Barrett Franks R Code for Final Presentation

library(hrbrthemes)

library(ggplot2)

library(corrplot)

library(tidyverse)

library(dplyr)

library(paletteer)

library(RColorBrewer)

library(dplyr)

library(RCurl)

fname<- file.choose()

#extra<- read.table(fname, header=F, fileEncoding = "UCS-2LE")

pw <- read.delim(fname)

#pw1 <- data.frame(X0000=unlist(pw, use.names = FALSE))

names(pw)[names(pw) == 'X0000'] <- 'Name'

names(pw)[names(pw) == 'X00000000'] <- 'All\_Passwords'

head(pw, 10)

dim(pw)

colnames(pw)

str(pw)

summary(pw)

#pw1 <- data.frame(X0000=unlist(pw, use.names = FALSE))

top\_10 <- sort(table(pw$All\_Passwords),decreasing=TRUE)[1:10]

top\_10\_df <- data.frame(top\_10)

library(treemapify)

library(tidyverse)

top\_10\_df$string\_count <- paste("(", top\_10\_df$Freq, ")", sep = "")

ggplot(top\_10\_df, aes(area = Freq, fill = Var1,

label = paste(Var1, string\_count, sep = "\n"))) +

geom\_treemap() +

geom\_treemap\_text(colour = "white",

place = "centre",

size = 15) +

theme(legend.position = "none")

#library(fuzzyjoin)

#stringdist\_left\_join(pw, pw, by = c("Name" = "All\_Passwords"), method = "soundex")

library(stringr)

#pw$test <- mapply(grepl, pattern=pw$Name, x=pw$All\_Passwords)

library(dplyr)

library(data.table) # For like function (%like%)

pw1 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("john", collapse="|")))

pw2 <- pw1 %>%

filter(str\_detect(Name, str\_c("john", collapse="|")))

pw2$Lazy\_Name <- 'John'

pw2$Gender <- 'Male'

head(pw2, 10)

pw3 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("mike", collapse="|")))

pw4 <- pw3 %>%

filter(str\_detect(Name, str\_c("mike", collapse="|")))

pw4$Lazy\_Name <- 'Mike'

pw4$Gender <- 'Male'

head(pw4, 10)

pw5 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("james", collapse="|")))

pw6 <- pw5 %>%

filter(str\_detect(Name, str\_c("james", collapse="|")))

pw6$Lazy\_Name <- 'James'

pw6$Gender <- 'Male'

head(pw6, 10)

pw19 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("charles", collapse="|")))

pw20 <- pw19 %>%

filter(str\_detect(Name, str\_c("charles", collapse="|")))

pw20$Lazy\_Name <- 'Charles'

pw20$Gender <- 'Male'

head(pw20, 10)

pw7 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("william", collapse="|")))

pw8 <- pw7 %>%

filter(str\_detect(Name, str\_c("william", collapse="|")))

pw8$Lazy\_Name <- 'William'

pw8$Gender <- 'Male'

head(pw8, 10)

pw9 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("robert", collapse="|")))

pw10 <- pw9 %>%

filter(str\_detect(Name, str\_c("robert", collapse="|")))

pw10$Lazy\_Name <- 'Robert'

pw10$Gender <- 'Male'

head(pw10, 10)

pw11 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("david", collapse="|")))

pw12 <- pw11 %>%

filter(str\_detect(Name, str\_c("david", collapse="|")))

pw12$Lazy\_Name <- 'David'

pw12$Gender <- 'Male'

head(pw12, 10)

pw13 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("richard", collapse="|")))

pw14 <- pw13 %>%

filter(str\_detect(Name, str\_c("richard", collapse="|")))

pw14$Lazy\_Name <- 'Richard'

pw14$Gender <- 'Male'

head(pw14, 10)

pw15 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("joseph", collapse="|")))

pw16 <- pw15 %>%

filter(str\_detect(Name, str\_c("joseph", collapse="|")))

pw16$Lazy\_Name <- 'Joseph'

pw16$Gender <- 'Male'

head(pw16, 10)

pw17 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("thomas", collapse="|")))

pw18 <- pw17 %>%

filter(str\_detect(Name, str\_c("thomas", collapse="|")))

pw18$Lazy\_Name <- 'Thomas'

pw18$Gender <- 'Male'

head(pw18, 10)

Lazy\_Male <- rbind(pw2, pw4,pw6,pw8,pw10,pw12,pw14,pw16,pw18,pw20)

view(Lazy\_Male)

########################

pw1 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("mary", collapse="|")))

pw2 <- pw1 %>%

filter(str\_detect(Name, str\_c("mary", collapse="|")))

pw2$Lazy\_Name <- 'Mary'

pw2$Gender <- 'Female'

head(pw2, 10)

pw3 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("patricia", collapse="|")))

pw4 <- pw3 %>%

filter(str\_detect(Name, str\_c("patricia", collapse="|")))

pw4$Lazy\_Name <- 'Patricia'

pw4$Gender <- 'Female'

head(pw4, 10)

pw5 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("linda", collapse="|")))

pw6 <- pw5 %>%

filter(str\_detect(Name, str\_c("linda", collapse="|")))

pw6$Lazy\_Name <- 'Linda'

pw6$Gender <- 'Female'

head(pw6, 10)

pw19 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("barbara", collapse="|")))

pw20 <- pw19 %>%

filter(str\_detect(Name, str\_c("barbara", collapse="|")))

pw20$Lazy\_Name <- 'Barbara'

pw20$Gender <- 'Female'

head(pw20, 10)

pw7 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("elizabeth", collapse="|")))

pw8 <- pw7 %>%

filter(str\_detect(Name, str\_c("elizabeth", collapse="|")))

pw8$Lazy\_Name <- 'Elizabeth'

pw8$Gender <- 'Female'

head(pw8, 10)

pw9 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("jennifer", collapse="|")))

pw10 <- pw9 %>%

filter(str\_detect(Name, str\_c("jennifer", collapse="|")))

pw10$Lazy\_Name <- 'Jennifer'

pw10$Gender <- 'Female'

head(pw10, 10)

pw11 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("maria", collapse="|")))

pw12 <- pw11 %>%

filter(str\_detect(Name, str\_c("maria", collapse="|")))

pw12$Lazy\_Name <- 'Maria'

pw12$Gender <- 'Female'

head(pw12, 10)

pw13 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("susan", collapse="|")))

pw14 <- pw13 %>%

filter(str\_detect(Name, str\_c("susan", collapse="|")))

pw14$Lazy\_Name <- 'Susan'

pw14$Gender <- 'Female'

head(pw14, 10)

pw15 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("margaret", collapse="|")))

pw16 <- pw15 %>%

filter(str\_detect(Name, str\_c("margaret", collapse="|")))

pw16$Lazy\_Name <- 'Margaret'

pw16$Gender <- 'Female'

head(pw16, 10)

pw17 <- pw %>%

filter(str\_detect(All\_Passwords, str\_c("dorothy", collapse="|")))

pw18 <- pw17 %>%

filter(str\_detect(Name, str\_c("dorothy", collapse="|")))

pw18$Lazy\_Name <- 'Dorothy'

pw18$Gender <- 'Female'

head(pw18, 10)

Lazy\_People <- rbind(pw2, pw4,pw6,pw8,pw10,pw12,pw14,pw16,pw18,pw20, Lazy\_Male)

view(Lazy\_People)

top\_lazy <- sort(table(Lazy\_People$Lazy\_Name),decreasing=TRUE)[1:10]

top\_lazy\_df <- data.frame(top\_lazy)

library(treemapify)

library(tidyverse)

ggplot(data=top\_lazy\_df, aes(x=Var1, y=Freq)) +

geom\_bar(stat="identity", position=position\_dodge())

theme\_minimal()

pw\_test <- pw

regexp <- "[[:digit:]]+"

temp\_df <- data.frame(str\_extract(pw$All\_Passwords, regexp))

#temp\_df$temp <- temp\_df$str\_extract.pw.All\_Passwords..regexp.

colnames(temp\_df) <- c("Temp")

temp\_df <- na.omit(temp\_df)

#view(temp\_df)

head(temp\_df,20)

colnames(temp\_df)

top\_25 <- sort(table(temp\_df$Temp),decreasing=TRUE)[1:15]

top\_25\_df <- data.frame(top\_25)

head(top\_25\_df,25)

top\_25\_df$string\_count <- paste("(", top\_25\_df$Freq, ")", sep = "")

ggplot(top\_25\_df, aes(area = Freq, fill = Var1,

label = paste(Var1, string\_count, sep = "\n"))) +

geom\_treemap() +

geom\_treemap\_text(colour = "white",

place = "centre",

size = 15) +

theme(legend.position = "none")

pw$pw\_length <- str\_count(pw$All\_Passwords)

library(scales)

# Density curve

ggplot(pw, aes(x=pw\_length)) + geom\_histogram(binwidth=.75) +

xlim(c(2, 12)) +

scale\_y\_continuous(breaks= pretty\_breaks())

head(pw,100)

letters\_only <- table(grepl("^[A-Za-z]+$", pw$All\_Passwords, perl = T)) #will be false

letters\_only <- data.frame(letters\_only)

#colnames(letters\_only) <- c("T\_F")

letters\_only$What <- "Letters Only"

numbers\_only <- table(str\_detect(pw$All\_Passwords, "^[:digit:]+$")) #will be true

numbers\_only <- data.frame(numbers\_only)

#colnames(numbers\_only) <- c("T\_F")

numbers\_only$What <- "Numbers Only"

special\_only <- table(grepl('[^[:alnum:]]', pw$All\_Passwords, perl = T))

special\_only <- data.frame(special\_only)

#colnames(special\_only) <- c("T\_F")

special\_only$What <- "Special Only"

#let\_and\_num <- str.matches("[a-zA-Z0-9]\*")

all\_combo <- rbind(letters\_only, numbers\_only,special\_only)

all\_combo[nrow(all\_combo) + 1,] = c("TRUE",3668897, "Combo of All")

all\_combo <- subset(all\_combo, ! Var1 %in% c("FALSE"))

#view(all\_combo)

# Basic piechart

ggplot(all\_combo, aes(x="", y=Freq, fill=What)) +

geom\_bar(stat="identity", width=1) +

coord\_polar("y", start=0) +

theme\_void() # remove background, grid, numeric labels

library(hrbrthemes)

library(plotly)

fname<- file.choose()

hack <- read.csv(file = paste0(fname)

, header = TRUE

, stringsAsFactors = FALSE)

colnames(hack)

ggplot(hack, aes(x = Contents, y = Number.of.Characters, fill = Interpretation)) +

geom\_tile()