BACS_HW10

Description

```
Date:         2021/5/02
Student ID:   109071505
Credit Help:  109078516
```

Q1. Let's make an automated recommendation system for the PicCollage mobile app.

(a) Let's explore to see if any sticker bundles seem intuitively similar:

(a_i) Download PicCollage onto your mobile from the iOS/Android app store and take a look at the style and content of various bundles in their Sticker Store: how many recommendations does each bundle have?

```
When I click into "Bundle page", there is no block for bundle recommend
ations.
But when I click into "Search", there are 29 bundle recommendations.
```

(a_ii) Find a single sticker bundle that is both in our limited data set and also in the app's Sticker Store (e.g., "sweetmothersday"). Then, use your intuition to recommend (guess!) five other bundles in our dataset that might have similar usage patterns as this bundle.

```
I choose "autumn" as target bundle.
Recommendations (guess, according to the similarity of bundle name and
season):
1. simplyautumn
2. helloautumn
3. givethanks
4. Halloween2012StickerPack
5. halloweenparty
```

(b) Let's find similar bundles using geometric models of similarity:

(b_i) Let's create cosine similarity based recommendations for all bundles:

(b_i_1) Create a matrix or data.frame of the top 5 recommendations for all bundles.

```
# Download the CSV file.
data <- read.csv('piccollage_accounts_bundles.csv', row.names = 1, head
er=TRUE)

# Create Cosine Similarity.
library(lsa)
cos_sim <- cosine(as.matrix(data))

# List Top 5 recommendations for all bundles
Top_5_rec <- matrix(nrow = nrow(cos_sim), ncol = 6)
for (i in c(1:nrow(cos_sim)))
  Top_5_rec[i, ] <- head(names(sort(cos_sim[i, ], decreasing = TRUE)))

rownames(Top_5_rec) <- Top_5_rec[, 1] # set column 1 as row names
Top_5_rec <- Top_5_rec[, -1]  # remove column 1
colnames(Top_5_rec) <- c("1st", "2nd", "3rd", "4th", "5th") # rename co
lumn names
head(Top_5_rec) # Because there are too many bundles, I just show the r
esults of first six bundles.
```

```
##                 1st                 2nd                 3rd
## Maroon5V        "OddAnatomy"        "beatsmusic"        "xoxo"
## between         "BlingStickerPack"  "xoxo"              "gwen"
## pellington      "springrose"        "X8bit2"            "mmlm"
## StickerLite     "HeartStickerPack"  "HipsterChicSara"   "Mom2013"
## saintvalentine  "nashnext"          "givethanks"        "teenwitch"
## HipsterChicSara "Random"            "HeartStickerPack"  "wonderland"
##                 4th                 5th
## Maroon5V        "alien"             "word"
## between         "OddAnatomy"        "AccessoriesStickerPack"
## pellington      "julyfourth"        "tropicalparadise"
## StickerLite     "Emome"             "Random"
## saintvalentine  "togetherwerise"    "lovestinks2016"
## HipsterChicSara "Emome"             "StickerLite"
```

(b_i_2) Create a new function that automates the above functionality: it should take an accounts-bundles matrix as a parameter, and return a data object with the top 5 recommendations for each bundle in our data set, using cosine similarity.

```
Rec <- function(ac_bundles){
  cos_sim <- cosine(ac_bundles)

  # List Top 5 recommendations for all bundles in matrix
  Top_5_rec <- matrix(nrow = nrow(cos_sim), ncol = 6)
  for (i in c(1:nrow(cos_sim)))
    Top_5_rec[i, ] <- head(names(sort(cos_sim[i, ], decreasing = TRUE))
)

  rownames(Top_5_rec) <- Top_5_rec[, 1] # set column 1 as row names
  Top_5_rec <- Top_5_rec[, -1]  # remove column 1
  colnames(Top_5_rec) <- c("1st", "2nd", "3rd", "4th", "5th") # rename
column names
  Top_5_rec
}

Top_5_rec_func <- Rec(as.matrix(data))
head(Top_5_rec_func) # Because there are too many bundles, I just show
the results of first six bundles.
```

```
##                 1st                2nd                3rd
## Maroon5V        "OddAnatomy"       "beatsmusic"       "xoxo"
## between         "BlingStickerPack" "xoxo"             "gwen"
## pellington      "springrose"       "X8bit2"           "mmlm"
## StickerLite     "HeartStickerPack" "HipsterChicSara"  "Mom2013"
## saintvalentine  "nashnext"         "givethanks"       "teenwitch"
## HipsterChicSara "Random"           "HeartStickerPack" "wonderland"
##                 4th              5th
## Maroon5V        "alien"          "word"
## between         "OddAnatomy"     "AccessoriesStickerPack"
## pellington      "julyfourth"     "tropicalparadise"
## StickerLite     "Emome"          "Random"
## saintvalentine  "togetherwerise" "lovestinks2016"
## HipsterChicSara "Emome"          "StickerLite"
```

It's the same result as the previous question.

(b_i_3) What are the top 5 recommendations for the bundle you chose to explore earlier?

```
Top_5_rec[rownames(Top_5_rec) == "autumn", ]
```

```
##               1st               2nd               3rd               4th
##             "mmlm"       "julyfourth" "tropicalparadise"        "bestdaddy"
##               5th
##       "justmytype"
```

(b_ii) Let's create correlation based recommendations.

(b_ii_1&2) Reuse the function you created above (don't change it; don't use the cor() function), but this time give the function an accounts-bundles matrix where each bundle (column) has already been mean-centered in advance.

```
bundle_means <- apply(data, 2, mean)
bundle_means_matrix <- t(replicate(nrow(data), bundle_means))
ac_bundles_mc_b <- data - bundle_means_matrix

Top_5_rec_cor <- Rec(as.matrix(ac_bundles_mc_b))
head(Top_5_rec_cor)
```

```
##                 1st                2nd
## Maroon5V        "OddAnatomy"       "beatsmusic"
## between         "BlingStickerPack" "xoxo"
## pellington      "springrose"       "X8bit2"
## StickerLite     "HeartStickerPack" "AnimalFriendsStickerPack"
## saintvalentine  "nashnext"         "givethanks"
## HipsterChicSara "Random"           "HeartStickerPack"
##                 3rd                4th              5th
## Maroon5V        "xoxo"             "alien"          "word"
## between         "gwen"             "OddAnatomy"     "AccessoriesStickerPack"
## pellington      "tropicalparadise" "mmlm"           "julyfourth"
## StickerLite     "between"          "Emome"          "HipsterChicSara"
## saintvalentine  "teenwitch"        "togetherwerise" "lovestinks2016"
## HipsterChicSara "wonderland"       "Emome"          "StickerLite"
```

(b_ii_3) Now what are the top 5 recommendations for the bundle you chose to explore earlier?

```
Top_5_rec_cor[rownames(Top_5_rec_cor) == "autumn", ]
```

```
##          1st          2nd          3rd          4th          5th
##       "mmlm" "julyfourth"  "bestdaddy" "justmytype"   "gudetama"
```

(b_iii) Let's create adjusted-cosine based recommendations.

(b_iii_1&2) Reuse the function you created above (you should not have to change it), But this time give the function an accounts-bundles matrix where each account (row) has already been mean-centered in advance.

```
account_means <- apply(data, 1, mean)
account_means_matrix <- replicate(ncol(data), account_means)
ac_bundles_mc_a <- data - account_means_matrix

Top_5_rec_adjcos <- Rec(as.matrix(ac_bundles_mc_a))
head(Top_5_rec_adjcos)
```

```
##                 1st                2nd                3rd
## Maroon5V        "OddAnatomy"       "word"             "xoxo"
## between         "BlingStickerPack" "xoxo"             "gwen"
## pellington      "springrose"       "X8bit2"           "backtocool"
## StickerLite     "HeartStickerPack" "Mom2013"          "HipsterChicSara"
## saintvalentine  "togetherwerise"   "givethanks"       "teenwitch"
## HipsterChicSara "Random"           "HeartStickerPack" "wonderland"
##                 4th                5th
## Maroon5V        "beatsmusic"       "supercute"
## between         "Monsterhigh"      "OddAnatomy"
## pellington      "tropicalparadise" "julyfourth"
## StickerLite     "Emome"            "Random"
## saintvalentine  "mrcurlsport"      "arrows"
## HipsterChicSara "Emome"            "StickerLite"
```

(b_iii_3) What are the top 5 recommendations for the bundle you chose to explore earlier?

```
Top_5_rec_adjcos[rownames(Top_5_rec_adjcos) == "autumn", ]
```

```
##                1st                2nd                3rd                4th
## "tropicalparadise"       "julyfourth"         "gudetama"          "X8bit2"
##                5th
##  "sweetmothersday"
```

(c) (not graded) Are the three sets of geometric recommendations similar in nature (theme/keywords) to the recommendations you picked earlier using your intuition alone? What reasons might explain why your computational geometric recommendation models produce different results from your intuition?

```
No, they are totally different.
Possible reason 1: I checked the excel file and found that many bundles
 rarely used by accounts, maybe we should collect more account data.
Possible reason 2: If someone uses "autumn" bundle, then he/she would n
ot use other bundles related to "autumn" because they are the same styl
e.
```

(d) (not graded) What do you think is the conceptual difference in cosine similarity, correlation, and adjusted-cosine?

```
1. Cosine similarity: item-item collaborative filtering.
2. Correlation: based on the concept of "Mean centering bundle" (How ma
ny times did each account use bundle_j, relative to other accounts).
3. Adjusted-cosine: based on the concept of "Mean centering account" (H
ow many times did each bundle used by account_i, relative to other bund
les).
```

Q2. Correlation is at the heart of many data analytic methods so let's explore it further. For each of the scenarios below, create the described set of points in the simulation. You might have to create each scenario a few times to get a general sense of them. Visual examples of scenarios a-d are shown below.

(a) Create a horizontal set of random points, with a relatively narrow but flat distribution.

(a_i) What raw slope of x and y would you generally expect?

```
Slope will be close to "0".
```

(a_ii) What is the correlation of x and y that you would generally expect?

```
Correlation will be close to "0".
```

(b) Create a completely random set of points to fill the entire plotting area, along both x-axis and y-axis.

(b_i) What raw slope of x and y would you generally expect?

```
Slope will be close to "0".
```

(b_ii) What is the correlation of x and y that you would generally expect?

```
Correlation will be close to "0".
```

(c) Create a diagonal set of random points trending upwards at 45 degrees.

(c_i) What raw slope of x and y would you generally expect? (note that x, y have the same scale)

```
Slope will be close to "1".
```

(c_ii) What is the correlation of x and y that you would generally expect?

```
Correlation will be close to "1".
```

(d) Create a diagonal set of random trending downwards at 45 degrees.

(d_i) What raw slope of x and y would you generally expect? (note that x, y have the same scale)

```
Slope will be close to "-1".
```

(d_ii) What is the correlation of x and y that you would generally expect?

```
Correlation will be close to "-1".
```

(e) Apart from any of the above scenarios, find another pattern of data points with no correlation (r ≈ 0). (optionally: can create a pattern that visually suggests a strong relationship but produces r ≈ 0?).

```
If we draw a circle, then correlation will be close to "0", but it visu
ally suggests a strong relationship.
```

(f) Apart from any of the above scenarios, find another pattern of data points with perfect correlation (r ≈ 1). (optionally: can you find a scenario where the pattern visually suggests a different relationship?).

```
Draw several points vertically at [x = 10, y = 0 ~ 15], then draw sever
al points trending upwards at 60 degrees at [x = 30 ~ 40, y = 30 ~ 50],
 correlation will be close to "1", but the pattern visually suggests tw
o independent lines.
```

(g) Let's see how correlation relates to simple regression, by simulating any linear relationship you wish:

(g_i) Run the simulation and record the points you create: pts <- interactive_regression()

The interactive_regression() can't be run by Rmarkdown, so I record the points manually.

```
# Create 5 points by interactive_regression(), then record it manually.
pts <- data.frame(
  x = c(0.6537097, 11.3977869, 18.7711732, 28.2512413, 45.9473684),
  y = c(13.997149, 8.352103, 19.642195, 35.166073, 40.458304)
)
```

(g_ii) Use the lm() function to estimate the regression intercept and slope of pts to ensure they are the same as the values reported in the simulation plot: summary(lm(ptsy ptsx))

```
summary( lm( pts$y ~ pts$x ))
```

```
##
## Call:
## lm(formula = pts$y ~ pts$x)
##
## Residuals:
##      1      2      3      4      5
##  5.145 -8.246 -2.271  6.419 -1.046
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.3811     5.2015   1.611   0.2055
## pts$x         0.7209     0.1997   3.610   0.0365 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.878 on 3 degrees of freedom
## Multiple R-squared:  0.8129, Adjusted R-squared:  0.7505
## F-statistic: 13.03 on 1 and 3 DF,  p-value: 0.03651
```

(g_iii) Estimate the correlation of x and y to see it is the same as reported in the plot: cor(pts)

```
cor(pts)
```

```
##           x         y
## x 1.0000000 0.9015887
## y 0.9015887 1.0000000
```

(g_iv) Now, re-estimate the regression using standardized values of both x and y from pts

```
standardized_pts <- data.frame(
  x = (pts$x - mean(pts$x))/sd(pts$x),
  y = (pts$y - mean(pts$y))/sd(pts$y)
)

summary( lm( standardized_pts$y ~ standardized_pts$x ))

##
## Call:
## lm(formula = standardized_pts$y ~ standardized_pts$x)
##
## Residuals:
##       1       2       3       4       5
##  0.3737 -0.5989 -0.1649  0.4662 -0.0760
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -4.965e-17  2.234e-01    0.00   1.0000
## standardized_pts$x  9.016e-01  2.498e-01    3.61   0.0365 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4995 on 3 degrees of freedom
## Multiple R-squared:  0.8129, Adjusted R-squared:  0.7505
## F-statistic: 13.03 on 1 and 3 DF,  p-value: 0.03651

cor(standardized_pts)

##           x         y
## x 1.0000000 0.9015887
## y 0.9015887 1.0000000
```

(g_v) What is the relationship between correlation and the standardized simple-regression estimates?

```
The standardized simple-regression estimates (slope) is 0.9016, it's eq
ual to correlation.
```