HW 4

CJK

2023-07-09

This is the markdown file for homework assignment 4 for Advanced Analytics. The focus of this homework is creating comment categories via regular expression Final deliverables will be excel files.

Here is the prompt for the assignment: Using the Honeywell Glassdoor Comments from Class, create at least 20 comment categories. These can include subcategories (e.g. Benefits-Healthcare, Benefits-PTO, etc.). Run these over both the Pros and the Cons. You can create 2 different Excel sheets or 1 Excel sheet with one tab for Pros and one tab for Cons.

I will start by ensuring I have the appropriate packages downloaded and in my library for regular expressions.

tryCatch(require(pacman),finally=utils:::install.packages(pkgs='pacman',repos='http://cran.r-project.org'));

## Loading required package: pacman

##   
## The downloaded binary packages are in  
## /var/folders/\_6/7mn81ts936g1klwprl9hyq7m0000gn/T//Rtmp9WNQrg/downloaded\_packages

require(pacman)  
  
pacman::p\_load(openxlsx,   
 tidyverse,   
 devtools,  
 conflicted,  
 lubridate,  
 sentimentr)  
  
#Loading libraries  
library(openxlsx)  
library(tidyverse)  
library(devtools)  
library(conflicted)  
library(lubridate)  
library(sentimentr)

I will also use the conflict\_prefer() function to tell R that when I use the filter() function, I want the function from the dplyr package.

conflict\_prefer("filter", "dplyr")

## [conflicted] Will prefer dplyr::filter over any other package.

I will now read in the CSV file and give it the initial name of ‘Data’. I need to make sure it’s reading in from the project that matches my github repository.

library(readr)  
Data <- read\_csv("~/Documents/DirectoryR/UGA IOMP 2023 AA/some-pig-4/glassdoortest1.csv")

## New names:  
## Rows: 2000 Columns: 5  
## ── Column specification  
## ──────────────────────────────────────────────────────── Delimiter: "," chr  
## (4): date, title, pros, cons dbl (1): ...1  
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ  
## Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
## • `` -> `...1`

My data is 2000 observations of 5 variables. Now I will take a look at my column names to get to know my variables better, and I will give an initial view of my data as well.

colnames(Data)

## [1] "...1" "date" "title" "pros" "cons"

View(Data)

The variable titled ‘…1’ appears to be an ID number, so I will change that variable name to ID.

Data <- Data %>%  
 rename(ID = '...1')

I’m now going to create two new dataframes, one for the ‘pros’ comments and one for the ‘cons’ comments. I will keep the entire original dataframe intact as it is.

Comments\_df\_pros <- Data %>%  
 select(c(ID, pros)) %>%  
 filter(!is.na(pros)) %>%  
 rename('comments' = 'pros')  
  
Comments\_df\_cons <- Data %>%  
 select(c(ID, cons)) %>%  
 filter(!is.na(cons)) %>%  
 rename('comments' = 'cons')

I know from experience in class that a few diamond with a question mark symbols are in several of the comments, and this can cause problems with R correctly reading the text and with R being able to kick anything out to excel later on. I will use the stringi package to make sure this symbol is removed and won’t affect my progress later on.

library(stringi)  
  
Comments\_df\_pros <- Comments\_df\_pros %>%  
 as\_tibble() %>%  
 mutate(comments = str\_replace\_all(comments, "\uFFFD",""))  
  
Comments\_df\_cons <- Comments\_df\_cons %>%  
 as\_tibble() %>%  
 mutate(comments = str\_replace\_all(comments, "\uFFFD",""))

And now that I’ve made the adjustment for that symbol, I can also change comments to be all lowercase. This can help later on so I don’t have to worry about any expressions I create being case-sensitive (for instance, I want to build just one expression that can detect benefits, Benefits, and BENEFITS all equally.)

Comments\_df\_pros <- Comments\_df\_pros %>%  
 mutate(comments = tolower(comments))  
  
Comments\_df\_cons <- Comments\_df\_cons %>%  
 mutate(comments = tolower(comments))

And I can remove all line breaks as well. This will help ensure accurate pattern matching when building out regular expressions.

Comments\_df\_pros$comments <- gsub("[\r\n]", "", Comments\_df\_pros$comments)  
  
Comments\_df\_pros <- Comments\_df\_pros %>%  
 select(ID, comments) %>%  
 na.omit()  
  
Comments\_df\_cons$comments <- gsub("[\r\n]", "", Comments\_df\_cons$comments)  
  
Comments\_df\_cons <- Comments\_df\_cons %>%  
 select(ID, comments) %>%  
 na.omit()

And lastly, I know from viewing the data that a few comments are ‘#name?’ - these should be treated as blanks, so I’m going to tell R to treat any comments of #name? as NAs.

Comments\_df\_pros[Comments\_df\_pros == '#name?'] <- NA  
Comments\_df\_cons[Comments\_df\_cons == '#name?'] <- NA

And now I’m going to remove these NA responses from my dataframes. I can just do a listwise deletion since the comments are separated out (so if someone answered the pros but not the cons question, it will retain their comment in one dataframe and remove the NA from the other).

summary(is.na(Comments\_df\_pros$comments))

## Mode FALSE TRUE   
## logical 1968 32

Comments\_df\_pros <- na.omit(Comments\_df\_pros)  
  
summary(is.na(Comments\_df\_cons$comments))

## Mode FALSE TRUE   
## logical 1974 26

Comments\_df\_cons <- na.omit(Comments\_df\_cons)

And now I will just double check my work. Both should return with FALSE for all remaining numbers of observations (1968 for pros, 1974 for cons).

summary(is.na(Comments\_df\_pros$comments))

## Mode FALSE   
## logical 1968

summary(is.na(Comments\_df\_cons$comments))

## Mode FALSE   
## logical 1974

That worked, and now my dataframes are all ready for creating regular expressions. Before I do this though, I am going to brainstorm some good categories and subcategories of what to search for. What are some themes and patterns in the data that would capture valuable insights for both the pros and the cons comments?

Potential categories could include: - Company culture (subcategories could pertain to leadership/management, org values, company policies, etc) - Employee health (subcategories could relate to stress and burnout, work-life balance, health insurance, etc) - Job enrichment (subcategories could deal with autonomy and flexibility at work, recognition at work, and anything to do with career advancement opportunities)

I will further expand these out and brainstorm them, and other ideas, offline and add my full list below when it is complete!

After my brainstorm, here is my final list of four categories with their subcategories. I ultimately decided to make job enrichment and career growth two different categories since career growth had a lot going on once I dug deeper.

Job Enrichment: \* Remote / Hybrid Options \* Flexibility of Hours / Schedule \* Autonomy at Work \* Recognition at Work \* Employee Perks & Incentives

Career Growth: \* Training & Development \* Career Advancement Opportunities \* Succession & Career Planning \* Miscellaneous Growth Opportunities (Conferences, Certifications, etc)

Employee Health: \* Work-Life Balance \* Stress & Burnout \* Work Safety / Environment \* General Well-Being \* Job Security

Company Culture: \* Leadership / Management - General \* Direct Manager Relationships \* Organizational Values \* Dynamics with Co-workers \* Employee Voice \* Workplace Policies

I will tackle Job Enrichment first. I will comment out within each code chunk to describe what I am hoping to capture with each REGEX pattern I create.

I will also get my reps in with using start\_time from the lubridate package to provide start and end times for the different searches.

start\_time <- now()

Remote / Hyprid Options

enrich\_remote <- c('(?=.\*remote|home|virtual)(?=.\*(?:work))', #this will get any comments that have the word remote/remotely, virtual/virtually, and home with any word that has 'work' in it (work, works, worked, worker(s), or working)  
 '\\btelecommut\\w\*\\b', #this will capture telecommute, telecommutes, telecommuted, or telecommuting  
 '\\b(?:tele\\W+(?:\\w+\\W+){0,1}?commuting)\\b', #this will capture both telecommuting and tele commuting in case anyone made them two separate words instead of one  
 '\\bhybrid\\b', #this one is simple, it will capture the word hybrid (I'm thinking the only reason the word hybrid would show up is how it relates to remote work options)  
 '(?=.\*remote|home|virtual)(?=.\*(?:option|opportunit))' #this one is similar to the first line - it will capture any variant or remote, home or virtual when in the same comment as option(s) or opportunity/opportunities  
  
)  
  
enrich\_remote\_pattern <- paste(enrich\_remote, collapse = "|") #This will combine everything I input for the above subcategory of 'remote work' for job enrichment into one pattern  
  
enrich\_remote\_cons <- as.data.frame(Comments\_df\_cons[grep(enrich\_remote\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),]) # This takes the pattern I built out and searches over the comments in the 'cons' column from the dataframe I built   
  
TESTCONS <- Comments\_df\_cons %>%  
 mutate(enrich\_remote = ifelse(comments %in% enrich\_remote\_cons$comments, "Y",  
 "N")) #This creates a new object, TEST, from my cons dataframe and if any of the comments in the "comments" column match (%in%) the comments exactly, they get a "Y". If not they get a "N" in the new "enrich\_remote" column  
  
#I will now repeat the last two steps for my pros dataframe as well  
  
enrich\_remote\_pros <- as.data.frame(Comments\_df\_pros[grep(enrich\_remote\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- Comments\_df\_pros %>%  
 mutate(enrich\_remote = ifelse(comments %in% enrich\_remote\_pros$comments, "Y",  
 "N"))

All of that worked! So I now have my first subcategory of Remote/Hybrid Work with 34 pros comments and 70 cons comments. I did view the two dataframes to ensure they are picking up the types of comments I wanted, and they seem to be doing what I would expect them to do. I will now repeat these steps for my other Job Enrichment subcategories.

Flexibility of Hours / Schedule

enrich\_flexible <- c('(?=.\*flex?)(?=.\*(?:hour|schedul|time|arrang))',  
 '(?=.\*choos?)(?=.\*(?:hour|schedul|time|arrang))', #This one should pick up on flex/flexible/flexibility or choose/chooses/choosing combined with either hour/hours or schedule/schedules/scheduled/scheduling or time/times or arrangement/arranging/arrange  
 '\\b(?:set\\W+(?:\\w+\\W+){0,3}?schedule?|hour?)\\b', #This one will capture any comments that say the word 'set', and within three words say schedule(s) or hour(s). So if someone says 'I set my own schedule' or 'I set my hours' it should pick up these, and similar comments.  
 '\\bflexible\\s+work\\s?week\\b' #This will capture either the phrase flexible workweek or flexible work week.   
  
)  
  
enrich\_flexible\_pattern <- paste(enrich\_flexible, collapse = "|")  
  
enrich\_flexible\_cons <- as.data.frame(Comments\_df\_cons[grep(enrich\_flexible\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(enrich\_flexible = ifelse(comments %in% enrich\_flexible\_cons$comments, "Y",  
 "N"))  
  
enrich\_flexible\_pros <- as.data.frame(Comments\_df\_pros[grep(enrich\_flexible\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(enrich\_flexible = ifelse(comments %in% enrich\_flexible\_pros$comments, "Y",  
 "N"))

Once more, I viewed the dataframes, and the above code seems to have correctly picked up on the patterns of comments dealing with flexible schedules.

Autonomy at Work

enrich\_autonomy <- c('^.\*autonom.\*$','^.\*independen.\*$','^.\*empower.\*$',#These three will select comments that mention autonomy/autonomous, independent/independence, freedom, or empower/empowering/empowerment  
 '\\b(?:control\\W+(?:\\w+\\W+){0,4}work)\\b', #This one will pick up on control and work within four words of each other to select phrases like 'I have control over my work' or 'I don't have control over what I work on'  
 '\\b(?:micro\\W+(?:\\w+\\W+){0,1}?management|managing|manage|manages|managed)\\b' #I tried to find a good way of doing the 'manag?' to capture all variants of micromanag\_\_\_\_\_ but I couldn't get it to work as seamlessly as I'd hoped! I'm sure there's a prettier way, but this should take care of all variants and forms of micro-manage, micro manage, or micromanage. (I'd like to know how to combine the concepts of the individual word variant code for autonomy and implement it here, but I couldn't get that to work!)  
   
)  
  
  
enrich\_autonomy\_pattern <- paste(enrich\_autonomy, collapse = "|")  
  
enrich\_autonomy\_cons <- as.data.frame(Comments\_df\_cons[grep(enrich\_autonomy\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(enrich\_autonomy = ifelse(comments %in% enrich\_autonomy\_cons$comments, "Y",  
 "N"))  
  
enrich\_autonomy\_pros <- as.data.frame(Comments\_df\_pros[grep(enrich\_autonomy\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(enrich\_autonomy = ifelse(comments %in% enrich\_autonomy\_pros$comments, "Y",  
 "N"))

Once I adjusted the micromanagement code, it seems that the patterns were able to successfully capture Autonomy at Work.

Recognition at Work

enrich\_recognition <- c('^.\*recogniz.\*$', #This one should capture recognize(s/d) or recognizing (at first, I played around with recognition, but several comments mentioned 'brand recognition' or 'name recognition of the company. I played around for a long time trying to figure out a way of doing 'recognition where it doesn't follow the word brand or name' but I was unsuccessful. Below are some of my attempts!)  
# '\\b(?!brand\\s+)recognition\\b'  
# '\\brecognition\\b(?!(?:\\s+(?:brand|name)\\s+recognition))'  
# '\\brecognition\\b(?!\\s+(?:brand\\b|name\\b)\\s+recognition)'  
# '\\brecognition\\b(?!(?:\\s+(?:brand|name)\\b)\\s+recognition)'  
 '^.\*acknowledg.\*$', #This should capture the words acknowledge, acknowledgement, acknowledging, etc.  
 '(?=.\*praise|notice|focus|attention|recognition)(?=.\*(?:performance|achievement|accomplishment|performing|effort))' #This should capture 'recognition' type words (praise, notice, etc) when the comment also mentions a 'performance' type word (achievement, accomplishment, etc).  
   
)  
  
  
  
enrich\_recognition\_pattern <- paste(enrich\_recognition, collapse = "|")  
  
enrich\_recognition\_cons <- as.data.frame(Comments\_df\_cons[grep(enrich\_recognition\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(enrich\_recognition = ifelse(comments %in% enrich\_recognition\_cons$comments, "Y",  
 "N"))  
  
enrich\_recognition\_pros <- as.data.frame(Comments\_df\_pros[grep(enrich\_recognition\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(enrich\_recognition = ifelse(comments %in% enrich\_recognition\_pros$comments, "Y",  
 "N"))

Employee Perks & Incentives

enrich\_perks <- c('^.\*perk.\*$','^.\*incentiv.\*$', #These two should pick up on perk(s) and incentive(s) as well as incentivize.  
 '^.\*bonus.\*$','^.\*reward.\*$','^.\*gift.\*$' #These three should pick up on bonus(es), reward(s) and gift(s) - I structured reward and gift code to avoid other variants of the words like rewarding, gifted, etc.  
#Overall, I kept these codes more simple because, in looking at the data, these word types only seemed to appear in reference to employee perks and weren't used in other instances.   
   
)  
  
enrich\_perks\_pattern <- paste(enrich\_perks, collapse = "|")  
  
enrich\_perks\_cons <- as.data.frame(Comments\_df\_cons[grep(enrich\_perks\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(enrich\_perks = ifelse(comments %in% enrich\_perks\_cons$comments, "Y",  
 "N"))  
  
enrich\_perks\_pros <- as.data.frame(Comments\_df\_pros[grep(enrich\_perks\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(enrich\_perks = ifelse(comments %in% enrich\_perks\_pros$comments, "Y",  
 "N"))

The Job Enrichment category is complete. I will now move to the Career Growth category.

Training & Development

career\_train <- c('^.\*learn.\*$','^.\*train.\*$', #These two codes should capture learn(s), learned, learning, and train(s) trained, training, trainer(s).  
 #This should capture the word skill(s) being used within three words of either the word gain(s), develop(s) (or variants like developing or development), or the word build(s) (or building).  
 '\\b(?:skill[s]?\\W+(?:\\w+\\W+){0,3}?(?:develop\\w\*|build\\w\*|gain\\w\*)|(?:develop\\w\*|build\\w\*|gain\\w\*)\\W+(?:\\w+\\W+){0,3}?skill[s]?)\\b',  
 '\\b(?:talent?\\W+(?:\\w+\\W+){0,3}?(?:develop\\w\*|build\\w\*|gain\\w\*)|(?:develop\\w\*|build\\w\*|gain\\w\*)\\W+(?:\\w+\\W+){0,3}?talent?)\\b',  
 '\\b(?:knowledge?\\W+(?:\\w+\\W+){0,3}?(?:develop\\w\*|build\\w\*|gain\\w\*)|(?:develop\\w\*|build\\w\*|gain\\w\*)\\W+(?:\\w+\\W+){0,3}?knowledge?)\\b' #The three long codes will, in order, find comments with the word (1) skill(s), or (2) talent (or variants like talented), or (3) knowledge(able) within three words of either develop(s) (or developed/development), build(s), or gain(s)  
   
)  
  
career\_train\_pattern <- paste(career\_train, collapse = "|")  
  
career\_train\_cons <- as.data.frame(Comments\_df\_cons[grep(career\_train\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(career\_train = ifelse(comments %in% career\_train\_cons$comments, "Y",  
 "N"))  
  
career\_train\_pros <- as.data.frame(Comments\_df\_pros[grep(career\_train\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(career\_train = ifelse(comments %in% career\_train\_pros$comments, "Y",  
 "N"))

Career Advancement Opportunities

career\_advance <- c('\\b(?:career\\s+advancement|advancement\\s+opportunities)\\b', #This should capture either the phrase career advancement or advancement opportunities  
 '^.\*promot.\*$', #This should capture any variant of the word promote (promoted, promotes, promotion, promotions, etc)  
 '\\bupward\\s+mobility\\b', #This should capture the specific phrase 'upward mobility'  
 '\\bmove\\s+up\\b' #Lastly, this will detect the phrase 'move up' (after examining the comments, this phrase alone, without any other qualifiers, will capture more related comments and not capture false positives - this phrase only seems to be used in relation to career advancement)  
   
)  
  
career\_advance\_pattern <- paste(career\_advance, collapse = "|")  
  
career\_advance\_cons <- as.data.frame(Comments\_df\_cons[grep(career\_advance\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(career\_advance = ifelse(comments %in% career\_advance\_cons$comments, "Y",  
 "N"))  
  
career\_advance\_pros <- as.data.frame(Comments\_df\_pros[grep(career\_advance\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(career\_advance = ifelse(comments %in% career\_advance\_pros$comments, "Y",  
 "N"))

Succession & Career Planning

career\_plan <- c('^.\*succession.\*$', #This one detects the word succession  
 '\\bcareer\\s+plan\\w\*\\b', #This will detect career followed by plan, plans, or planning  
 '\\bcareer\\s+develop\\w\*\\b', #Similar to the above, this picks up on career followed by developing, development, develop.  
 '\\badvancement\\s+plan\\w\*\\b' #This one detects the phrase advancement plan(s)/planning  
   
)  
  
career\_plan\_pattern <- paste(career\_plan, collapse = "|")  
  
career\_plan\_cons <- as.data.frame(Comments\_df\_cons[grep(career\_plan\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(career\_plan = ifelse(comments %in% career\_plan\_cons$comments, "Y",  
 "N"))  
  
career\_plan\_pros <- as.data.frame(Comments\_df\_pros[grep(career\_plan\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(career\_plan = ifelse(comments %in% career\_plan\_pros$comments, "Y",  
 "N"))

Miscellaneous Growth Opportunities (Extras)

career\_extra <- c('^.\*certification.\*$','^.\*conference.\*$', #First code will detect the word certification(s), second code will detect the word conference  
 '\\b(?:tuition|education|college)\\W+(?:\\w+\\W+){0,3}?(?:reimbursement|stipend|reimbursements|stipends)\\b|\\b(?:reimbursement|stipend|reimbursements|stipends)\\W+(?:\\w+\\W+){0,3}?(?:tuition|education|college)\\b', #This will detect either tuition, education, or college within 3 words (before or after) the word reimbursement(s) or stipends(s)  
 '\\bcontinuing\\s+education\\b' #This will detect the phrase 'continuing education'  
   
)  
  
career\_extra\_pattern <- paste(career\_extra, collapse = "|")  
  
career\_extra\_cons <- as.data.frame(Comments\_df\_cons[grep(career\_extra\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(career\_extra = ifelse(comments %in% career\_extra\_cons$comments, "Y",  
 "N"))  
  
career\_extra\_pros <- as.data.frame(Comments\_df\_pros[grep(career\_extra\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(career\_extra = ifelse(comments %in% career\_extra\_pros$comments, "Y",  
 "N"))

The second category, Career Advancement, is complete. I will now move to Employee Health.

Work-Life Balance

health\_wlbal <- c('\\bwork[-\\s]?life[-\\s]?balance\\b', #This will pick up on the phrase work life balance and allow for optional hyphens  
 '\\btime\\s+off\\b', #This will find the phrase time off  
 '\\bsocial\\s+life\\b', #This will find the phrase social life  
 '\\b(?:work\\W+(?:\\w+\\W+){0,3}life)\\b',  
 '\\b(?:life\\W+(?:\\w+\\W+){0,3}work)\\b', #These will find the word work within 3 words of the word life (taking either order into account)  
 '\\b(?:balance\\W+(?:\\w+\\W+){0,3}work)\\b',  
 '\\b(?:work\\W+(?:\\w+\\W+){0,3}balance)\\b' #And similarly, these will find the word balance within 3 words of the word work (taking either order into account)  
  
)  
  
  
health\_wlbal\_pattern <- paste(health\_wlbal, collapse = "|")  
  
health\_wlbal\_cons <- as.data.frame(Comments\_df\_cons[grep(health\_wlbal\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(health\_wlbal = ifelse(comments %in% health\_wlbal\_cons$comments, "Y",  
 "N"))  
  
health\_wlbal\_pros <- as.data.frame(Comments\_df\_pros[grep(health\_wlbal\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(health\_wlbal = ifelse(comments %in% health\_wlbal\_pros$comments, "Y",  
 "N"))

Stress & Burnout

health\_stress <- c('^.\*stress.\*$', #This will detect stress, stressed, stressful, stressor(s), etc  
 '\\bburnout\\b', '\\bburn[-\\s]?out\\b', #The first code detects the word burnout, the second detects both burn-out and burn out  
 '\\bexhaust\\w\*\\b', '\\boverwork\\w\*\\b', #The first code detects exhausted, exhausts, exhausting, etc. The second detects overworks, overworked, overworking, etc.  
 '\\bwork\\s?load\\b', #This detects the word workload or work load  
 '^.\*overtime.\*$' #This detects the word overtime  
   
)  
  
health\_stress\_pattern <- paste(health\_stress, collapse = "|")  
  
health\_stress\_cons <- as.data.frame(Comments\_df\_cons[grep(health\_stress\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(health\_stress = ifelse(comments %in% health\_stress\_cons$comments, "Y",  
 "N"))  
  
health\_stress\_pros <- as.data.frame(Comments\_df\_pros[grep(health\_stress\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(health\_stress = ifelse(comments %in% health\_stress\_pros$comments, "Y",  
 "N"))

Work Safety

health\_safety <- c('\\bwork\\w\*\\s+safety\\b', #This will detect the phrase work safety, or workplace safety as well  
 '\\bsafety\\s+training[s]?\\b', #This will detect the phrase safety training  
 '\\brisk\\s+assessment[s]?\\b', #This will detect the phrase risk assessment(s)  
 '\\bsafety\\s+risk[s]?\\b', #This will detect the phrase safety risk(s)  
 '\\binjur\\w\*\\b', #This will detect the word injury or injuries  
 '^.\*safe.\*$' #This will detect safe, unsafe, safety, etc. I employed this code when the above patterns returned with very little. I did vet the output and didn't see anything out of place when employing this code (so no one talking about 'playing it safe' with a business decision or something else not related to actual work safety).  
  
)  
  
  
health\_safety\_pattern <- paste(health\_safety, collapse = "|")  
  
health\_safety\_cons <- as.data.frame(Comments\_df\_cons[grep(health\_safety\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(health\_safety = ifelse(comments %in% health\_safety\_cons$comments, "Y",  
 "N"))  
  
health\_safety\_pros <- as.data.frame(Comments\_df\_pros[grep(health\_safety\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(health\_safety = ifelse(comments %in% health\_safety\_pros$comments, "Y",  
 "N"))

General Wellbeing

health\_wellbeing <- c('\\bwellbeing\\b', '\\bwell[-\\s]?being\\b', #These two codes will detect either wellbeing, well-being, or well being  
 '\\bmental[-\\s]?health\\b', #This will detect mental-health or mental health  
 '\\bphysical[-\\s]?health\\b', #This will detect physical-health or physical health  
 '\\bwellness[-\\s]?programs?\\b', #This wil detect the phrase wellness program(s)  
 '\\bhealthcare\\b', '\\bhealth[-\\s]?care\\b', #This will find healthcare, health care, or health-care  
 '\\bhealth\\s+insurance\\b',  
 '\\bmedical\\s+insurance\\b', #These two codes will detect the phrases health insurance and medical insurance  
 '^.\*health.\*$' #I opted to just use the detection for any use of health. After I used this, I evaluated the results to check for false positives and didn't see anything out of place (ie nothing about the health of the business, etc)  
   
)  
  
health\_wellbeing\_pattern <- paste(health\_wellbeing, collapse = "|")  
  
health\_wellbeing\_cons <- as.data.frame(Comments\_df\_cons[grep(health\_wellbeing\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(health\_wellbeing = ifelse(comments %in% health\_wellbeing\_cons$comments, "Y",  
 "N"))  
  
health\_wellbeing\_pros <- as.data.frame(Comments\_df\_pros[grep(health\_wellbeing\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(health\_wellbeing = ifelse(comments %in% health\_wellbeing\_pros$comments, "Y",  
 "N"))

Job Security

health\_security <- c('\\bjob\\s+(security|stability)\\b', #This will detect the phrase job security or job stability  
 '^.\*layoff.\*$', #This will detect the word layoff(s)  
 '^.\*furlough.\*$', #This will detect the word furlough(s)  
 '\\blay[-\\s]?off[s]?\\b', #This will detect lay off or lay-off (and plural of both)  
 '\\beconomic\\s+downturn\\b', #This will detect the phrase economic downturn  
 '\\bjob\\s+(insecurity|instability)\\b', #This will detect job insecurity or job instability  
 '^.\*restructur.\*$', #This will detect the words restructure(s/d) or restructuring  
 '^.\*outsourc.\*$' #This will detect the words outsource(s/d) or outsourcing  
   
   
)  
  
health\_security\_pattern <- paste(health\_security, collapse = "|")  
  
health\_security\_cons <- as.data.frame(Comments\_df\_cons[grep(health\_security\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(health\_security = ifelse(comments %in% health\_security\_cons$comments, "Y",  
 "N"))  
  
health\_security\_pros <- as.data.frame(Comments\_df\_pros[grep(health\_security\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(health\_security = ifelse(comments %in% health\_security\_pros$comments, "Y",  
 "N"))

I now have my last category, Company Culture, to complete, and I am done!

Leadership / Management - General

culture\_lead <- c('^.\*leader.\*$', #This should capture leader(s), leadership  
 '^.\*management.\*$', #This should capture the word management (I want to get more specific with 'manager' in the next subcategory so I won't include that yet)  
 '^.\*executive\*$' #This should capture executive(s)  
   
   
)  
  
culture\_lead\_pattern <- paste(culture\_lead, collapse = "|")  
  
culture\_lead\_cons <- as.data.frame(Comments\_df\_cons[grep(culture\_lead\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(culture\_lead = ifelse(comments %in% culture\_lead\_cons$comments, "Y",  
 "N"))  
  
culture\_lead\_pros <- as.data.frame(Comments\_df\_pros[grep(culture\_lead\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(culture\_lead = ifelse(comments %in% culture\_lead\_pros$comments, "Y",  
 "N"))

Direct Manager Relationships

culture\_mngr <- c('^.\*manager.\*$', #This should capture manager(s) (I specifically made this mutually exclusive from the previous subcategory)  
 '^.\*supervisor.\*$' #This should capture supervisor(s)  
   
)  
  
culture\_mngr\_pattern <- paste(culture\_mngr, collapse = "|")  
  
culture\_mngr\_cons <- as.data.frame(Comments\_df\_cons[grep(culture\_mngr\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(culture\_mngr = ifelse(comments %in% culture\_mngr\_cons$comments, "Y",  
 "N"))  
  
culture\_mngr\_pros <- as.data.frame(Comments\_df\_pros[grep(culture\_mngr\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(culture\_mngr = ifelse(comments %in% culture\_mngr\_pros$comments, "Y",  
 "N"))

Organizational Values

culture\_value <- c('^.\*value.\*$', #Will detect the word value(s/d)   
 '\\bmission\\b', #Will detect the word mission (set up strict word boundaries to not also capture words like 'commission')  
 '^.\*standard\*$', #Will detect the word standard(s)  
 '^.\*ethics.\*$', #Will detect the word ethics  
 '^.\*culture.\*$', #Will detect the word culture  
 '^.\*integrity.\*$' #Will detect the word integrity  
   
)  
  
culture\_value\_pattern <- paste(culture\_value, collapse = "|")  
  
culture\_value\_cons <- as.data.frame(Comments\_df\_cons[grep(culture\_value\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(culture\_value = ifelse(comments %in% culture\_value\_cons$comments, "Y",  
 "N"))  
  
culture\_value\_pros <- as.data.frame(Comments\_df\_pros[grep(culture\_value\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(culture\_value = ifelse(comments %in% culture\_value\_pros$comments, "Y",  
 "N"))

Dynamics with Co-workers

culture\_cowork <- c('\\bcoworker\\w\*\\b', '\\bco[-\\s]?worker\\w\*\\b', #Will detect coworker(s), co worker(s), or co-worker(s)  
 '\\bteamwork\\w\*\\b', '\\bteam[-\\s]?work\\w\*\\b', #Will detect teamwork, team work, or team-work  
 '\\bteammate\\w\*\\b', '\\bteam[-\\s]?mate\\w\*\\b', #Will detect teammate, team mate, or team-mate  
 '^.\*camaraderie.\*$', #Will detect the word camaraderie  
 '^.\*relationship.\*$' #Will detect the word relationship(s)  
   
)  
  
culture\_cowork\_pattern <- paste(culture\_cowork, collapse = "|")  
  
culture\_cowork\_cons <- as.data.frame(Comments\_df\_cons[grep(culture\_cowork\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(culture\_cowork = ifelse(comments %in% culture\_cowork\_cons$comments, "Y",  
 "N"))  
  
culture\_cowork\_pros <- as.data.frame(Comments\_df\_pros[grep(culture\_cowork\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(culture\_cowork = ifelse(comments %in% culture\_cowork\_pros$comments, "Y",  
 "N"))

Employee Voice

culture\_voice <- c('^.\*voice.\*$', #Will detect the word voice  
 '^.\*input.\*$', #Will detect the word input  
 '^.\*participat.\*$', #Will detect participating, participation, participate(s/d)  
 '^.\*involvement.\*$', #Will detect involvement  
 '^.\*opinion.\*$', #Will detect opinion(s)  
 '^.\*listen.\*$' #Will detect listens, listened, listening  
   
)  
  
culture\_voice\_pattern <- paste(culture\_voice, collapse = "|")  
  
culture\_voice\_cons <- as.data.frame(Comments\_df\_cons[grep(culture\_voice\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(culture\_voice = ifelse(comments %in% culture\_voice\_cons$comments, "Y",  
 "N"))  
  
culture\_voice\_pros <- as.data.frame(Comments\_df\_pros[grep(culture\_voice\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(culture\_voice = ifelse(comments %in% culture\_voice\_pros$comments, "Y",  
 "N"))

Workplace Policies

culture\_policy <- c('^.\*polic\*$', #Will detect policy/policies  
 '^.\*bureaucra\*$', #Will detect bureaucracy/bureaucratic  
 '^.\*politic\*$', #Will detect politics/political  
 '^.\*procedur\*$', #Will detect procedure(s), procedural  
 '^.\*rule.\*$', #Will detect rule(s)  
 '^.\*regulation.\*$', #Will detect regulation(s)  
 '\\bred\\s+tape\\b',  
 '^.\*decision.\*$' #Will detect decision(s)  
   
)  
  
culture\_policy\_pattern <- paste(culture\_policy, collapse = "|")  
  
culture\_policy\_cons <- as.data.frame(Comments\_df\_cons[grep(culture\_policy\_pattern, Comments\_df\_cons$comments, value = FALSE, perl = TRUE),])   
  
TESTCONS <- TESTCONS %>%  
 mutate(culture\_policy = ifelse(comments %in% culture\_policy\_cons$comments, "Y",  
 "N"))  
  
culture\_policy\_pros <- as.data.frame(Comments\_df\_pros[grep(culture\_policy\_pattern, Comments\_df\_pros$comments, value = FALSE, perl = TRUE),])   
  
TESTPROS <- TESTPROS %>%  
 mutate(culture\_policy = ifelse(comments %in% culture\_policy\_pros$comments, "Y",  
 "N"))

All regex patterns have been created for each subcategory. I will now add an ‘other’ category to both my pros and cons lists, and I will end the lubridate start time.

TESTCONS <- TESTCONS %>%  
 mutate(Other = apply(TESTCONS, 1, function(y){ ifelse("Y" %in% y, "N", "Y")}))  
  
TESTPROS <- TESTPROS %>%  
 mutate(Other = apply(TESTPROS, 1, function(y){ ifelse("Y" %in% y, "N", "Y")}))

end\_time <- now()  
  
print(difftime(end\_time, start\_time))

## Time difference of 8.419585 secs

And last but certainly not least, I will create my fancy excel sheets, one for the cons and one for the pros. I have left the comments as is from Rob Stilson (stealing like an artist, but giving credit!) to remind me what each line of code does. Also note that I will copy and paste both the code and the comments to repeat the process for the pros after creating the worksheet for the cons (so the comments will be identical for both, but that’s easier than removing them all from the second chunk).

INTRO <- c("Honeywell Comments",  
  
 "Data Source: Glassdoor",  
  
 "Data As Of: Q3 2020",  
  
 "Prepared on: 7/12/2023",  
  
 "Prepared by: Charlotte Kilbourn")  
  
wb <- openxlsx::createWorkbook() #Create a work book  
  
  
#Comment Report  
  
addWorksheet(wb, "Comment Report Cons") #name the worksheet in Excel  
  
writeData(wb, "Comment Report Cons", INTRO) #Write your INTRO  
  
  
#Create style  
  
style1 <- createStyle(fontColour = "#327471", textDecoration = "Bold") #Choose your custom font color (https://www.rgbtohex.net/) and make it bold. Call it style1  
  
   
  
addStyle(wb, style = style1, rows= 1:5, cols = 1, sheet = "Comment Report Cons") #add this style to your worksheet. Tell it which rows and columns  
  
writeData(wb, "Comment Report Cons", TESTCONS, startRow = 8) #put your DF (in this case TEST) into the sheet under your writing (row 8)  
  
hs1 <- createStyle(textDecoration = "Bold") #create a new style for heading  
  
addStyle(wb, style = hs1, rows = 8, cols = 1:50, sheet = "Comment Report Cons") #Tell it where to go. We'll do 50 columns in this case so it can grow if needed  
  
#Freeze Panes  
  
#Also check here: https://stackoverflow.com/questions/37677326/applying-style-to-all-sheets-of-a-workbook-using-openxlsx-package-in-r  
  
freezePane(wb, "Comment Report Cons", firstActiveRow = 9) #Freeze those panes. You know you want to. Tell it where to start.  
  
#Add filter  
  
addFilter(wb, "Comment Report Cons", row = 8, cols = 1:50) #Add your filter as well. If you're trying to impress, you might as well go all in :)  
  
#Now we'll do a fancy save by customizing the file name using paste0 and system time. We'll also assume this was for the previous month. You'll also need to make this path the one you want on your computer.   
  
  
saveWorkbook(wb, paste0("~/Documents/DirectoryR/UGA IOMP 2023 AA//some-pig-4/Comment ReportCons", format(floor\_date(Sys.Date()-months(1), "month"), "%B\_%Y") , ".xlsx"), overwrite = TRUE)

INTRO <- c("Honeywell Comments",  
  
 "Data Source: Glassdoor",  
  
 "Data As Of: Q3 2020",  
  
 "Prepared on: 7/12/2023",  
  
 "Prepared by: Charlotte Kilbourn")  
  
wb <- openxlsx::createWorkbook() #Create a work book  
  
  
#Comment Report  
  
addWorksheet(wb, "Comment Report Pros") #name the worksheet in Excel  
  
writeData(wb, "Comment Report Pros", INTRO) #Write your INTRO  
  
  
#Create style  
  
style1 <- createStyle(fontColour = "#327471", textDecoration = "Bold") #Choose your custom font color (https://www.rgbtohex.net/) and make it bold. Call it style1  
  
   
  
addStyle(wb, style = style1, rows= 1:5, cols = 1, sheet = "Comment Report Pros") #add this style to your worksheet. Tell it which rows and columns  
  
writeData(wb, "Comment Report Pros", TESTPROS, startRow = 8) #put your DF (in this case TEST) into the sheet under your writing (row 8)  
  
hs1 <- createStyle(textDecoration = "Bold") #create a new style for heading  
  
addStyle(wb, style = hs1, rows = 8, cols = 1:50, sheet = "Comment Report Pros") #Tell it where to go. We'll do 50 columns in this case so it can grow if needed  
  
#Freeze Panes  
  
#Also check here: https://stackoverflow.com/questions/37677326/applying-style-to-all-sheets-of-a-workbook-using-openxlsx-package-in-r  
  
freezePane(wb, "Comment Report Pros", firstActiveRow = 9) #Freeze those panes. You know you want to. Tell it where to start.  
  
#Add filter  
  
addFilter(wb, "Comment Report Pros", row = 8, cols = 1:50) #Add your filter as well. If you're trying to impress, you might as well go all in :)  
  
#Now we'll do a fancy save by customizing the file name using paste0 and system time. We'll also assume this was for the previous month. You'll also need to make this path the one you want on your computer.   
  
  
saveWorkbook(wb, paste0("~/Documents/DirectoryR/UGA IOMP 2023 AA//some-pig-4/Comment ReportPros", format(floor\_date(Sys.Date()-months(1), "month"), "%B\_%Y") , ".xlsx"), overwrite = TRUE)