```
# Kayelin Santa Elena, Haley Noorani
# MGSC 410
# Final Project

##############------------------ Clustering -----------------###############

# import packages
library(forcats)
library(zoo)
library(factoextra)
library(cluster)
library(scales)
library(mondate)
library(RColorBrewer)
library(ggplot2)
library(dplyr)
library(tictoc)

# import new data
subs <- read.csv("/Users/kksizzle/Desktop/MGSC 410/final project/Rosetta (1).csv")

# train and test subset data for model testing
# set.seed(310)
# subs_indx <-  sample(1:nrow(subs), 0.5*nrow(subs), replace=FALSE)
# subs_subset <- subs[subs_indx,]

# entire dataset
subs_indx <- subs
subs_subset <- subs

#---------------- Rearange Data ----------------#

# dummy variables
subs_subset$Subscription.Type <- ifelse(subs_subset$Subscription.Type == 'Limited', 1, 0)
subs_subset$Subscription.Event.Type <- ifelse(subs_subset$Subscription.Event.Type == '
INITIAL_PURCHASE', 1, 0)
subs_subset$Purchase.Store <- ifelse(subs_subset$Purchase.Store == 'Web', 1, 0)
subs_subset$Demo.User <- ifelse(subs_subset$Demo.User == 'No', 1, 0)
subs_subset$Free.Trial.User <- ifelse(subs_subset$Free.Trial.User == 'No', 1, 0)
subs_subset$Auto.Renew <- ifelse(subs_subset$Auto.Renew == 'No', 1, 0)
subs_subset$User.Type <- ifelse(subs_subset$User.Type == 'Consumer', 1, 0)
subs_subset$Email.Subscriber <- ifelse(subs_subset$Email.Subscriber == 'No', 1, 0)
subs_subset$Push.Notifications <- ifelse(subs_subset$Push.Notifications == 'No', 1, 0)
```

```
# one hot encoding
onehotdf <- subs_subset[ , c("ID","Currency","Country","Lead.Platform")] # new df for
categorical vars

subs_subset1 <- select(subs_subset,-c("Language","Currency","Country","Lead.Platform",
                                "Subscription.Start.Date","Subscription.Expiration",
                                "Free.Trial.Start.Date", "Free.Trial.Expiration")) # new df for w/out cat.
vars. + sub/trial dates cols

# one-hot coding
library(caret)
dummies <- dummyVars(" ~ .", onehotdf)
onehotdfdummies <- data.frame(predict(dummies, newdata = onehotdf))


subs_subset2  <- data.frame(subs_subset1,onehotdfdummies) # combine new encoded vars
with the df
subs_subset2 <- select(subs_subset2, -ID) # drop ID column, not needed in cluster

#factor one-hot dummies {so that it works with kproto()}
require(plyr)
subs_subset2[,18:27] <- colwise(as.factor)(subs_subset2[,18:27])


#---------------- K prototype clustering ----------------#

#install.packages("clustMixType")
library(clustMixType)

# Check for  optimal number of clusters
tic() #timer
wss<-vector()
for (i in 2:15){ wss[i] <- sum(kproto(subs_subset2, i,na.rm = FALSE)$withinss)}

par(mfrow=c(1,1))
plot(1:15, wss, type="b", xlab="Number of Clusters",
    ylab="Within groups sum of squares",
    main="Assessing the Optimal Number of Clusters with the Elbow Method",
    pch=20, cex=2)

# apply k-prototyps
kpres <- kproto(subs_subset2, 6, na.rm = FALSE)
subs_subset2$cluster = kpres$cluster  # add cluster column to df
toc() #timer
```

```
#------new df to make graphs and plots-----#

# use this when using train and test subset
# new_subs <- subs[subs_indx,]
# df <- select(new_subs, -c("ID","Subscription.Start.Date","Subscription.Expiration",
#                           "Free.Trial.Start.Date", "Free.Trial.Expiration"))

df <- select(subs, -c("ID","Subscription.Start.Date","Subscription.Expiration",
                      "Free.Trial.Start.Date", "Free.Trial.Expiration"))

df$cluster = kpres$cluster # add cluster column to df

# save df to excel
# install.packages("writexl")
# library(writexl)
# write_xlsx(df,"/Users/kksizzle/Desktop/MGSC 410/final project/clusters.xlsx")

#------Graphs-----#

# df <- read.csv("/Users/kksizzle/Desktop/MGSC 410/final project/clusters.csv")

ggplot(df, aes(x = User.Type, fill = User.Type))+
  geom_bar(position = "dodge") +
  facet_wrap(~cluster, scale = 'free') +
  labs(x = "User Type", y = "Count",
       title = "User Type by Cluster", fill = "User Type")

ggplot(df, aes(Lead.Platform, fill = Lead.Platform)) +
  geom_bar(position = "dodge") +
  facet_wrap(~cluster, scale = 'free') +
  labs(x = "Lead Platform", y = "Count",
       title = "Lead Platform by Cluster", fill = "Lead Platform")

ggplot(df, aes(Email.Subscriber, fill = Email.Subscriber)) +
  geom_bar(position = "dodge") +
  facet_wrap(~cluster, scale = 'free') +
  labs(x = "Email Subscriber", y = "Count",
       title = "Email Subscriber by Cluster", fill = "Email Subscriber")

ggplot(df, aes(y = Open.Count, x = factor(cluster), fill = factor(cluster))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Cluster", y = "Total number of times emails were opened by subscriber in the past 90
days",
      title = "Open Count by Cluster", fill = "Cluster") +
  theme(axis.title.y = element_text(size = 8.75))

ggplot(df, aes(y = Send.Count, x = factor(cluster), fill = factor(cluster))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Cluster", y = "Number of emails sent to subscriber in the past 90 days",
      title = "Send Count by Cluster", fill = "Cluster") +
  theme(axis.title.y = element_text(size = 8.75))

ggplot(df, aes(y = Unique.Open.Count, x = factor(cluster), fill = factor(cluster))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Cluster", y = "Unique number of times emails were opened by subscriber in the past
90 days",
      title = "Unique Open Count by Cluster", fill = "Cluster") +
  theme(axis.title.y = element_text(size = 8.75))

ggplot(df, aes(Country, fill = Country)) +
  geom_bar(position = "dodge") +
  facet_wrap(~cluster, scale = 'free') +
  labs(x = "Country", y = "Count",
      title = "Country where subscriber lives by Cluster", fill = "Country")

ggplot(df, aes(x = factor(cluster), fill = Subscription.Event.Type))+
  geom_bar(position = 'fill') +
  labs(x = "Cluster", y = "Count",
      title = "Subscription Event Type by Cluster",
      fill = "Subscription Event Type")

ggplot(df, aes(User.Type, fill = Subscription.Event.Type))+
  geom_bar(position = "dodge") +
  facet_wrap(~cluster, scale = 'free') +
  labs(x = "User Type", y = "Count",
      title = "User Type and Subscription Event Type by Cluster",
      fill = "Subscription Event Type")

ggplot(df, aes(x = factor(cluster), fill = factor(cluster)))+
  geom_bar(stat = "count")  +
  labs(x = "Cluster", y = "Count",
      title = "Number of Subscribers in each Cluster",
      fill = "Cluster")
```