_nrc,amazon_nrc) %>%
  ggplot(aes(company, sentiment, fill=company))+
  geom_col(show.legend=FALSE)+
  facet_wrap(~company, ncol =1, scales= "free_y")
```

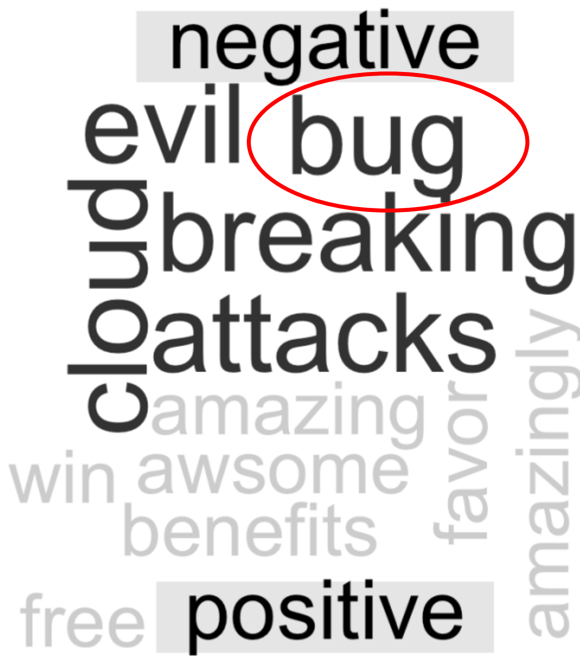



```
##################################################################
######## Most common positive and negative words #############
##################################################################
# wordcloud
library(reshape2)
install.packages("wordcloud")
library(wordcloud)
tidy_tweet_microsoft %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("gray20", "gray80"), max.words = 500)
```

```
tidy_tweet_google %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("gray20", "gray80"), max.words = 800)

tidy_tweet_amazon %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("gray20", "gray80"), max.words = 700)
```

# Microsoft


# Google



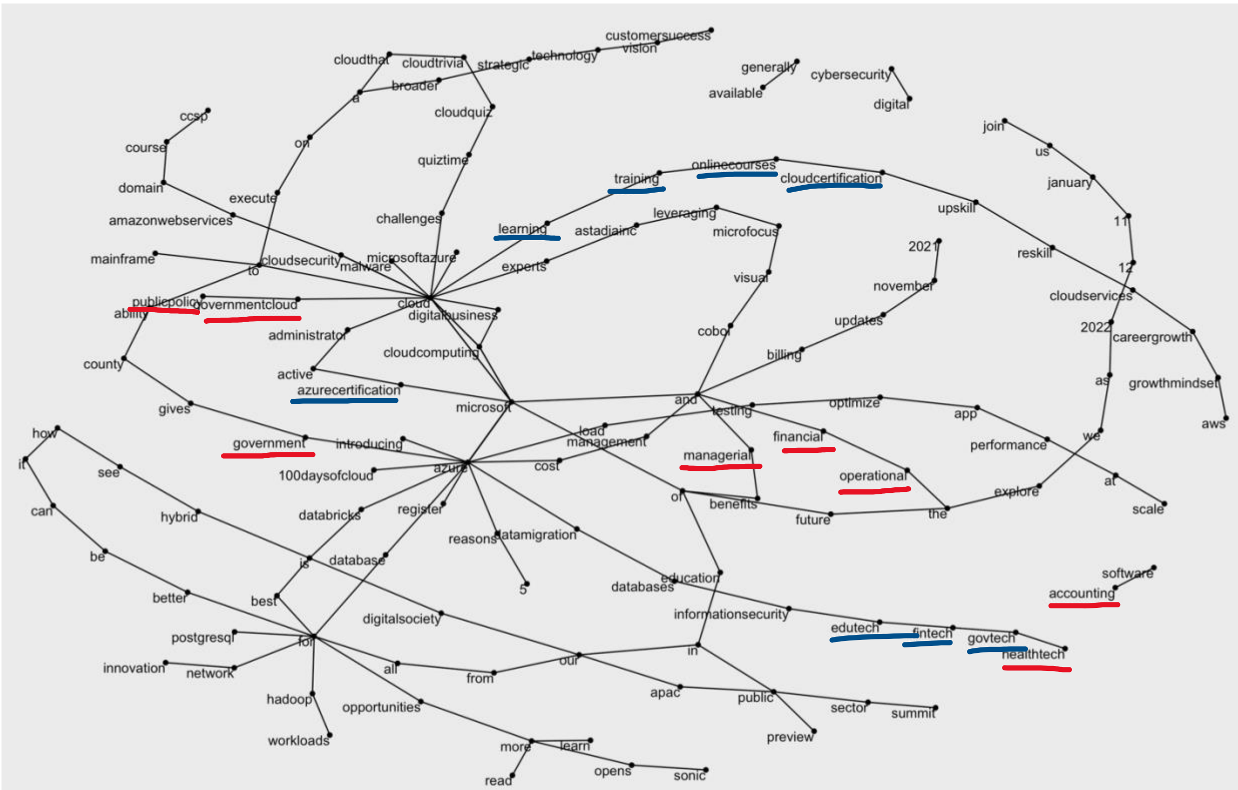

# Amazon


```
############################   Ngrams   ##########################
# microsoft
microsoft_quadrograms <- tweet_microsoft_df %>%
  unnest_tokens(quadrograms,text,token="ngrams",n=4) %>%
  separate(quadrograms,c("word1","word2","word3","word4"),sep = " ") %>%
  filter(!word1 %in% stop_words) %>%
  filter(!word2 %in% stop_words) %>%
  filter(!word3 %in% stop_words) %>%
  filter(!word4 %in% stop_words)
```

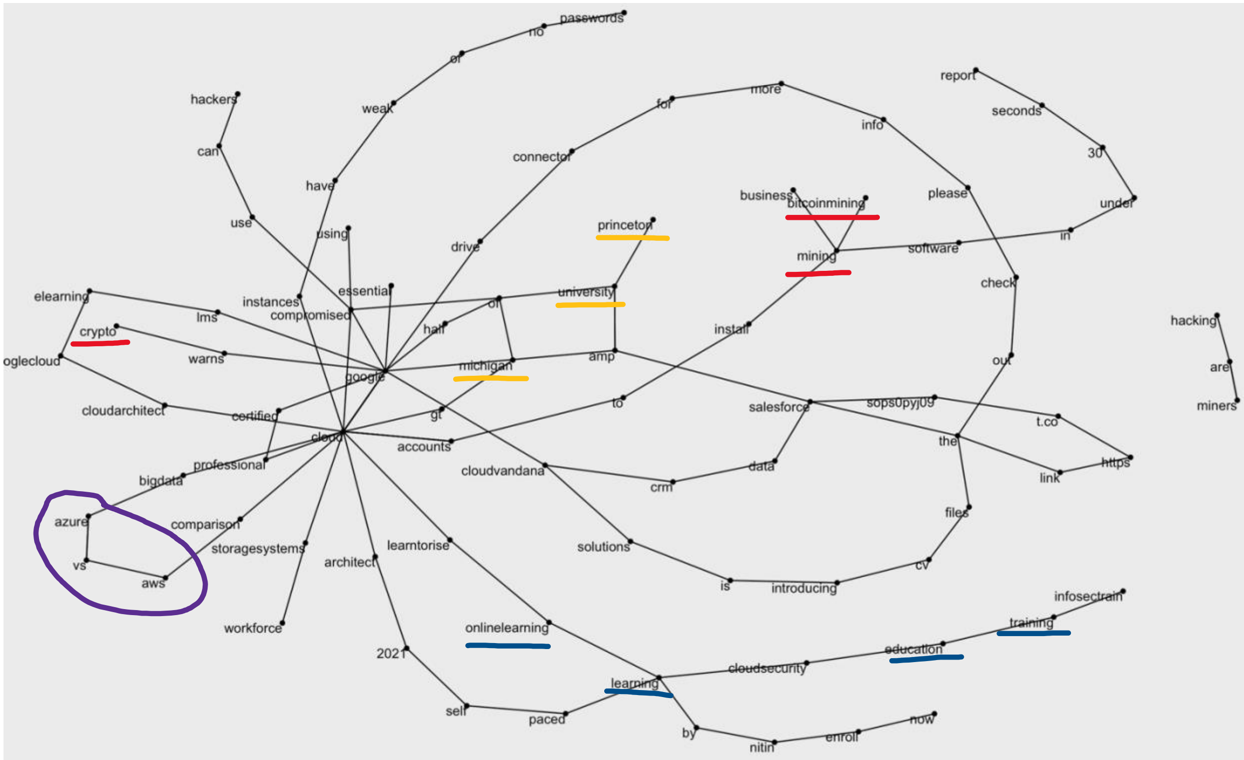
```
microsoft_quadrograms_counts <- microsoft_quadrograms %>%
  count(word1,word2,word3,word4,sort = TRUE)

microsoft_quadrogram_graph <- microsoft_quadrograms_counts %>%
  filter(n>3) %>%
  graph_from_data_frame()

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

# google
google_quadrograms <- tweet_google_df %>%
  unnest_tokens(quadrograms,text,token="ngrams",n=4) %>%
  separate(quadrograms,c("word1","word2","word3","word4"),sep = " ") %>%
  filter(!word1 %in% stop_words) %>%
  filter(!word2 %in% stop_words) %>%
  filter(!word3 %in% stop_words) %>%
  filter(!word4 %in% stop_words)

google_quadrograms_counts <- google_quadrograms %>%
  count(word1,word2,word3,word4,sort = TRUE)

google_quadrogram_graph <- google_quadrograms_counts %>%
  filter(n>3) %>%
  graph_from_data_frame()

ggraph(google_quadrogram_graph,layout="fr") +
  geom_edge_link() +
  geom_node_point() +
  geom_node_text(aes(label=name),vjust=1,hjust=1)
# amazon
amazon_quadrograms <- tweet_amazon_df %>%
  unnest_tokens(quadrograms,text,token="ngrams",n=4) %>%
  separate(quadrograms,c("word1","word2","word3","word4"),sep = " ") %>%
  filter(!word1 %in% stop_words) %>%
  filter(!word2 %in% stop_words) %>%
  filter(!word3 %in% stop_words) %>%
  filter(!word4 %in% stop_words)

amazon_quadrograms_counts <- amazon_quadrograms %>%
  count(word1,word2,word3,word4,sort = TRUE)

amazon_quadrogram_graph <- amazon_quadrograms_counts %>%
```

```
  filter(n>1) %>%
  graph_from_data_frame()

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

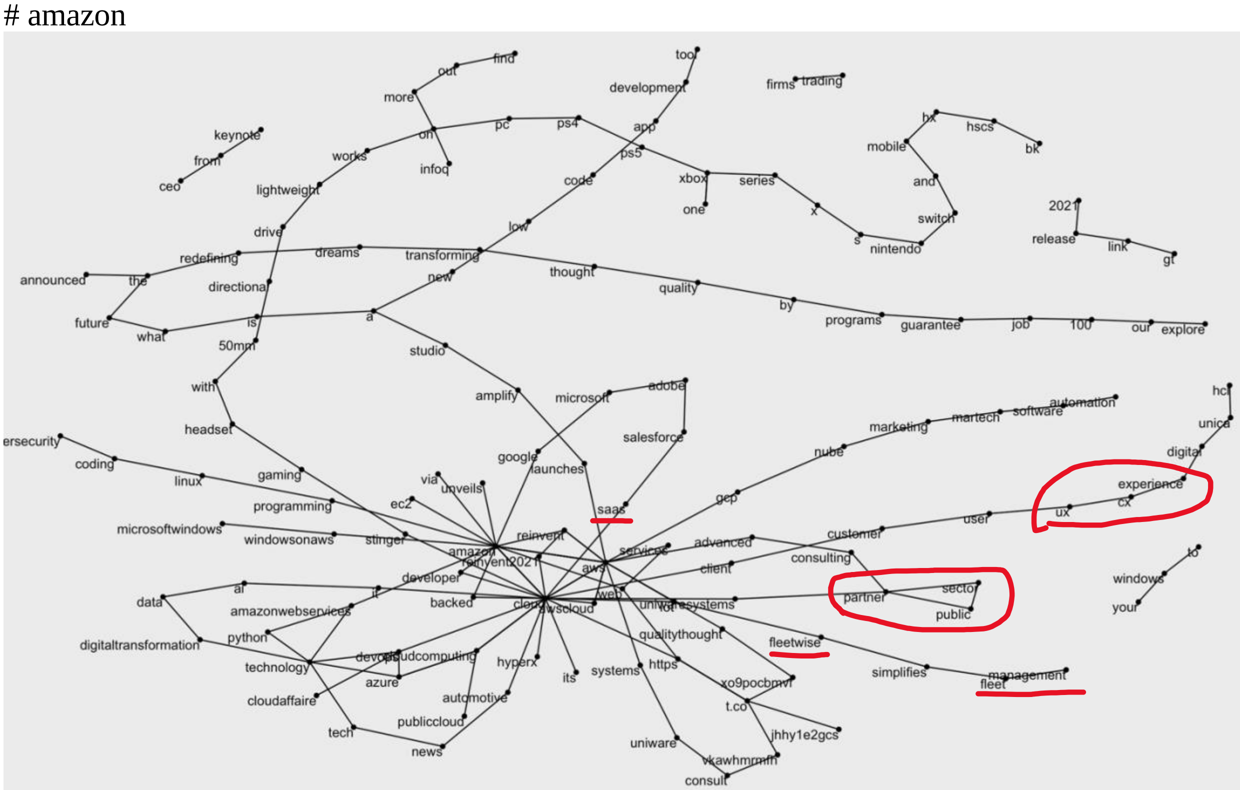
# microsoft




# google




# amazon

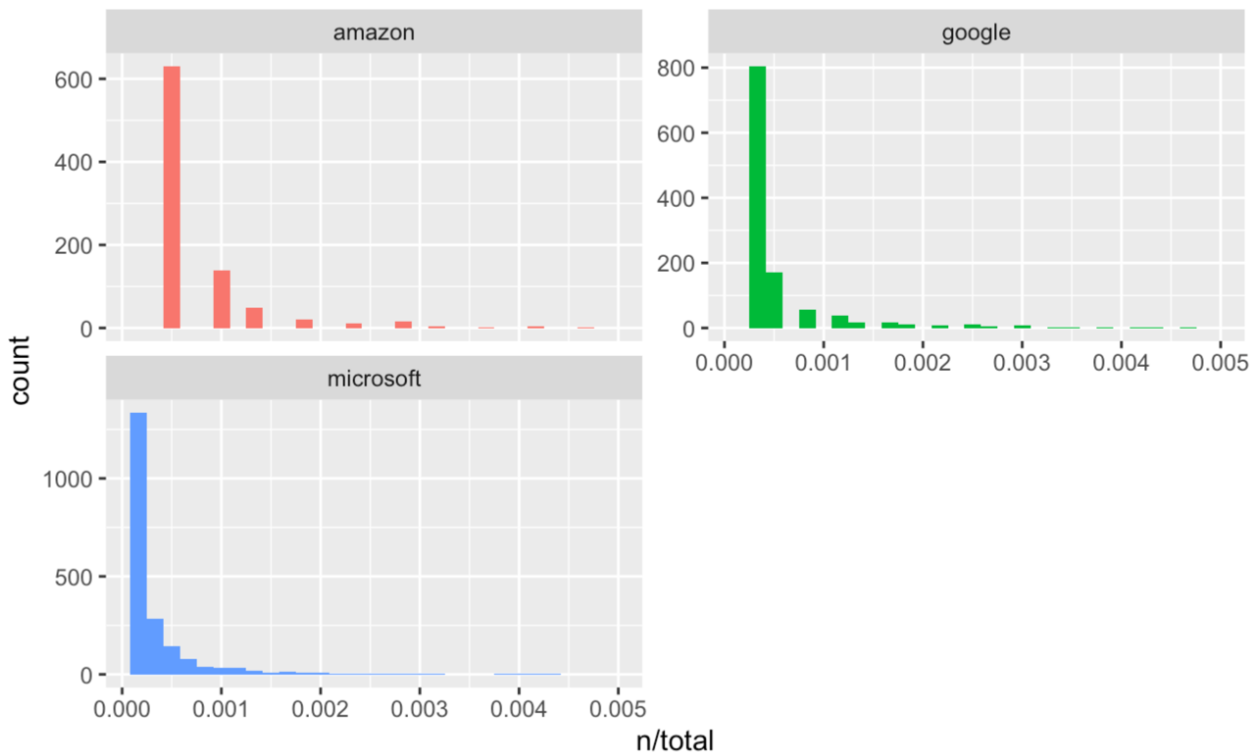



#############################   tf-idf   ###########################

#we're grouping by the country this time

all_tokens <- bind_rows(mutate(tidy_tweet_microsoft, author="microsoft"),

```
                                    mutate(tidy_tweet_google
                                           , author= "google"),
                                    mutate(tidy_tweet_amazon, author="amazon")
)
comparison_tokens <- all_tokens %>%
  count(author, word, sort=TRUE) %>%  #### count on the "country" level, each country each
"doc"
  ungroup()

total_words <- all_tokens %>%
  group_by(author) %>%
  summarize(total=sum(n))

words <- left_join(all_tokens, total_words)

print(words)

ggplot(words, aes(n/total, fill = author))+
  geom_histogram(show.legend=FALSE)+
  xlim(NA, 0.005) +
  facet_wrap(~author, ncol=2, scales="free_y")
```




```
######################################################
################## TF_IDF ############################
######################################################
```

```
company_words <- words %>%
  bind_tf_idf(word, author, n)  # why is "country" here?

company_words # we get all the zeors because we are looking at stop words ... too common

arranged_idf <- company_words %>%
  arrange(desc(tf_idf))
#what can we say about these words?

#############
# looking at the graphical apprach:
company_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(author) %>%
  top_n(10) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=author))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~author, ncol=2, scales="free")+
  coord_flip()
```