Hubway Case

Kun Qian

2/26/2021

```
set.seed(1234)
# Load data
hubway <- read.csv("HubwayTrips.csv")
# Take a look at the data
str(hubway)
```

```
## 'data.frame':    194302 obs. of  9 variables:
##  $ Duration : int  743 872 514 1337 493 620 1112 259 445 1155 ...
##  $ Morning  : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Afternoon: int  0 1 1 1 0 1 1 0 0 0 ...
##  $ Evening  : int  1 0 0 0 0 0 0 1 1 0 ...
##  $ Night    : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Weekday  : int  1 1 1 1 0 0 1 0 0 1 ...
##  $ Weekend  : int  0 0 0 0 1 1 0 1 1 0 ...
##  $ Male     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Age      : int  17 17 17 17 17 17 17 17 17 17 ...
```

Variable Normalizastion

i. Why normalize?

Clustering algorithm depends highly on the calculation of euclidean distance. If we don't normalize the data, the algorithm will automatically give variables that have larger scale a greater weight. However, we don't that to happen since we want to treat each variable equally so that we can use the information from all of them to determine clusters.

ii. Normalization

```
library(caret)
# Use the preprocess function to normalize the data
preprocess <- preProcess(hubway)
hubway.norm <- predict(preprocess, hubway)
# check if all columns are normalized
library(psych)
describe(hubway.norm)[3:4]
```

```
##           mean sd
## Duration     0  1
## Morning      0  1
## Afternoon    0  1
## Evening      0  1
## Night        0  1
```

```
## Weekday      0  1
## Weekend      0  1
## Male         0  1
## Age          0  1
```

K-Means

i. How many trips arer in each of the clusters?

```
# run K-means
hubwayKMC <- kmeans(hubway.norm, 10)
# Create a vector to store the clusters
hubwayKMCGroups <- hubwayKMC$cluster
table(hubwayKMCGroups)

## hubwayKMCGroups
##     1     2     3     4     5     6     7     8     9    10
## 16287 31309  9893 15638 18632 30299 26187  4827 27482 13748
```

ii. Compare the clusters

```
# Add the cluster columns to the data set
hubway$cluster <- hubwayKMCGroups
# get the mean values of the unnormalized cluster centroids
library(dplyr)
centroids.unnormalized <- hubway %>% group_by(cluster) %>%
summarise(across(everything(), list(mean)))
centroids.unnormalized

## # A tibble: 10 x 10
##    cluster Duration_1 Morning_1 Afternoon_1 Evening_1 Night_1 Weekday_1
##      <int>      <dbl>     <dbl>       <dbl>     <dbl>   <dbl>     <dbl>
##  1       1       616.  1           0           0           0     1
##  2       2       796.  0.227       0.502       0.271       0     0
##  3       3      1389.  0.000404    0.000708    0.999       0     1.00
##  4       4       757.  0           1           0           0     1
##  5       5       655.  0           1           0           0     1
##  6       6       680.  0           0           1           0     1
##  7       7       582.  1           0           0           0     1
##  8       8       749.  0           0           0           1     0.582
##  9       9       626.  0           1           0           0     1
## 10      10       716.  1           0           0           0     1
## # … with 3 more variables: Weekend_1 <dbl>, Male_1 <dbl>, Age_1 <dbl>
```

- Cluster1: Weekday morning trips by middle age male, low duration
- Cluster2: Weekend trips, mainly by male
- Cluster3: Long duration weekday trips, mainly by middle age male
- Cluster4: Weekday afternoon trips by female

- Cluster5: Weekday afternoon trips by middle age male
- Cluster6: Weekday evening trips mainly by young male
- Cluster7: Weekday morning trips by young male, low duration
- Cluster8: Night trips, mainly by young male
- Cluster9: Weekday afternoon trips by young male
- Cluster10: Weekday morning trips by female

iii. Interesting clusters

Cluster3 stand out as the duration of trips in this cluster almost doubles the duration in other clusters. Hubway might want to take a close look into this cluster and find out why the duration is so different.

iv.

I think it would be helpful to have *fewer* clusters than 10. More clusters means more granularity of the customer segments. Since we are interested in insights observed from the overall tendency within distinct user groups, having too many user groups is not helpful in summarizing trends, also they might not be that distinct as well. It's also hard for the business to implement improvement solutions if there're too many segments as it become taxing and costly to do so.

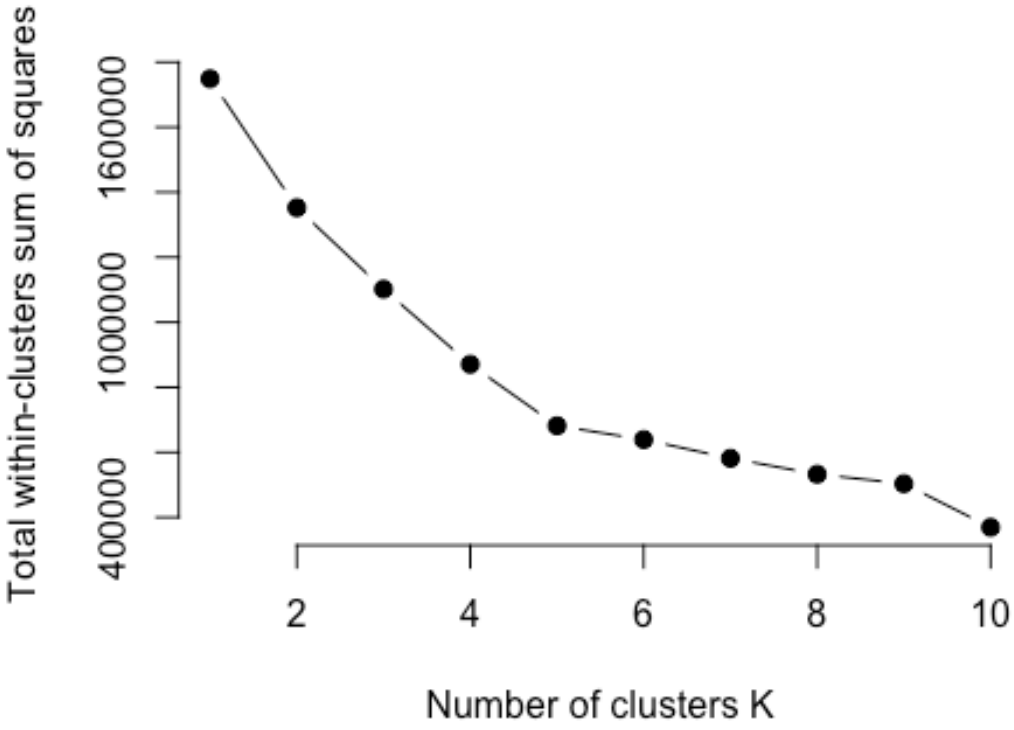
c. Try different number of clusters

```
# Apply the elbow method
# function to compute total within-cluster sum of square
wss <- function(n) {
  kmeans(hubway.norm, n, nstart = 10 )$tot.withinss
}

# Compute and plot wss for k = 1 to k = 15
k.values <- 1:10

# extract wss for 2-15 clusters
library(purrr)
wss_values <- map_dbl(k.values, wss)

plot(k.values, wss_values,
       type="b", pch = 19, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")
```



It seems like there's a tiny kink at 5 clusters and 7 clusters. Let's pick 5.

i. How many trips are in each of the cluster?

```
# run K-means
hubwayKMC5 <- kmeans(hubway.norm, 5)
# Create a vector to store the clusters
hubwayKMCGroups5 <- hubwayKMC5$cluster
table(hubwayKMCGroups5)
```

```
## hubwayKMCGroups5
##     1     2     3     4     5
## 40178 44632 30082 33333 46077
```

ii. Look at the unnormalized centorids

```
# Add the cluster columns to the data set
hubway$cluster5 <- hubwayKMCGroups5
# get the mean values of the unnormalized cluster centroids
centroids.unnormalized5 <- hubway[,-10] %>% group_by(cluster5) %>%
summarise(across(everything(), list(mean)))
centroids.unnormalized5
```

```
## # A tibble: 5 x 10
##   cluster5 Duration_1 Morning_1 Afternoon_1 Evening_1 Night_1 Weekday_1
```

```
##      <int>      <dbl>     <dbl>       <dbl>     <dbl>   <dbl>     <dbl>
## 1        1       775.     0           0         1      0              1
## 2        2       605.     0.951       0         0      0.0490         1
## 3        3       804.     0.458       0.521     0      0.0206         1
## 4        4       812.     0.214       0.471     0.255  0.0606         0
## 5        5       647.     0           1         0      0              1
## # … with 3 more variables: Weekend_1 <dbl>, Male_1 <dbl>, Age_1 <dbl>
```

- Cluster1: Weekday morning trips, 76% male, low duration
- Cluster2: Long duration weekday evening trips, 82% male, middle age
- Cluster3: Short duration weekday event trips, 72% male, young
- Cluster4: Weekday afternoon trips, 75% male
- Cluster5: Weekend trips throughout the day, 70% male

iii. Better insights

- All 5 clusters have more male than female, indicates the majority of current customers are male
- Except the cluster with abnormal duration, weekend trips tend to have slightly longer duration
- The youngest segment tend to use the service at weekday evening
- Old age users are correlated with abnormal high duration
- Weekday afternoon and evening have the most users