data\_exploration.r

kenim

2023-05-17

# DATA EXPLORATION ----  
  
# Load required libraries ----  
library(ggplot2)  
library(corrplot)

## corrplot 0.92 loaded

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(wordcloud)

## Warning: package 'wordcloud' was built under R version 4.2.3

## Loading required package: RColorBrewer

library(tm)

## Warning: package 'tm' was built under R version 4.2.3

## Loading required package: NLP

##   
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':  
##   
## annotate

library(SnowballC)  
library(RColorBrewer)  
library(wesanderson)

## Warning: package 'wesanderson' was built under R version 4.2.3

# display numbers in natural form  
options(scipen = 999)  
  
# Load cleaned data ----  
df <- read.csv("Data/data.csv")  
  
  
# Shape and structure of the data ----  
str(df)

## 'data.frame': 2767 obs. of 16 variables:  
## $ ID : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ success : chr "N" "N" "N" "Y" ...  
## $ brandSlogan : chr "Is One of Its Kind ERC-20 Decentralized Stable Asset" "The Ultimate Blockchain Gaming Platform" "Simple Automated Investment App Driven by AI & ML" "International Real Estate Crowdfunding Platform" ...  
## $ hasVideo : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ rating : num 4 4.3 4.4 4.3 4.3 4.7 4.1 4.5 4.8 4.2 ...  
## $ priceUSD : num 30 0.13 0.01 NA 0.03 0.1 0.02 2.8 50 0.1 ...  
## $ countryRegion : chr "Singapore" "Malta" "UK" "Netherlands" ...  
## $ startDate : chr "01/10/2019" "07/09/2018" "01/07/2019" "01/10/2019" ...  
## $ endDate : chr "01/10/2019" "12/10/2018" "30/06/2020" "15/12/2019" ...  
## $ teamSize : int 31 20 10 27 14 43 20 31 8 29 ...  
## $ hasGithub : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ hasReddit : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ platform : chr "Ethereum" "XAYA" "Stellar" "Separate blockchain" ...  
## $ coinNum : num 509999 225000000 5000000000 125000000 5000000000 ...  
## $ minInvestment : int 0 1 1 1 1 1 1 1 1 1 ...  
## $ distributedPercentage: num 0.49 0.41 0.4 0.13 0.5 0.5 0.25 0.1 0.05 0.15 ...

summary(df)

## ID success brandSlogan hasVideo   
## Min. : 1.0 Length:2767 Length:2767 Min. :0.0000   
## 1st Qu.: 692.5 Class :character Class :character 1st Qu.:0.0000   
## Median :1384.0 Mode :character Mode :character Median :1.0000   
## Mean :1384.0 Mean :0.7261   
## 3rd Qu.:2075.5 3rd Qu.:1.0000   
## Max. :2767.0 Max. :1.0000   
##   
## rating priceUSD countryRegion startDate   
## Min. :1.000 Min. : 0.00 Length:2767 Length:2767   
## 1st Qu.:2.600 1st Qu.: 0.04 Class :character Class :character   
## Median :3.100 Median : 0.12 Mode :character Mode :character   
## Mean :3.121 Mean : 19.01   
## 3rd Qu.:3.700 3rd Qu.: 0.50   
## Max. :4.800 Max. :39384.00   
## NA's :180   
## endDate teamSize hasGithub hasReddit   
## Length:2767 Min. : 1.00 Min. :0.0000 Min. :0.0000   
## Class :character 1st Qu.: 7.00 1st Qu.:0.0000 1st Qu.:0.0000   
## Mode :character Median :12.00 Median :1.0000 Median :1.0000   
## Mean :13.11 Mean :0.5779 Mean :0.6328   
## 3rd Qu.:17.00 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :75.00 Max. :1.0000 Max. :1.0000   
## NA's :154   
## platform coinNum minInvestment   
## Length:2767 Min. : 12 Min. :0.0000   
## Class :character 1st Qu.: 50000000 1st Qu.:0.0000   
## Mode :character Median : 180000000 Median :0.0000   
## Mean : 8177879989180 Mean :0.4532   
## 3rd Qu.: 600000000 3rd Qu.:1.0000   
## Max. :22619078416800000 Max. :1.0000   
##   
## distributedPercentage  
## Min. : 0.000   
## 1st Qu.: 0.400   
## Median : 0.550   
## Mean : 1.061   
## 3rd Qu.: 0.700   
## Max. :869.750   
##

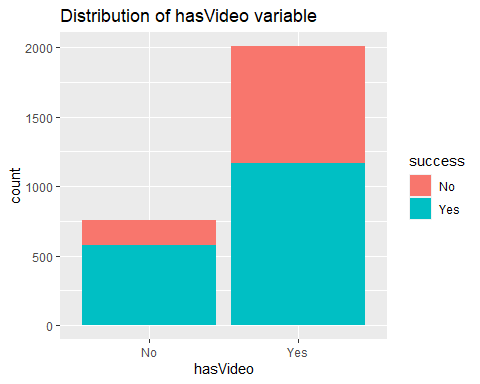
# Factor success variable  
df$success <- factor(df$success, levels = c("Y", "N"), labels = c("Yes", "No"))  
  
summary(df$success)

## Yes No   
## 1028 1739

# hasVideo  
df$hasVideo <- factor(df$hasVideo)  
summary(df$hasVideo)

## 0 1   
## 758 2009

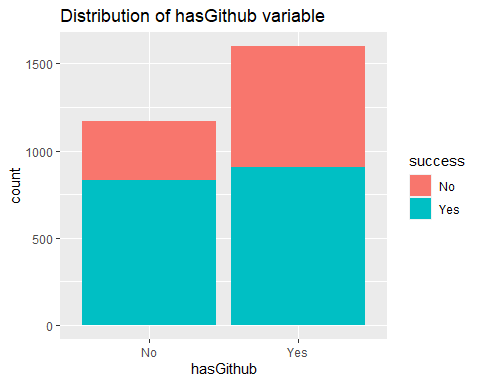
df %>% ggplot(aes(x = hasVideo, fill = success)) +  
 geom\_bar() +  
 labs(title = "Distribution of hasVideo variable") +  
 scale\_x\_discrete(labels = c("No", "Yes")) +  
 scale\_fill\_discrete(labels = c("No", "Yes"))



# hasGithub  
df$hasGithub <- factor(df$hasGithub)  
summary(df$hasGithub)

## 0 1   
## 1168 1599

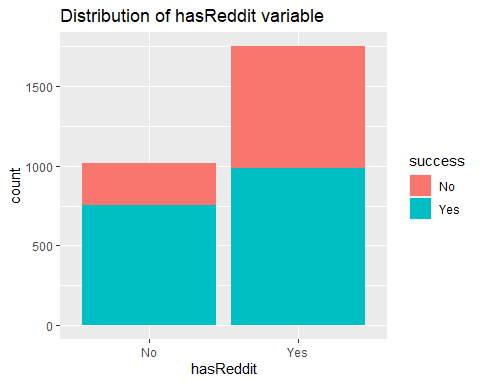
df %>% ggplot(aes(x = hasGithub, fill = success)) +  
 geom\_bar() +  
 labs(title = "Distribution of hasGithub variable") +  
 scale\_x\_discrete(labels = c("No", "Yes")) +  
 scale\_fill\_discrete(labels = c("No", "Yes"))



# hasReddit  
df$hasReddit <- factor(df$hasReddit)  
summary(df$hasReddit)

## 0 1   
## 1016 1751

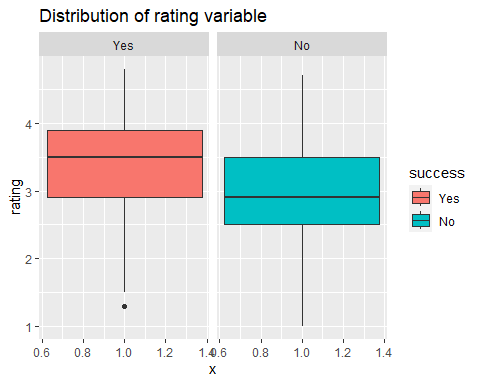
df %>% ggplot(aes(x = hasReddit, fill = success)) +  
 geom\_bar() +  
 labs(title = "Distribution of hasReddit variable") +  
 scale\_x\_discrete(labels = c("No", "Yes")) +  
 scale\_fill\_discrete(labels = c("No", "Yes"))



# rating  
df %>%  
 filter(success == "No") %>%  
 summarize(avg = mean(rating))

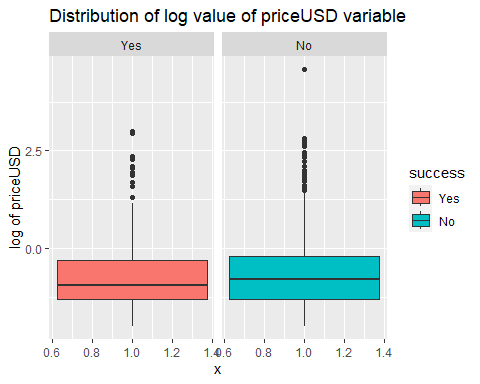
## avg  
## 1 2.965267

df %>% ggplot(aes(x = 1.0, y = rating, fill = success)) +  
 geom\_boxplot() +  
 labs(title = "Distribution of rating variable") +  
 facet\_wrap(~ success)



# priceUSD  
df %>%  
 ggplot(aes(x = 1.0, y = log(priceUSD, 10), fill = success)) +  
 geom\_boxplot() +  
 labs(title = "Distribution of log value of priceUSD variable") +  
 facet\_wrap(~ success) +  
 ylab("log of priceUSD")

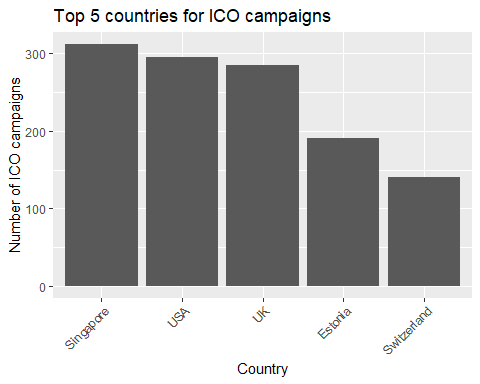
## Warning: Removed 332 rows containing non-finite values (`stat\_boxplot()`).



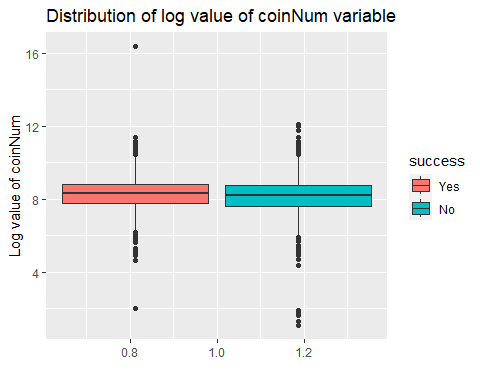
# countryRegion  
df$countryRegion <- factor(df$countryRegion)  
summary(df$countryRegion)

## Singapore USA UK   
## 312 296 285   
## Estonia Switzerland Russia   
## 191 140 138   
## Cayman Islands Germany   
## 71 67 61   
## Malta Netherlands Australia   
## 59 59 57   
## Canada British Virgin Islands France   
## 52 45 43   
## United Arab Emirates India Gibraltar   
## 41 38 36   
## Indonesia Seychelles South Korea   
## 31 31 30   
## Belize Cyprus Nigeria   
## 29 25 24   
## Romania Slovenia South Africa   
## 24 24 24   
## Czech Republic China Poland   
## 23 21 21   
## Bulgaria Ukraine Lithuania   
## 19 19 18   
## Spain Japan Georgia   
## 18 17 16   
## Israel Malaysia Turkey   
## 16 16 16   
## Ireland Latvia Philippines   
## 15 15 14   
## Thailand Italy Austria   
## 13 12 11   
## Brazil Saint Kitts and Nevis Liechtenstein   
## 11 11 10   
## Serbia Costa Rica Panama   
## 10 9 9   
## New Zealand Mauritius Mexico   
## 8 7 7   
## Norway Portugal Sweden   
## 7 7 7   
## Vietnam Belarus Croatia   
## 7 6 6   
## Luxembourg Greece Marshall Islands   
## 6 5 5   
## Belgium Bermuda Denmark   
## 4 4 4   
## Hungary Isle of Man Macedonia   
## 4 4 4   
## Anguilla Argentina Bahamas   
## 3 3 3   
## Chile Finland Afghanistan   
## 3 3 2   
## Colombia Iceland Kazakhstan   
## 2 2 2   
## Mongolia Pakistan Peru   
## 2 2 2   
## Tanzania Vanuatu Venezuela   
## 2 2 2   
## Andorra Armenia Bangladesh   
## 1 1 1   
## Barbados Bosnia and Herzegovina Cambodia   
## 1 1 1   
## Congo Curacao Curaçao   
## 1 1 1   
## Dominican Republic Ecuador Egypt   
## 1 1 1   
## French Polynesia Ghana Guinea-Bissau   
## 1 1 1   
## (Other)   
## 22

df\_countries\_five <- df %>%  
 group\_by(countryRegion) %>%  
 summarise(count = n()) %>%  
 arrange(desc(count)) %>%  
 head(5) %>%  
 as.data.frame()  
  
df\_countries\_five %>%  
 filter(countryRegion != "") %>%  
 ggplot(aes(x = reorder(countryRegion, -count), y = count)) +  
 geom\_bar(stat = "identity") +  
 ggtitle("Top 5 countries for ICO campaigns") +  
 xlab("Country") +  
 ylab("Number of ICO campaigns") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



# Number of coins  
df %>%  
 ggplot(aes(x = 1.0, y = log(coinNum, 10), fill = success)) +  
 geom\_boxplot() +  
 ggtitle("Distribution of log value of coinNum variable") +  
 xlab("") +  
 ylab("Log value of coinNum")

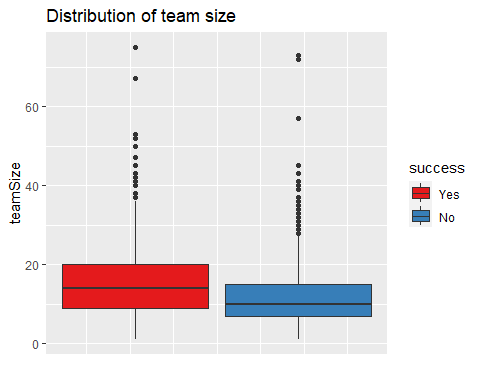


theme(axis.text.x=element\_blank(), axis.ticks.x=element\_blank())

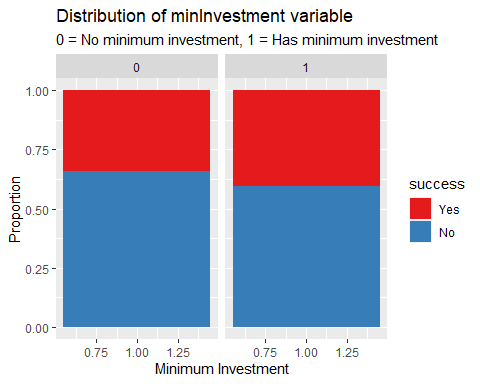
## List of 2  
## $ axis.text.x : list()  
## ..- attr(\*, "class")= chr [1:2] "element\_blank" "element"  
## $ axis.ticks.x: list()  
## ..- attr(\*, "class")= chr [1:2] "element\_blank" "element"  
## - attr(\*, "class")= chr [1:2] "theme" "gg"  
## - attr(\*, "complete")= logi FALSE  
## - attr(\*, "validate")= logi TRUE

# Team Size  
df %>%  
 ggplot(aes(x = 1.0, y = teamSize, fill = success)) +  
 geom\_boxplot() +  
 scale\_fill\_brewer(palette = "Set1") +  
 ggtitle("Distribution of team size") +  
 xlab("") +  
 theme(axis.text.x=element\_blank(), axis.ticks.x=element\_blank())

## Warning: Removed 154 rows containing non-finite values (`stat\_boxplot()`).

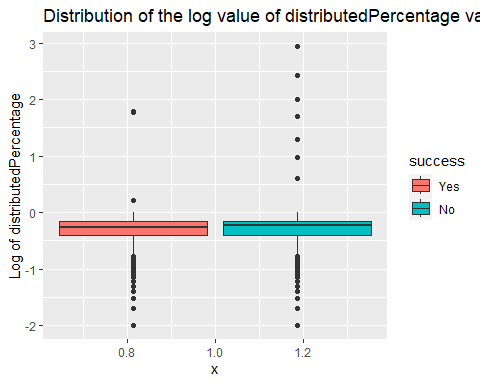


# Minimum investment  
df %>%  
 ggplot(aes(x = 1.0, y =length(minInvestment), fill = success)) +  
 geom\_col(position = "fill") +  
 facet\_wrap(~ minInvestment) +  
 scale\_fill\_brewer(palette = "Set1") +  
 ggtitle("Distribution of minInvestment variable",  
 subtitle = "0 = No minimum investment, 1 = Has minimum investment") +  
 ylab("Proportion") +  
 xlab("Minimum Investment")



# Distributed Percentage  
df %>%  
 ggplot(aes(x = 1.0, y = log(distributedPercentage, 10), fill = success)) +  
 geom\_boxplot() +  
 labs(title = "Distribution of the log value of distributedPercentage variable") +  
 ylab("Log of distributedPercentage")

## Warning: Removed 1 rows containing non-finite values (`stat\_boxplot()`).



# Common words used in ICO campaigns.  
# Create a corpus from the text column in the dataframe  
text <- c(df$brandSlogan)  
class(text)

## [1] "character"

docs <- Corpus(VectorSource(text))  
  
# Preprocessing the text data in the corpus  
docs <- docs %>%  
 tm\_map(removeNumbers) %>%  
 tm\_map(removePunctuation) %>%  
 tm\_map(stripWhitespace)

## Warning in tm\_map.SimpleCorpus(., removeNumbers): transformation drops documents

## Warning in tm\_map.SimpleCorpus(., removePunctuation): transformation drops  
## documents

## Warning in tm\_map.SimpleCorpus(., stripWhitespace): transformation drops  
## documents

docs <- tm\_map(docs, content\_transformer(tolower))

## Warning in tm\_map.SimpleCorpus(docs, content\_transformer(tolower)):  
## transformation drops documents

docs <- tm\_map(docs, removeWords, stopwords("english"))

## Warning in tm\_map.SimpleCorpus(docs, removeWords, stopwords("english")):  
## transformation drops documents

# Create a document-term matrix  
dtm <- TermDocumentMatrix(docs)   
matrix <- as.matrix(dtm)   
words <- sort(rowSums(matrix),decreasing=TRUE)   
df\_words <- data.frame(word = names(words),freq=words)  
  
# Create a word cloud  
set.seed(1234) # for reproducibility   
wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq = 10,  
 max.words=200,   
 random.order=FALSE, rot.per=0.35, colors=brewer.pal(8, "Dark2"))

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : blockchainbased could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : innovative could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : healthcare could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : international could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : investing could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : solutions could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : gambling could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : reality could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : bringing could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : cryptocurrencies could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : property could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : building could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : music could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : create could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : entertainment could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : loyalty could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : insurance could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : investments could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : every could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : powering could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : innovation could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : share could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : easy could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : experience could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : esports could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : together could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : education could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : sell could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : made could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : change could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : ultimate could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : worldwide could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : reward could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : revolutionary could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : food could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : connecting could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : development could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : changing could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : welcome could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : games could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : revolutionizing could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : storage could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : making could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : pay could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : personal could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : control could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : trusted could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : advanced could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : artificial could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : tokenized could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : adult could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : time could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : better could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : virtual could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : build could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : transparent could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : company could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = df\_words$word, freq = df\_words$freq, min.freq =  
## 10, : anywhere could not be fit on page. It will not be plotted.

