Homework 8

Problem 2

The problem with this dataset is the following:

- Multiple variables are stored in one column: The Major column actually contains both degree and major information.
- Multiple observational units are stored in a cell: The Other.programming column contains multiple units which is better if we store one programming language one row.

Cleaning Process:

- I read in the data with "read.delim" and then add the id column for the future manipulating. Then I use "gsub" to do some replacement to keep different words with the same meaning in the same form.
- I then select the major column and expand it to make one major per row and create the "degree" column corresponding to the major.

Table 1: Major and Degree Information

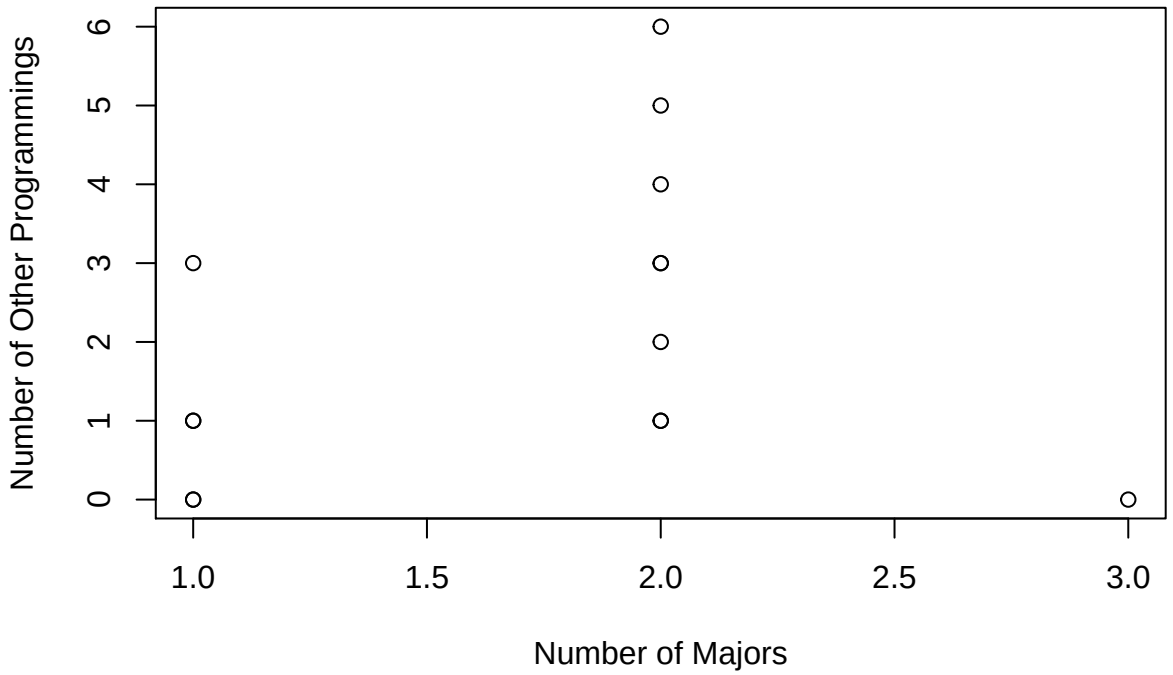
id	degree	single_major
1	BS	Math
2	BS	Math
3	BS	Finance
3	MS	Finance
4	BS	Math
4	BS	Stat

Then is not easy to also expand "Other.programming" and combine it to the degree and major dataframe so I'd like to create a cloumn for the number of other programming languages each person mastered and combined it with the dgree and major dataframe.

Table 2: Cleaned Dataset

id	degree	single_major	Platform	R.level	Num.Other.programming
1	BS	Math	PC	beg	0
2	BS	Math	Mac	beg	1
3	BS	Finance	PC	int	3
3	MS	Finance	PC	int	3
4	BS	Math	PC	int	3
4	BS	Stat	PC	int	3

Then I would like to see whether there is a relationship between the number of majors and the number of other programming languages people mastered.



From the above plot, we could see that there is actually no big influence of how many programming people know and how many majors they got.

Then I would like to see what's the popular programming languages other than R so I do the word cloud plot. It turns out SAS and Python are the most popular ones and Matlab is the second popular language.

Problem 3

In this problem I would like to use dataset Reuters21578, which is a famous dataset for text mining, to perform do some simple analysis as in case 8: mining NASA metadata (http://tidytextmining.com/nasa.html).

This dataset contains 21578 news articels from the Reuters newswire in 1987. There is a package "tm.corpus.Reuters21578" which could load this dataset into R.

Simple Exploration: Word Frequency

To begin I just did some simple word count. Since the word like "reuter","mln","dlrs","pct","cts" didn't give us too much information. I removed those words from the dataset.

Table 3: Word Frequency

word	n
mln	25565
dlrs	20567
reuter	18945
pct	17067
billion	10257
cts	8859
u.s	8650
company	8238
bank	6727
net	6064

Table 4: Word Frequecy after Remove Some Words

word	n
billion	10257
u.s	8650
company	8238
bank	6727
net	6064
corp	5596
market	5476
stock	5192
loss	5015
shares	4961

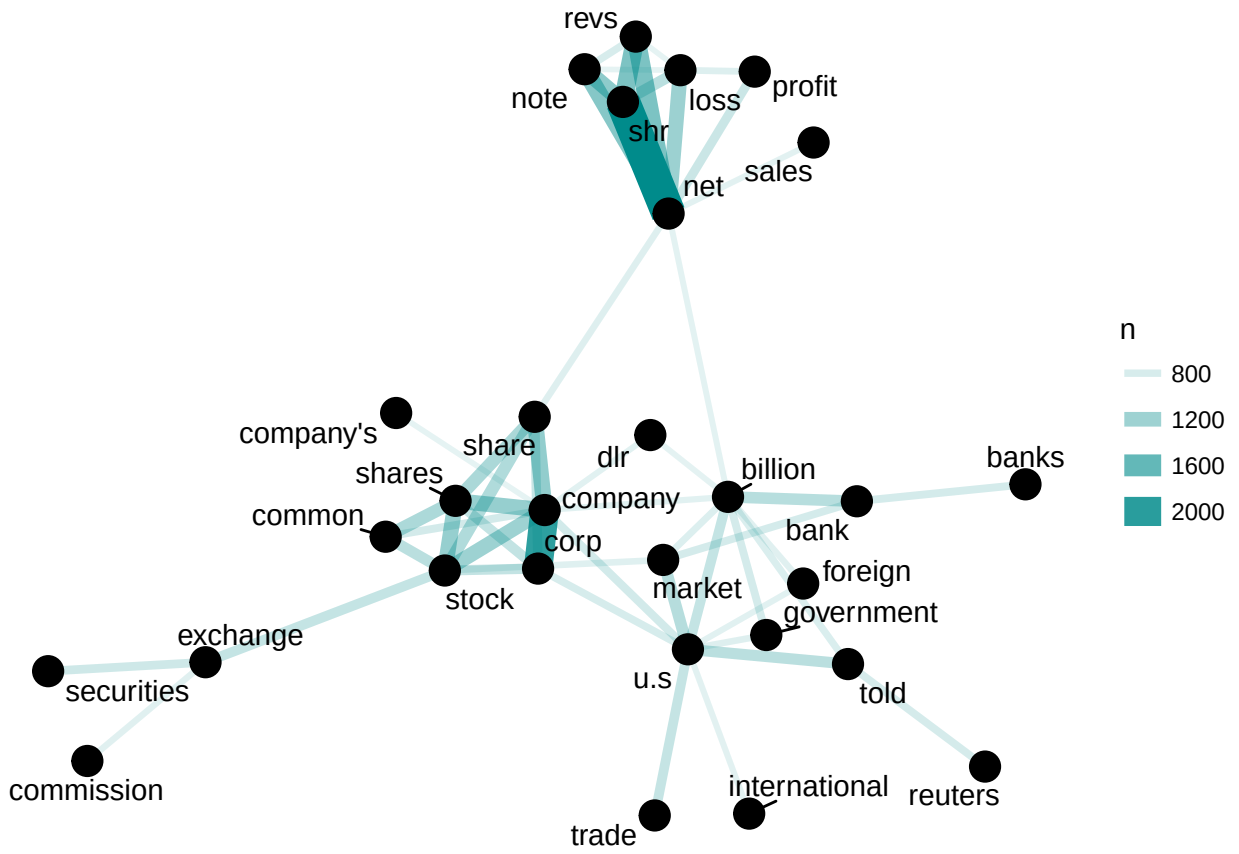
We could see the top 10 most frequeny words are related to business.

Word co-occurences

Then I just count the co-occurences of words and plot the networks of words.

Table 5: Co-occurance Frequency

item1	item2	n
shr	net	2277
corp	company	1969
net	revs	1426
shr	revs	1417
net	note	1364
shr	note	1310
company	share	1282
stock	shares	1250
company	shares	1250
net	loss	1216



This plot shows there indeed some clusters, where the top one could be something related to accounting and the left bottom could be something about business while the right bottom could be something about international affairs.

Appendix: R code

```
######### Problem 2######### read in the data
survey_data <- read.delim("survey_data.txt", header = T)
# add the id column for future manipulating substitude
# PC-Surface for PC and clean the R.level column
tidyed_data <- survey_data %>% mutate(id = row_number()) %>%
    separate(R.level, into = c("R.level", "alt"), sep = "/") %>%
    select(-alt) %>% mutate(Platform = gsub("PC-Surface",
    "PC", Platform)) %>% mutate(R.level = gsub("beginner",
    "beg", R.level)) %>% mutate(R.level = gsub("intermediate",
    "int", R.level, ignore.case = T)) %>% mutate(R.level = gsub("Int",
    "int", R.level)) %>% mutate(R.level = gsub("Intermed",
    "int", R.level, ignore.case = T))
#-------------------------------------------------------------------------
major <- as.character(survey_data$Major)
program <- toupper(as.character(survey_data$Other.programming))
## expand the multiple majors in one cell into several
## rows where one major per row create the degree columns
## corresponding to each person and major
deg_maj <- strsplit(major, split = "[,-/ ()]")  # split the major column by , - / ()
degree <- c()
```

```
id <- numeric(0)
single_major <- c()

for (i in 1:length(deg_maj)) {
    charvec <- deg_maj[[i]]
    charvec <- charvec[charvec != ""]  # delete '': empty element in the vector
    l <- length(charvec)
    if (!("BS" %in% charvec)) {
        degree <- c(degree, rep("BS", l))
        # BS isn't in the vector
        single_major <- c(single_major, charvec)
        id <- c(id, rep(i, l))
    } else if (!("MS" %in% charvec | "Master" %in% charvec)) {
        # BS in the vector but MS isn't
        degree <- c(degree, rep("BS", (l - 1)))
        single_major <- c(single_major, charvec[-l])
        id <- c(id, rep(i, (l - 1)))
    } else if (("MS" %in% charvec) | ("Master" %in% charvec)) {
        # both MS and BS in the vector
        degree <- c(degree, c("BS", "MS"))
        single_major <- c(single_major, c(charvec[1], charvec[l -
            1]))
        id <- c(id, rep(i, 2))
    }
}
#---------------------------------------------------------------------
# combine above three columns together
degree_full_data <- data.frame(id, degree, single_major) %>%
    filter(single_major != "Eng") %>% mutate(single_major = as.character(gsub("STAT",
    "Stat", single_major))) %>% mutate(degree = as.character(degree))
kable(head(degree_full_data), caption = "Major and Degree Information")
# dealing with Other.programming column
program_list <- c("MINITAB", "SAS", "MATLAB", "SQL", "PYTHON",
    "JAVA", "LINUX", "C++", "SPSS", "OBJ-C")
split_prog <- strsplit(as.character(program), split = "[,/ ]")
# create the column for number of other languages each
# person mastered
num_prog <- numeric(length(split_prog))
prog_freq <- numeric(length(program_list))  # count the frequency for each programming
for (i in 1:length(split_prog)) {
    ele <- split_prog[[i]]
    num <- 0
    for (j in 1:length(ele)) {
        if (ele[j] %in% program_list) {
            ind <- which(program_list == ele[j])
            prog_freq[ind] <- prog_freq[ind] + 1
            num <- num + 1
        }
    }
    num_prog[i] <- num
}
tidyed_data <- tidyed_data %>% mutate(Num.Other.programming = num_prog)
# left join these two dataset
```

```
full_data <- degree_full_data %>% left_join(tidyed_data) %>%
    select(-Major, -Other.programming)
kable(head(full_data), caption = "Cleaned Dataset")
analysis1 <- full_data %>% group_by(id) %>% summarise(Num.Major = n())
plot(analysis1$Num.Major, num_prog, xlab = "Number of Majors",
    ylab = "Number of Other Programmings")
program_info <- data.frame(word = program_list, freq = as.numeric(prog_freq))
wordcloud(words = program_info$word, freq = program_info$freq,
    min.freq = 1, max.words = 200, random.order = FALSE,
    rot.per = 0.35, colors = brewer.pal(8, "Dark2"))
data("Reuters21578")
l = length(Reuters21578)
content = numeric(l)
for (i in 1:l) {
    if (length(Reuters21578[[i]]$content) != 0) {
        content[i] = Reuters21578[[i]]$content
    } else {
        content[i] = 0
    }
}
del.row = which(content == 0)
content = content[-del.row]
l = length(content)
id = c(1:l)
data = data_frame(id = id, text = content)
data$text = as.character(data$text)
news_data <- data %>% unnest_tokens(word, text) %>% filter(str_detect(word,
    "[a-z']$"), !word %in% stop_words$word)
counts_data <- news_data %>% count(word, sort = T)
kable(head(counts_data, 10), caption = "Word Frequency")
my_stopwords <- data_frame(word = c("reuter", "mln", "dlrs",
    "pct", "cts"))
news_data <- news_data %>% anti_join(my_stopwords)
counts_data <- news_data %>% count(word, sort = T)
kable(head(counts_data, 10), caption = "Word Frequecy after Remove Some Words")
news_word_pairs <- news_data %>% anti_join(my_stopwords) %>%
    pairwise_count(word, id, sort = T, upper = F)
kable(head(news_word_pairs, 10), caption = "Co-occurance Frequency")
set.seed(1234)
# code is from http://tidytextmining.com/nasa.html
news_word_pairs %>% filter(n >= 700) %>% graph_from_data_frame() %>%
    ggraph(layout = "fr") + geom_edge_link(aes(edge_alpha = n,
    edge_width = n), edge_colour = "cyan4") + geom_node_point(size = 5) +
    geom_node_text(aes(label = name), repel = TRUE, point.padding = unit(0.2,
        "lines")) + theme_void()
```