Hult International Business School

Text Analytics – Prof. Thomas Kurnicki

Mauricio Marcon Teles

MSBA – 2019/2020

Part 1: Report

Business insight: Franchisor invests and creates a training program for the franchisees and their teams.

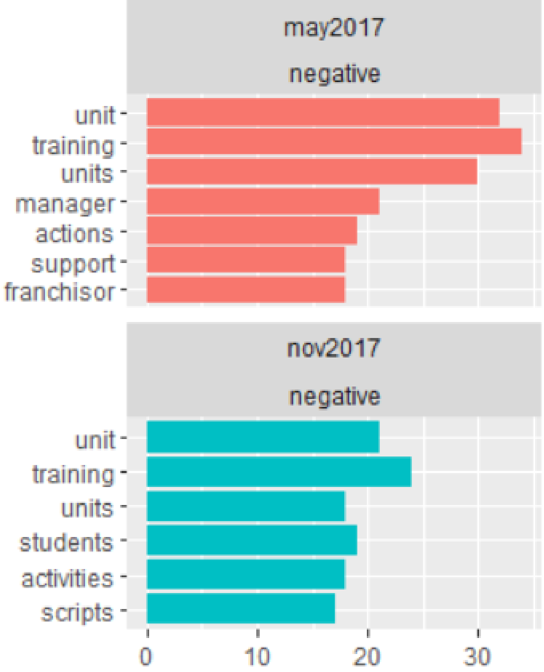
The data was collected in 2017, among the franchisees of Supera (an education franchise related to brain training), though two surveys (in May and November) to understand their needs and collect suggestions to the franchisor, aiming the success of both.

The survey had four sections representing each department, with tree quantitative (scores from 1 to 10) questions and two qualitative (open text). The document was translated from Brazilian Portuguese to English by Google functions.

The scores were averaged by department and considered as negative (below average), neutral (average), and positive (above average).

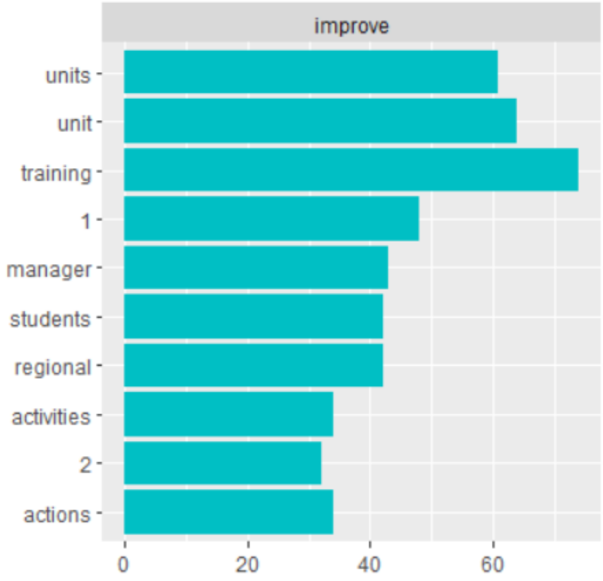
By considering this classification and the tokenization of the qualitative answers, it is possible to see that the franchisee's needs changed from May to November. Between the surveys, the focus of the moved from the franchisor and his team (manager, actions, support) to the product (activities and class scripts) and clients (students).

Fig. 1 – Histogram of most used words in negative responses



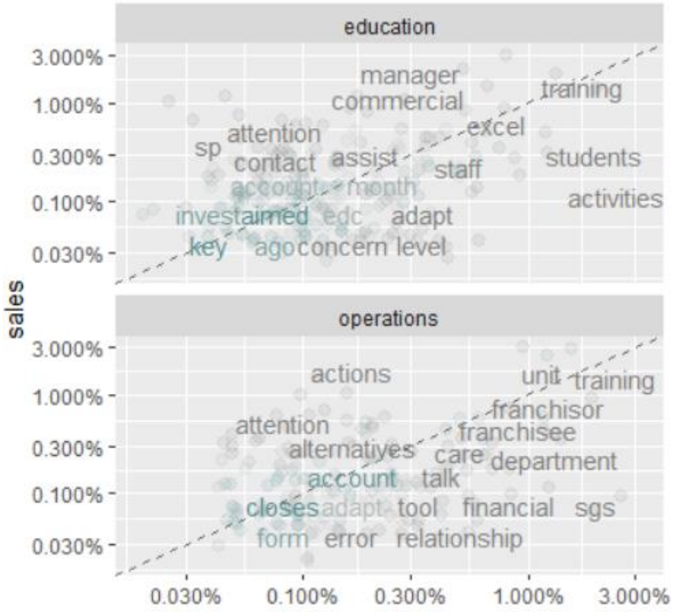
However, the word training appears in both surveys. By the graph below, it is possible to understand the reason why it counts so often. It is the most used word in the space designated to express their suggestions and requests.

Fig. 2 – Histogram of most used words in improvements



The correlogram below shows how training is so crucial to the franchisees. They use the word broadly among the operations, education, and sales department sections of the survey. These departments represent the core of any franchised unit, which means that the managers are looking for resources to improve their deliveries and, thus, their operational results.

Fig. 3 – Correlogram between the departments of Sales (axis y) and Education and Operation (axis x)



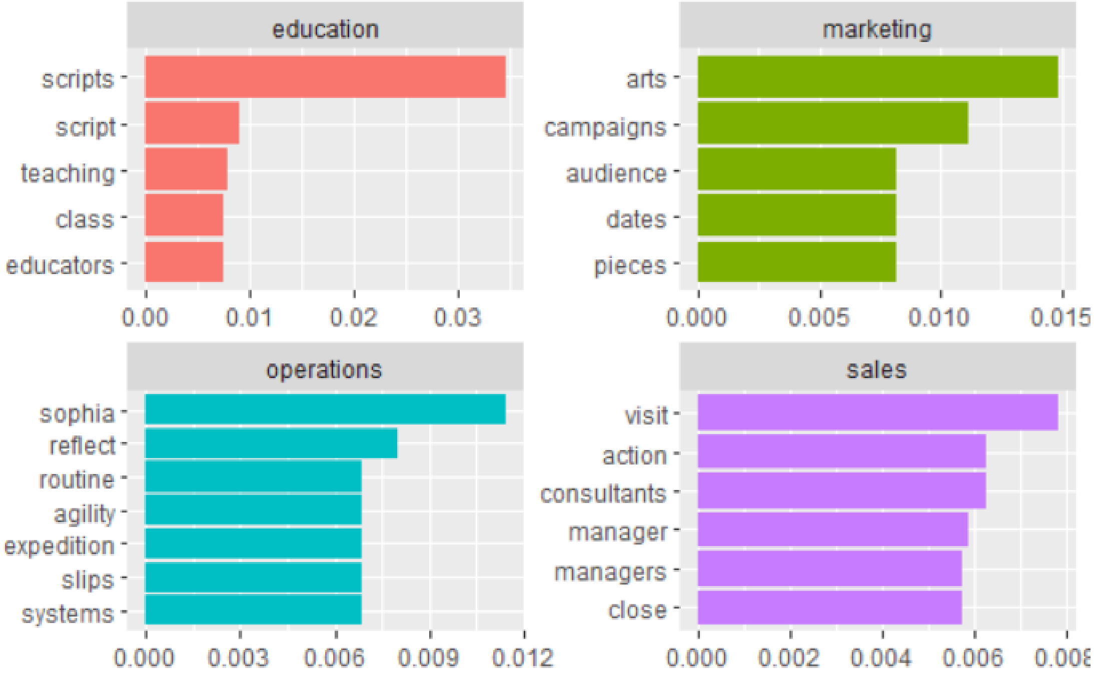
Considering the following TF-IDF statistics, it is possible to identify which kind of training the franchisees are searching for.

From the education department, their focus is on training their educators better to facilitate the sessions (teaching skills). It demands to understand the scripts, organize the sessions, and assure the students are achieving their individual goals.

Sophia, the most scored word from operations, is the CRM / ERP software all the franchised units must use to manage clients, sales, education process, students, and it is required by the franchisor, who also uses its admin access to use the data. Most managers do not know how to use the software properly, which clarifies the need for training.

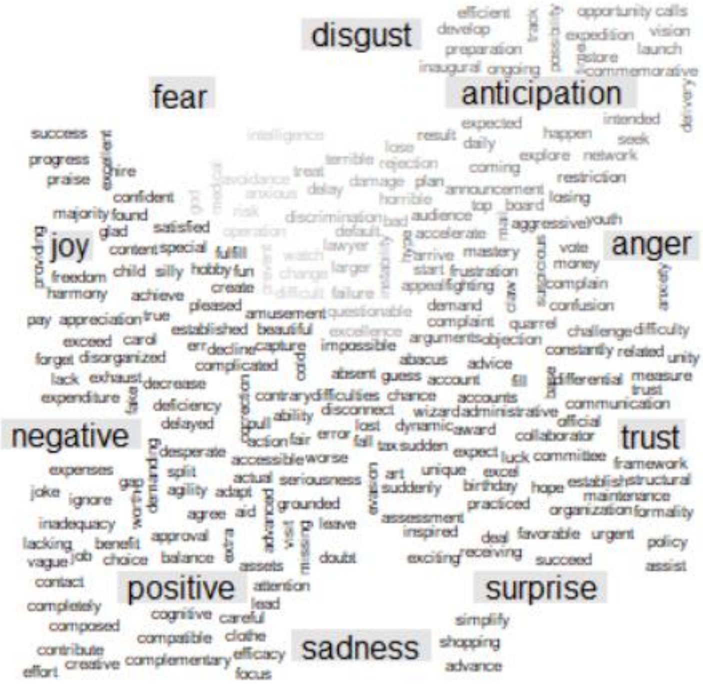
Consultants are sellers, those who contact the lead to invite him for a visit and execute local actions to collect new leads, perform local marketing, and create awareness of the product and brand. The turnover of these professionals is high, and the units often need to train new employees.

Fig. 4 – TF-IDF showing the most relevant terms per department



How trustful are this information and how to measure if the creation of a training program will succeed? The sentiment analysis of the words used in the survey shows that while answering the survey, the franchisees' answers relate to positive, trust, and joy. Moreover, some words express their sentiment of anticipation. Words like develop, preparation, and happen may show that they are not only requesting franchisor’s support but their willingness to engage in anything that may support their growth.

Fig. 5 – Comparison cloud of most frequent words



Business recommendation

Create a continuous improvement program mixing online and in-person training.

Short term plan

- Publish short videos with guidance with main doubts:
  - Operations: main features of the software, organized by department, where the franchisees can easily find guidance on how to use the tool in a focused and fast way to learn.
  - Education: how to execute some class activities and build community among the students of each group
  - Sales: elevator pitch, lead's management, and negotiation
- Designate a person as the focal point to guide the use of the video library and able to answer quick questions.
- Coordinate online sessions with specialists to answer more extended and more specific questions, a moment when the teams will be able to share experiences and add personal value to the franchise.

Long term plan

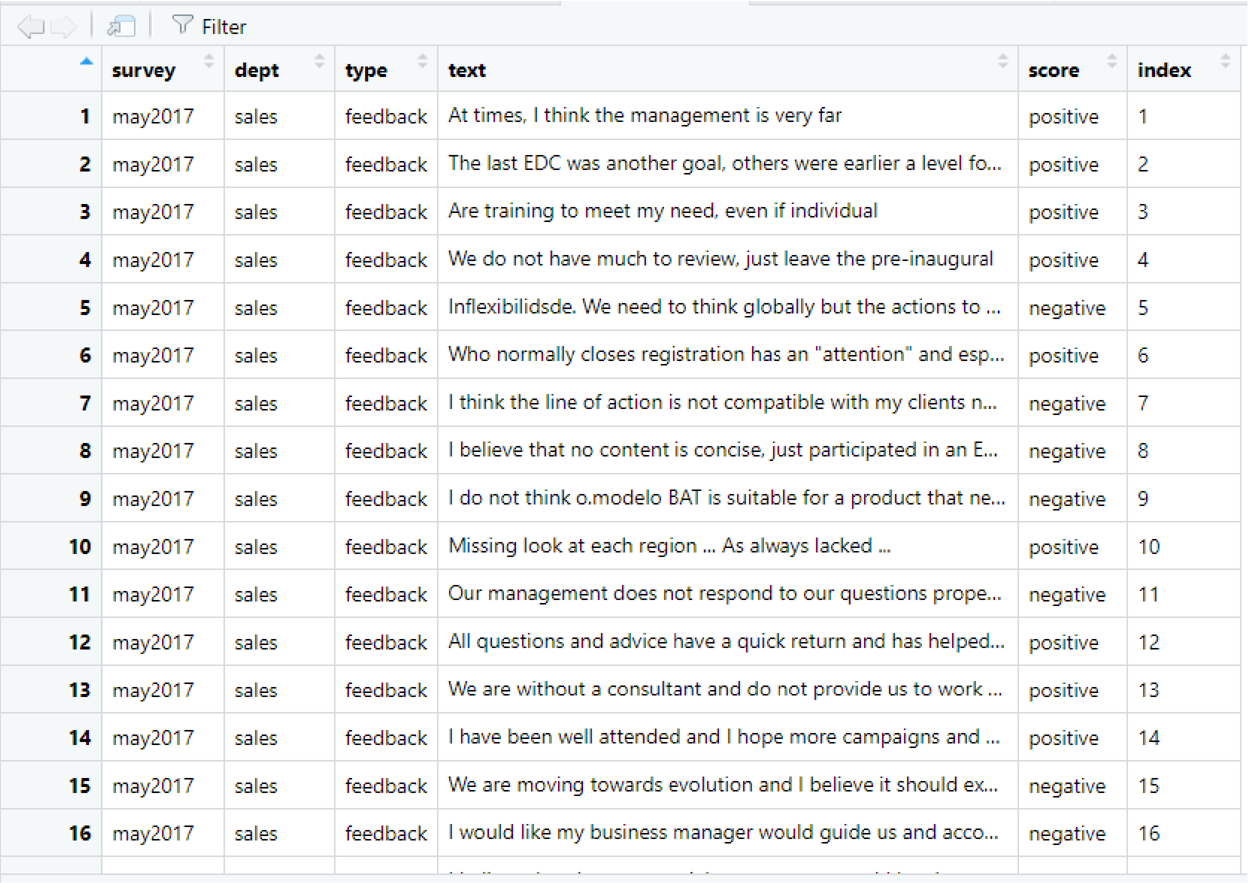
- Design compelling manuals and cheat sheets with main procedures and workflows

- Execute in-person training regionally. Prepare a group of specialists able to travel and organize tree days of training in the main cities of each region. It will be cost-effective to the franchisees and allow the franchisor to understand the regional market better.
- National conference. Once a year, execute a conference with the franchisees to share the best practices, learnings, insights, and individual success. Bring them together will foster union and skills sharing. Moreover, it will strengthen the belong feeling and open space to announce the next steps, validate ideas, and share the main achievements.

Part 2: Code and Output

Import data

```r
# open the data frame
supera_may <- read_excel("Survey Supera.xlsx", sheet = "2017may", col_names = TRUE)
supera_nov <- read_excel("Survey Supera.xlsx", sheet = "2017nov", col_names = TRUE)

###########  F U N C T I O N  ###########
#  split_answers
#  organize the data to include "location"
split_answers <- function(df_in, df_out, survey, dept, type, avg_score){
  df_out <- data.frame()
  for(i in 1:nrow(df_in)){
    if (!is.na(df_in[i, paste(dept,type,sep="_")])){
      df_out[i, "survey"] <- survey
      df_out[i, "dept"]   <- dept
      df_out[i, "type"]   <- type
      df_out[i, "text"]   <- supera_may[i, paste(dept,type,sep="_")]
      if (is.na(df_in[i, paste(dept,"score",sep="_")])) {
        df_out[i, "score"] <- "neutral"
      }else{
        if (df_in[i, paste(dept,"score",sep="_")] >= avg_score) {
          df_out[i, "score"] <- "positive"
        }else{
          df_out[i, "score"] <- "negative"
        }
      }
    }
  }
  df_out <- na.omit(df_out)
  return(df_out)
}


####  D A T A   P R E P A R A T I O N  ####
may_sales_avg <- mean(supera_may$sales_score, na.rm = TRUE)
may_education_avg <- mean(supera_may$education_score, na.rm = TRUE)
may_marketing_avg <- mean(supera_may$marketing_score, na.rm = TRUE)
may_operations_avg <- mean(supera_may$operations_score, na.rm = TRUE)

MSF <- split_answers(supera_may, SF, "may2017", "sales", "feedback", may_sales_avg)
MSI <- split_answers(supera_may, SF, "may2017", "sales", "improve", may_sales_avg)
MEF <- split_answers(supera_may, SF, "may2017", "education", "feedback", may_education_avg)
MEI <- split_answers(supera_may, SF, "may2017", "education", "improve", may_education_avg)
MMF <- split_answers(supera_may, SF, "may2017", "marketing", "feedback", may_marketing_avg)
MMI <- split_answers(supera_may, SF, "may2017", "marketing", "improve", may_marketing_avg)
MOF <- split_answers(supera_may, SF, "may2017", "operations", "feedback", may_operations_avg)
MOI <- split_answers(supera_may, SF, "may2017", "operations", "improve", may_operations_avg)

nov_sales_avg <- mean(supera_nov$sales_score, na.rm = TRUE)
nov_education_avg <- mean(supera_nov$education_score, na.rm = TRUE)
nov_marketing_avg <- mean(supera_nov$marketing_score, na.rm = TRUE)
nov_operations_avg <- mean(supera_nov$operations_score, na.rm = TRUE)

NSF <- split_answers(supera_nov, SF, "nov2017", "sales", "feedback", nov_sales_avg)
NSI <- split_answers(supera_nov, SF, "nov2017", "sales", "improve", nov_sales_avg)
NEF <- split_answers(supera_nov, SF, "nov2017", "education", "feedback", nov_education_avg)
NEI <- split_answers(supera_nov, SF, "nov2017", "education", "improve", nov_education_avg)
NMF <- split_answers(supera_nov, SF, "nov2017", "marketing", "feedback", nov_marketing_avg)
NMI <- split_answers(supera_nov, SF, "nov2017", "marketing", "improve", nov_marketing_avg)
NOF <- split_answers(supera_nov, SF, "nov2017", "operations", "feedback", nov_operations_avg)
NOI <- split_answers(supera_nov, SF, "nov2017", "operations", "improve", nov_operations_avg)

all_surveys <- bind_rows(MSF, MSI, MEF, MEI, MMF, MMI, MOF, MOI,
                         NSF, NSI, NEF, NEI, NMF, NMI, NOF, NOI)  %>%
               mutate(index=row_number())
```

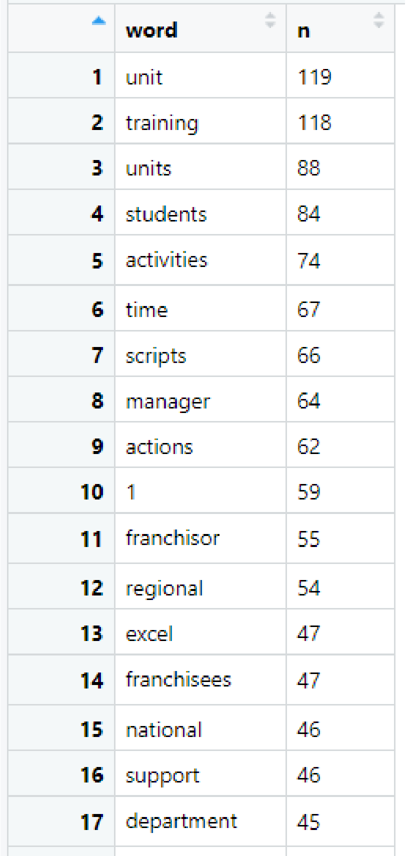


	survey	dept	type	text	score	index
1	may2017	sales	feedback	At times, I think the management is very far	positive	1
2	may2017	sales	feedback	The last EDC was another goal, others were earlier a level fo...	positive	2
3	may2017	sales	feedback	Are training to meet my need, even if individual	positive	3
4	may2017	sales	feedback	We do not have much to review, just leave the pre-inaugural	positive	4
5	may2017	sales	feedback	Inflexibilidsde. We need to think globally but the actions to ...	negative	5
6	may2017	sales	feedback	Who normally closes registration has an "attention" and esp...	positive	6
7	may2017	sales	feedback	I think the line of action is not compatible with my clients n...	negative	7
8	may2017	sales	feedback	I believe that no content is concise, just participated in an E...	negative	8
9	may2017	sales	feedback	I do not think o.modelo BAT is suitable for a product that ne...	negative	9
10	may2017	sales	feedback	Missing look at each region ... As always lacked ...	positive	10
11	may2017	sales	feedback	Our management does not respond to our questions prope...	negative	11
12	may2017	sales	feedback	All questions and advice have a quick return and has helped...	positive	12
13	may2017	sales	feedback	We are without a consultant and do not provide us to work ...	positive	13
14	may2017	sales	feedback	I have been well attended and I hope more campaigns and ...	positive	14
15	may2017	sales	feedback	We are moving towards evolution and I believe it should ex...	negative	15
16	may2017	sales	feedback	I would like my business manager would guide us and acco...	negative	16

Tokenization

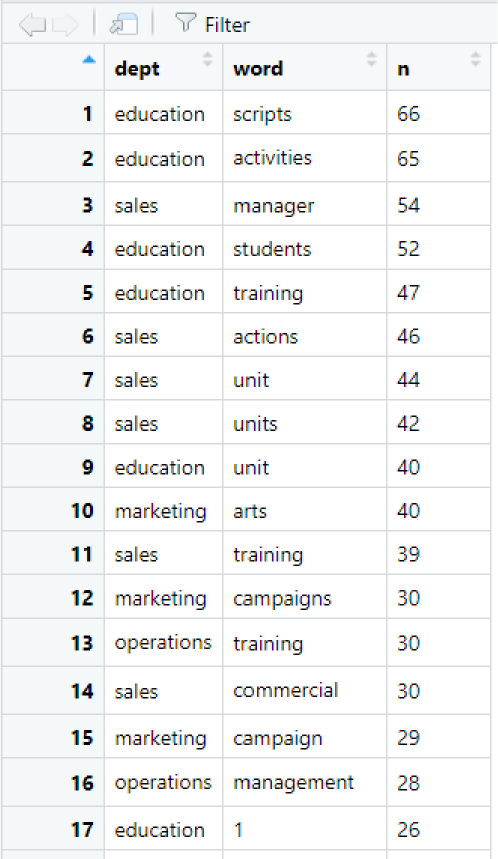
```
####  T O K E N I Z A T I O N  ####
all_tokens <- all_surveys %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>% #here's where we remove tokens
  count(word, sort=TRUE)

dept_tokens <- all_surveys %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>% #here's where we remove tokens
  count(dept, word, sort=TRUE)

type_tokens <- all_surveys %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>% #here's where we remove tokens
  count(type, word, sort=TRUE)

survey_tokens <- all_surveys %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>% #here's where we remove tokens
  count(survey, word, sort=TRUE)

score_tokens <- all_surveys %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>% #here's where we remove tokens
  count(score, word, sort=TRUE)
```

All tokens



	word	n
1	unit	119
2	training	118
3	units	88
4	students	84
5	activities	74
6	time	67
7	scripts	66
8	manager	64
9	actions	62
10	1	59
11	franchisor	55
12	regional	54
13	excel	47
14	franchisees	47
15	national	46
16	support	46
17	department	45

Dept tokens

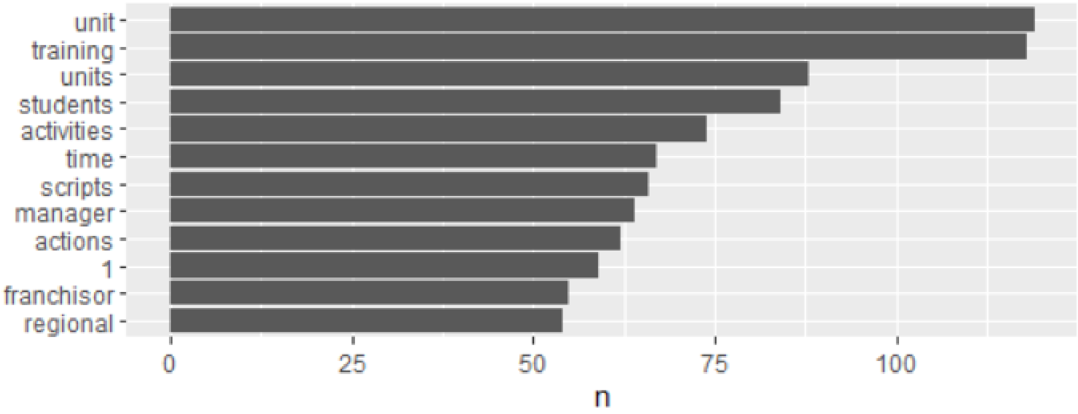


	dept	word	n
1	education	scripts	66
2	education	activities	65
3	sales	manager	54
4	education	students	52
5	education	training	47
6	sales	actions	46
7	sales	unit	44
8	sales	units	42
9	education	unit	40
10	marketing	arts	40
11	sales	training	39
12	marketing	campaigns	30
13	operations	training	30
14	sales	commercial	30
15	marketing	campaign	29
16	operations	management	28
17	education	1	26

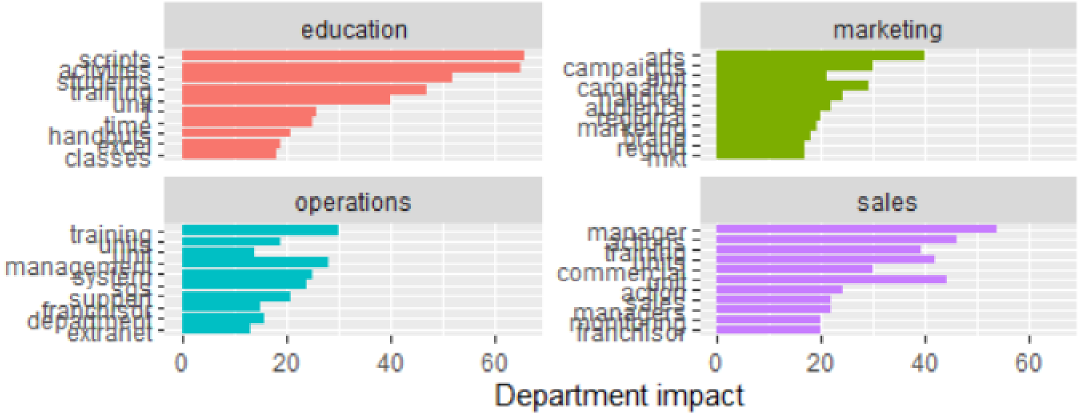
Histogram analysis

```
####  H I S T O G R A M S  ####
freq_hist <- all_tokens %>%
  filter(n > 50) %>%
  mutate(word=reorder(word,n)) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(freq_hist)

dept_tokens %>%
  group_by(dept) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(word, n, fill=dept)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~dept, scales = "free_y")+
  labs(y="Department impact", x=NULL)+
  coord_flip()
```

freq_hist:

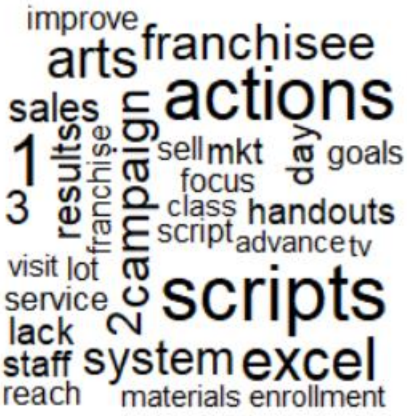


Dept_tokens histogram:



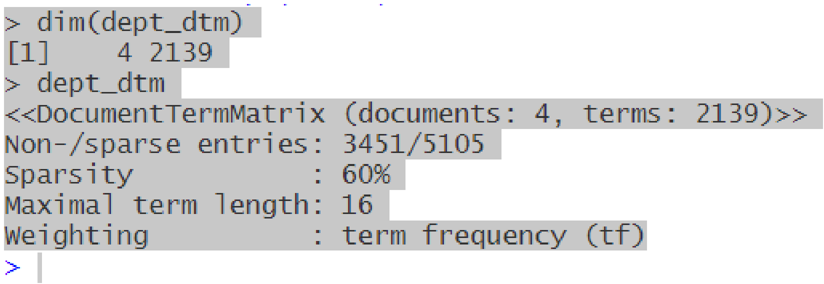
Wordcloud

```
####  W O R D C L O U D  ####
all_tokens %>%
  with(wordcloud(word, n, max.words = 100))
```



DTM Analysis

```
#####  D T M   A NA L Y S I S  #####
surveys_dtm <- all_surveys %>%
  unnest_tokens(word, text) %>%
  count(survey, word) %>%
  cast_dtm(survey, word, n)

dim(surveys_dtm)
surveys_dtm

dept_dtm <- all_surveys %>%
  unnest_tokens(word, text) %>%
  count(dept, word) %>%
  cast_dtm(dept, word, n)
```

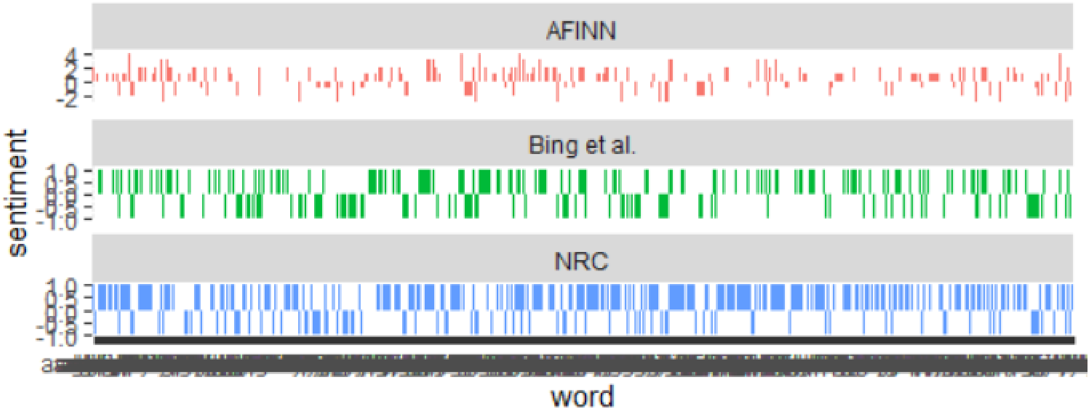


```
> dim(dept_dtm)
[1]    4 2139
> dept_dtm
<<DocumentTermMatrix (documents: 4, terms: 2139)>>
Non-/sparse entries: 3451/5105
Sparsity           : 60%
Maximal term length: 16
Weighting          : term frequency (tf)
>
```

Sentiment Analysis

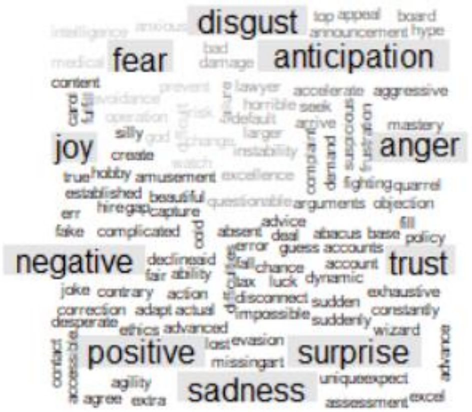
```
####  S E N T I M E N T S  ####
afinn <- all_tokens %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(word) %>% #using integer division to define larger sections of text
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")
```

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

bind_rows(afinn, bing_and_nrc) %>%
  ggplot(aes(word, sentiment, fill=method))+
  geom_col(show.legend=FALSE)+
  facet_wrap(~method, ncol =1, scales= "free_y")
```



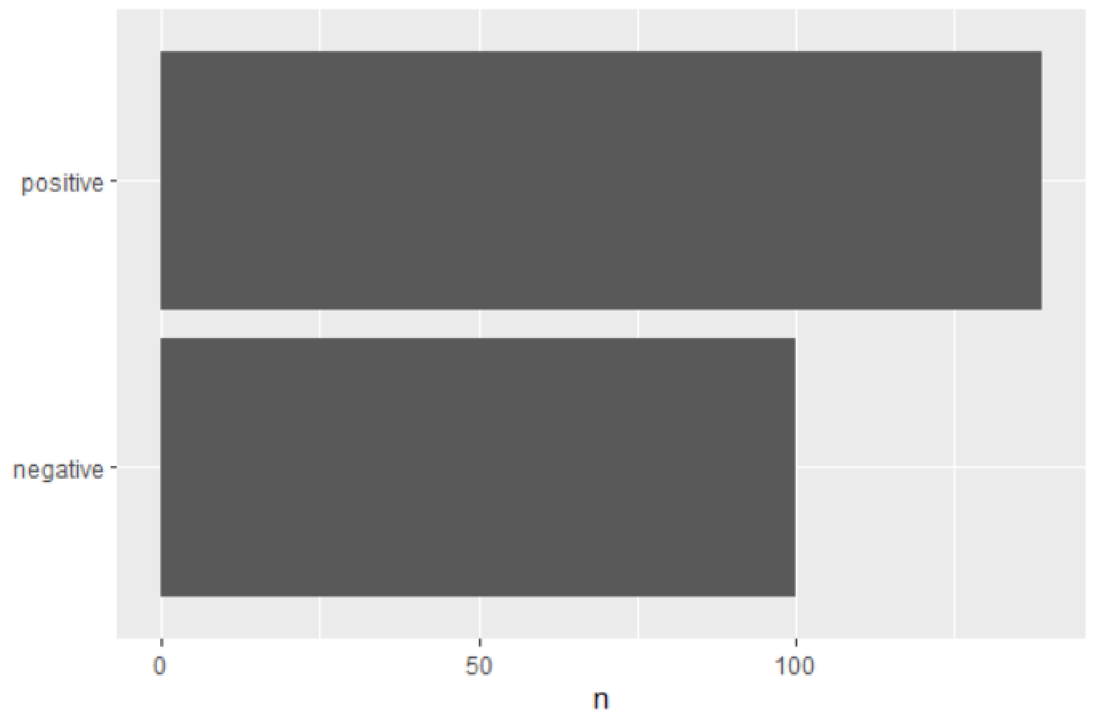
NRC Analysis

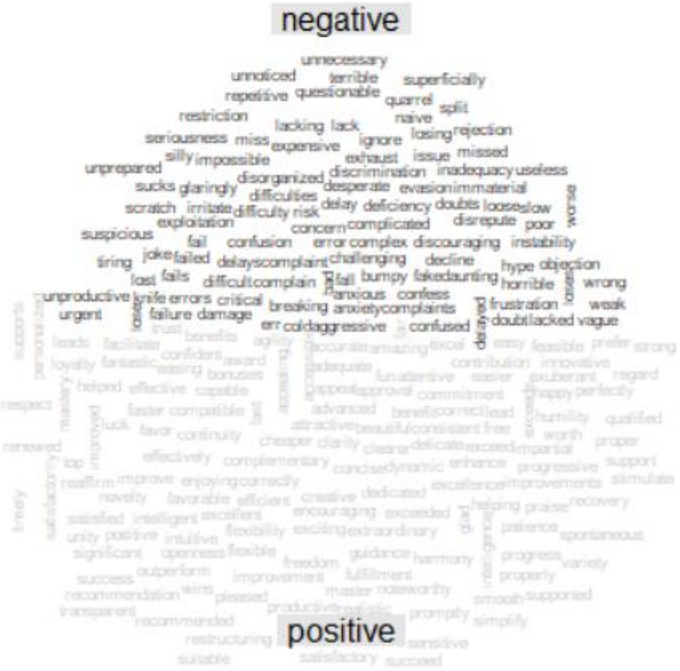
```
####  N R C  ####
nrc_data <- subset(sentiments, lexicon == "nrc")

tokens_sentiment <- all_tokens %>%
  inner_join(nrc_data)

tokens_sentiment_count <- tokens_sentiment %>%
  count(sentiment, sort=T)

nrc_hist <- tokens_sentiment_count %>%
  mutate(sentiment=reorder(sentiment,n)) %>%
  ggplot(aes(sentiment, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(nrc_hist)

all_tokens %>%
  inner_join(get_sentiments("nrc")) %>%
  count(word, sentiment, sort=TRUE) %>%
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("grey20","grey40", "grey60", "gray80"),
                   max.words=500,
                   scale=c(0.5,0.5),
                   fixed.asp=TRUE,
                   title.size=1
  )
```



Bing Analysis

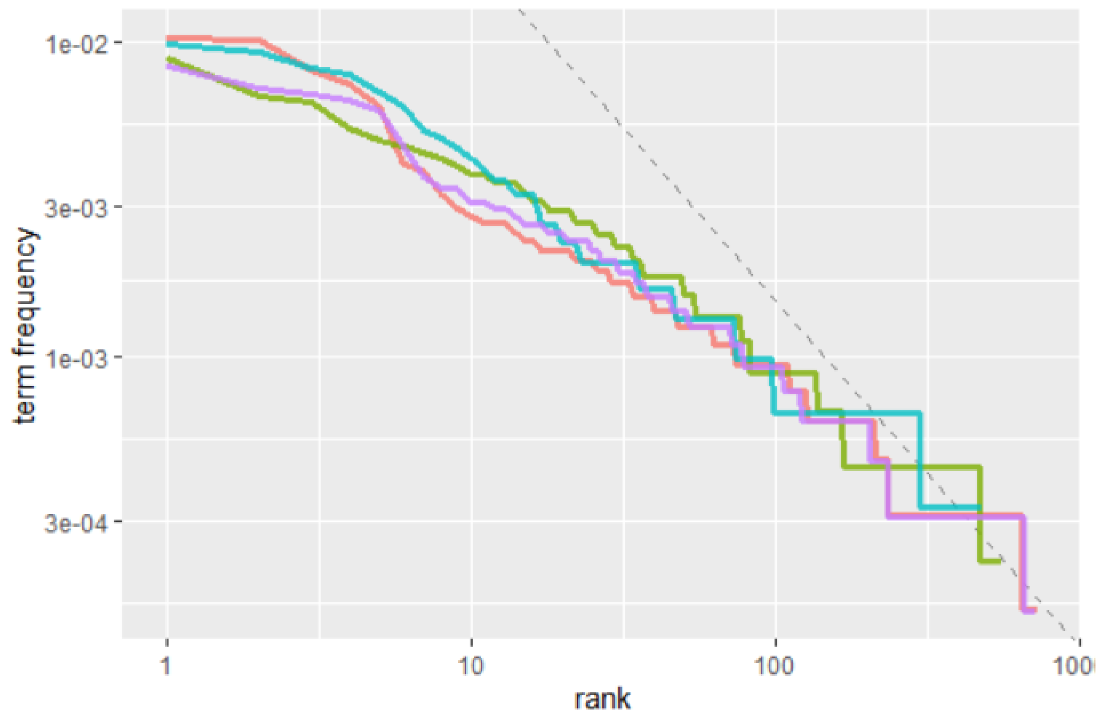
```
####   B I N G   ####
bing_data <- subset(sentiments, lexicon == "bing")

tokens_sentiment <- all_tokens %>%
  inner_join(bing_data)

tokens_sentiment_count <- tokens_sentiment %>%
  count(sentiment, sort=T)

bing_hist <- tokens_sentiment_count %>%
  mutate(sentiment=reorder(sentiment,n)) %>%
  ggplot(aes(sentiment, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(bing_hist)

bing_counts <- all_tokens %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T) %>%
  ungroup()

bing_counts %>%
  group_by(sentiment) %>%
  top_n(1) %>%
  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()

all_tokens %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=TRUE) %>%
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("grey20", "gray80"),
                   max.words=500,
                   scale=c(0.5,0.5),
                   fixed.asp=TRUE,
                   title.size=1
   )
```

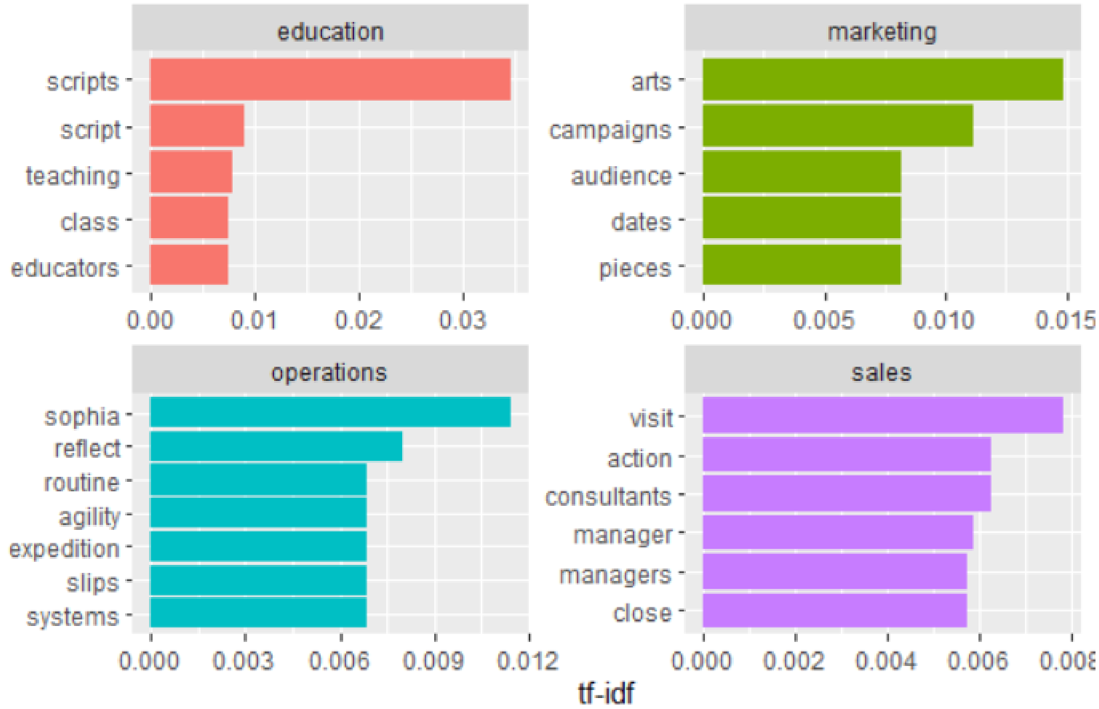

positive
negative
0
50
100
n


negative
unnecessary
unnoticed terrible superficially
repetitive questionable quarrel split
restriction naive
lacking lack losing rejection
seriousness miss expensive ignore
silly impossible exhaust issue missed
unprepared discrimination inadequacy useless
disorganized desperate evasion immaterial
sucks glaringly difficulties worse
scratch irritate difficulty risk delay deficiency doubts loose slow
exploitation disrepute poor
concern complicated
suspicious fail confusion error complex discouraging instability
tiring joke failed delays complaint challenging decline
hype objection
lost fails difficult complain fall bumpy fake daunting horrible wrong
unproductive knife errors critical anxious confess
urgent failure damage breaking anxiety complaints frustration weak
err cold aggressive confused delayed doubt lacked vague
supports trust benefits agility
leads facilitate accurate amazing excel easy feasible prefer strong
loyalty fantastic confident award contribution innovative
easing bonuses adequate
helped effective capable fun attentive easier exuberant regard
respect appeal approval commitment happy perfectly
faster compatible advanced benefit correct lead humility qualified
luck favor continuity attractive beautiful consistent free worth
renewed cheaper clarity cleaner delicate exceed impartial proper
effectively complementary concise dynamic enhance progressive support
top improved
reaffirm improve enjoying correctly excellence improvements stimulate
novelty favorable efficient creative dedicated
timely satisfactory helping praise recovery
satisfied intelligent excellent encouraging exceeded
unity positive intuitive flexibility exciting extraordinary patience spontaneous
significant openness flexible guidance harmony progress variety
freedom
success outperform improvement fulfillment properly
recommendation wins pleased master noteworthy smooth supported
transparent productive promptly simplify
recommended
positive
restructuring sensitive
suitable satisfactory succeed

ZIPF and RF-IDF Analysis

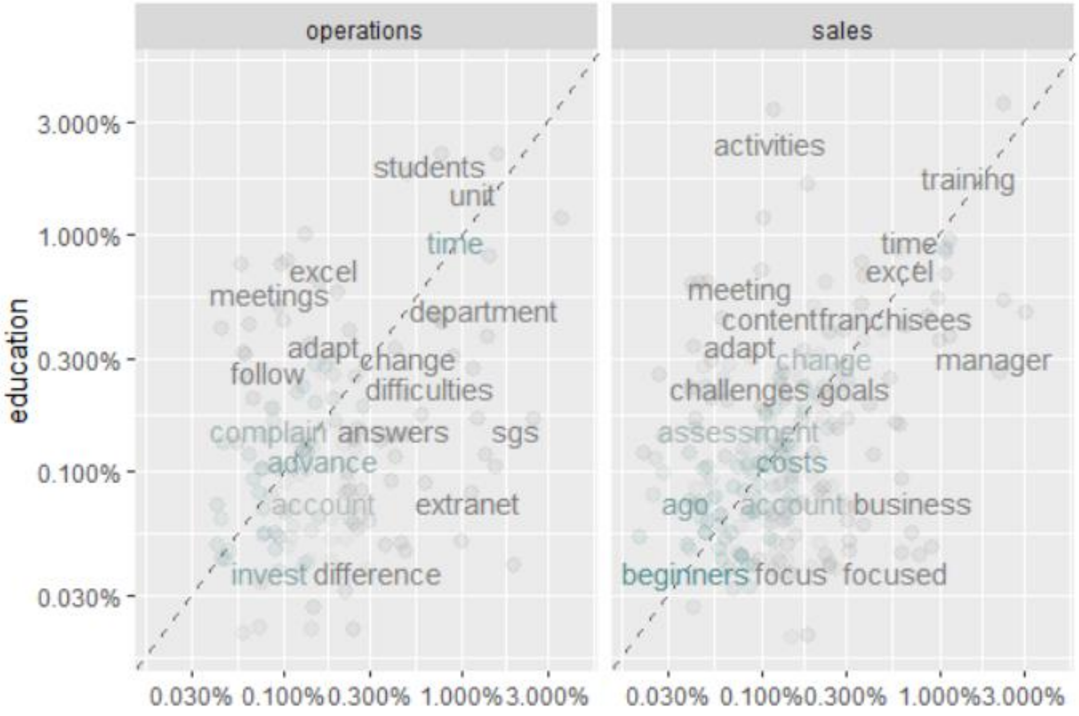
```
####  Z I P F  ####
dept_words <- all_surveys %>%
  unnest_tokens(word, text) %>%
  count(dept, word, sort=TRUE) %>%
  group_by(dept) %>%
  summarize(total=sum(n))

dept_words <- left_join(dept_tokens, dept_words)

ggplot(dept_words, aes(n/total, fill = dept))+
  geom_histogram(show.legend=FALSE)+
  xlim(NA, 0.001) +
  facet_wrap(~dept, ncol=2, scales="free_y")

dept_rank <- dept_words %>%
  group_by(dept) %>%
  mutate(rank = row_number(),
         `term frequency` = n/total)

dept_rank %>%
  ggplot(aes(rank, `term frequency`, color=dept))+
  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()

dept_words <- dept_words %>%
  bind_tf_idf(word, dept, n)

dept_words %>%
  arrange(desc(tf_idf))

dept_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(dept) %>%
  top_n(5) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=dept))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~dept, ncol=2, scales="free")+
  coord_flip()
```

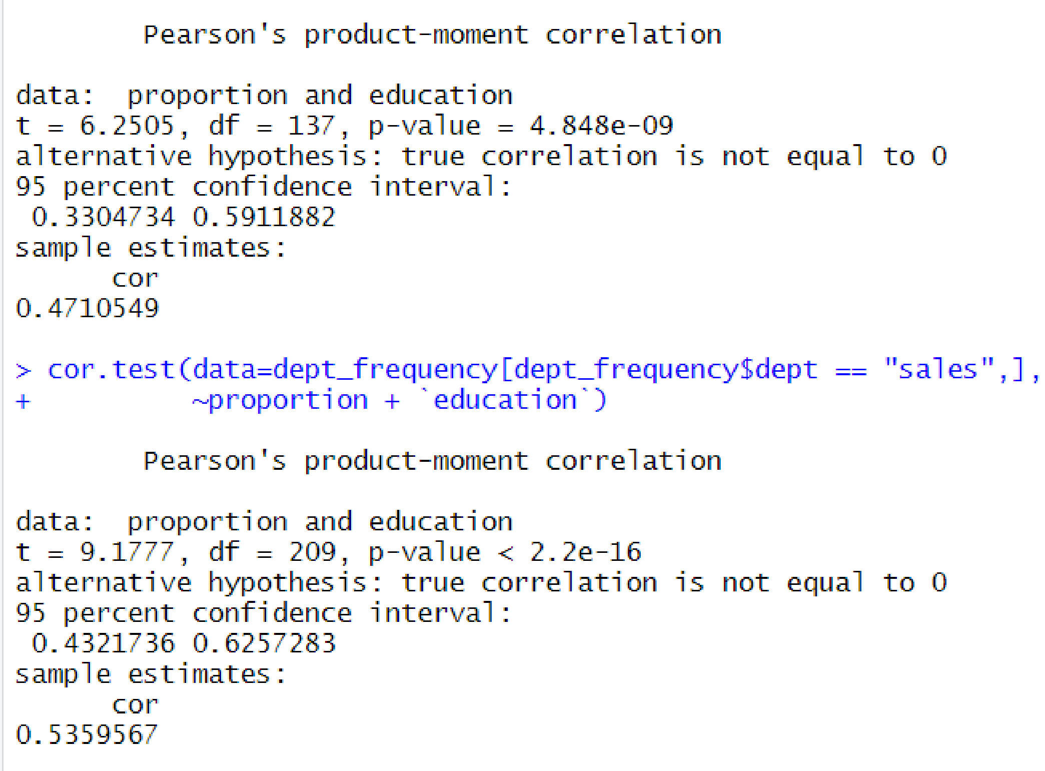

1e-02
3e-03
1e-03
3e-04
term frequency
1
10
100
1000
rank


education
scripts
script
teaching
class
educators
0.00
0.01
0.02
0.03
marketing
arts
campaigns
audience
dates
pieces
0.000
0.005
0.010
0.015
operations
sophia
reflect
routine
agility
expedition
slips
systems
0.000
0.003
0.006
0.009
0.012
sales
visit
action
consultants
manager
managers
close
0.000
0.002
0.004
0.006
0.008
tf-idf

Frequency Analysis

```
#### F R E Q U E N C Y  A N A L Y S I S ####
dept_frequency <- all_surveys %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  mutate(word=str_extract(word, "[a-z']+")) %>%
  count(dept, word) %>%
  group_by(dept) %>%
  mutate(proportion = n/sum(n)) %>%
  select(-n) %>%
  spread(dept, proportion) %>%
  gather(dept, proportion, `sales`, `marketing`, `operations`)

dept_frequency <- dept_frequency %>% filter(dept != 'marketing')
ggplot(dept_frequency, aes(x=proportion, y=`education`,
                      color = abs(`education`- 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=0.5) +
  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(~dept, ncol=2)+
  theme(legend.position = "none")+
  labs(y= "education", x=NULL)

cor.test(data=dept_frequency[dept_frequency$dept == "education",],
         ~proportion + `sales`)

cor.test(data=dept_frequency[dept_frequency$dept == "marketing",],
         ~proportion + `sales`)

cor.test(data=dept_frequency[dept_frequency$dept == "operations",],
         ~proportion + `sales`)
```





```
        Pearson's product-moment correlation

data:  proportion and education
t = 6.2505, df = 137, p-value = 4.848e-09
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3304734 0.5911882
sample estimates:
      cor 
0.4710549 

> cor.test(data=dept_frequency[dept_frequency$dept == "sales",],
+          ~proportion + `education`)

        Pearson's product-moment correlation

data:  proportion and education
t = 9.1777, df = 209, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4321736 0.6257283
sample estimates:
      cor 
0.5359567 
```