BASM HW9

Answers are in black text.

Question 1) Let's make an automated recommendation system for the PicCollage app.

```
library(data.table)

library(dplyr)

library(lsa)

ac_bundles_dt <- fread("piccollage_accounts_bundles.csv")
ac_bundles_matrix <- as.matrix(ac_bundles_dt[, -1, with=FALSE])
```

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

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

There's no recommedations in my app (Android version).

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

I thought the following bundles could be recommended :

親親媽咪趣 / 暖暖媽咪趣 / 我愛家人趣 / crush on you / dear to me

b). Let's find similar bundles using geometric methods:

i. Let's create cosine similarity based recommendations for all bundles:

1. Create a matrix or data.frame of the top 5 recommendations for all bundles.

```
sim_matrix <- cosine(ac_bundles_matrix)
#str(sim_matrix)

diag(sim_matrix) <- 100 #set similarity to 100 of itself
recom_df <- data.frame(stringsAsFactors = F)
for (bundle in row.names(sim_matrix)){
  recom_df <- rbind(recom_df, names(sim_matrix[bundle ,order(sim_matrix
[bundle,],decreasing = T)])[1:6]
                          ,stringsAsFactors = F )
}
rownames(recom_df) <- row.names(sim_matrix)
recom_df <- recom_df[,-1]
colnames(recom_df) <- c("1st","2nd","3rd","4th","5th")
```

```
#View(recom_df)
recom_df %>% head(4)
```

	1st <chr>	2nd <chr>	3rd <chr>	4th <chr>	5th <chr>
Maroon5V	OddAnatomy	beatsmusic	xoxo	alien	word
between	BlingStickerPack	xoxo	gwen	OddAnatomy	AccessoriesStickerPack
pellington	springrose	8bit2	mmlm	julyfourth	tropicalparadise
StickerLite	HeartStickerPack	HipsterChicSara	Mom2013	Emome	Random

```
#Another way to create sorted name matrix

row_reco <- t(apply(sim_matrix, 1, function(x) names(sort(x,decreasing
= T) )))
```

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.

```
recom_mtMaker <- function(ac_bundles_matrix){
library(lsa)
sim_matrix <- cosine(ac_bundles_matrix)
recom_df <- data.frame(stringsAsFactors = F)
for (bundle in row.names(sim_matrix)){
  recom_df <- rbind(recom_df, names(sim_matrix[bundle ,order(sim_matrix
[bundle,],decreasing = T)])[1:6]
                          ,stringsAsFactors = F )
}
rownames(recom_df) <- row.names(sim_matrix)
recom_df <- recom_df[,-1]
colnames(recom_df) <- c("1st","2nd","3rd","4th","5th")
return(recom_df)
}

recom_mtMaker(ac_bundles_matrix) %>% head()
```

	1st <chr>	2nd <chr>	3rd <chr>	4th <chr>
Maroon5V	OddAnatomy	beatsmusic	xoxo	alien
between	BlingStickerPack	xoxo	gwen	OddAnatomy
pellington	springrose	8bit2	mmlm	julyfourth
StickerLite	HeartStickerPack	HipsterChicSara	Mom2013	Emome
saintvalentine	nashnext	givethanks	teenwitch	togetherwerise
HipsterChicSara	Random	HeartStickerPack	wonderland	Emome

3. What are the top 5 recommendations for the bundle you chose to explore earlier?

```
recom_df["sweetmothersday",]
```

	1st <chr>	2nd <chr>	3rd <chr>	4th <chr>	5th <chr>
sweetmothersday	mmlm	julyfourth	tropicalparadise	bestdaddy	justmytype

ii. Let's create correlation based recommendations.

1. Reuse the function you created above (don't change it; don't use the cor() function)

2. But this time give the function an accounts-bundles matrix where each bundle (column) has been mean-centered in advance.

```
bundles_means <- apply(ac_bundles_matrix, 2, mean)
bundles_means_matrix <- t(replicate(nrow(ac_bundles_matrix),bundles_mea
ns))
ac_bundles_mc_b <- ac_bundles_matrix - bundles_means_matrix

recom_corBased <- recom_mtMaker(ac_bundles_mc_b)
recom_corBased %>% head(4)
```

	1st <chr>	2nd <chr>	3rd <chr>	4th <chr>
Maroon5V	OddAnatomy	beatsmusic	xoxo	alien
between	BlingStickerPack	xoxo	gwen	OddAnatomy
pellington	springrose	8bit2	tropicalparadise	mmlm
StickerLite	HeartStickerPack	AnimalFriendsStickerPack	between	Emome

3. Now what are the top 5 recommendations for the bundle you chose to explore earlier?

```
recom_corBased["sweetmothersday",]
```

	1st <chr>	2nd <chr>	3rd <chr>	4th <chr>	5th <chr>
sweetmothersday	mmlm	julyfourth	bestdaddy	justmytype	gudetama

iii. Let's create adjusted-cosine based recommendations.

1. Reuse the function you created above (you should not have to change it)

2. But this time give the function an accounts-bundles matrix where each account (row) has been mean-centered in advance.

```
accounts_means <- apply(ac_bundles_matrix, 1, mean)
accounts_means_matrix <- replicate(ncol(ac_bundles_matrix),accounts_mea
ns)
ac_bundles_mc_b <- ac_bundles_matrix - accounts_means_matrix

recom_adcorBased <- recom_mtMaker(ac_bundles_mc_b)
recom_adcorBased %>% head(4)
```

	1st <chr>	2nd <chr>	3rd <chr>	4th <chr>	5th <chr>
Maroon5V	OddAnatomy	word	xoxo	beatsmusic	supercute
between	BlingStickerPack	xoxo	gwen	Monsterhigh	OddAnatomy
pellington	springrose	8bit2	backtocool	tropicalparadise	julyfourth
StickerLite	HeartStickerPack	Mom2013	HipsterChicSara	Emome	Random

3. What are the top 5 recommendations for the bundle you chose to explore earlier?

```
recom_adcorBased["sweetmothersday",]
```

	1st <chr>	2nd <chr>	3rd <chr>	4th <chr>	5th <chr>
sweetmothersday	justmytype	julyfourth	gudetama	mmlm	bestdaddy

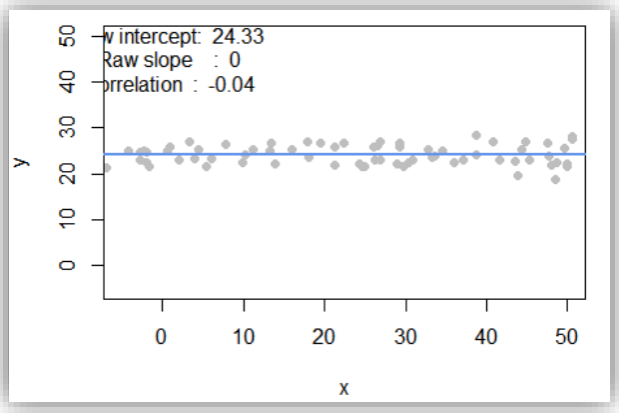
c). Compare the three sets of geometric recommendations similar in nature (theme/keywords) to the recommendations you picked earlier using your intuition. Why do you suppose they are different?

I used Android version with Chinese bundles names. Therefore, I can't tell whether geometric recommendations match the bundles I picked early. But, if there exists difference between geometric recommendations and my intuition choice, it just shows that my thought was unique.

Question 2) Correlation is at the heart of many data analytic methods so let's explore it further. For each of the scenarios below, create a set of points matching the description. You might have to create each scenario a few times to get a general sense of each. Visual examples of the first four scenarios is shown below.

```
source("demo_simple_regression.R")
interactive_regression()
```

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



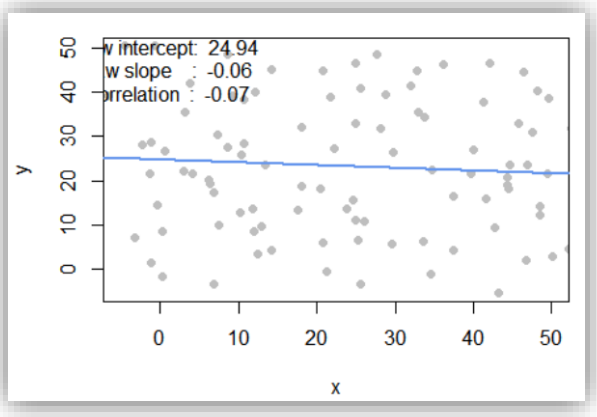
i. What raw slope of the x and y would you generally expect?

The slope of x on y is approximate to zero.

ii. What is the correlation of x and y that you would generally expect?

Since the points lie horizontally, I would generally expect the correlation to be zero.

b). Create a completely random set of points ranging all along the entire x-axis and y-axis (i.e., fill the plot)



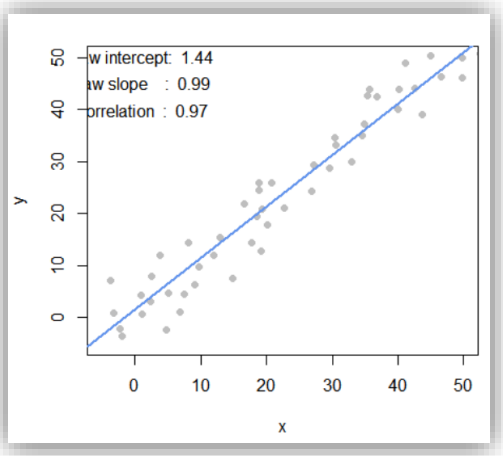
i. What raw slope of the x and y would you generally expect?

About Zero.

ii. What is the correlation of x and y that you would generally expect?

About Zero. Because we randomly pointed the data on the canvas.

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



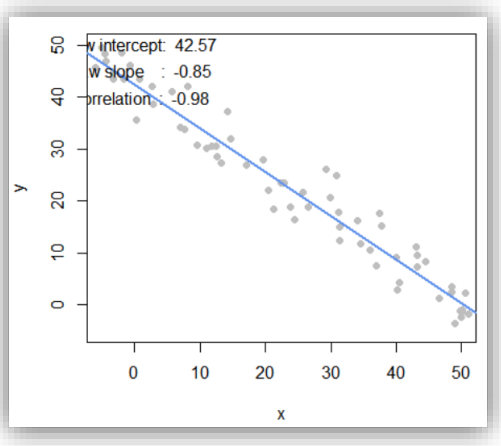
i. What raw slope of the x and y would you generally expect? (note that x, y have the same scale)

The slope should be one.

ii. What is the correlation of x and y that you would generally expect?

The values of x have the same positive trend with y. So the correlation should be close to one.

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



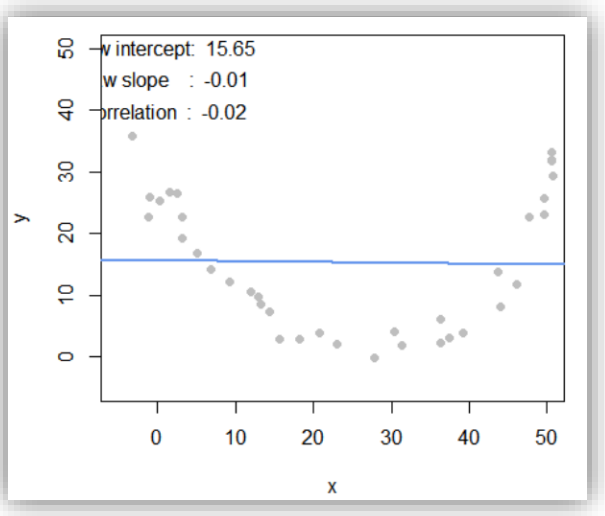
i. What raw slope of the x and y would you generally expect? (note that x, y have the same scale)

Negative one. Because the trend is 45 degrees downwards.

ii. What is the correlation of x and y that you would generally expect?

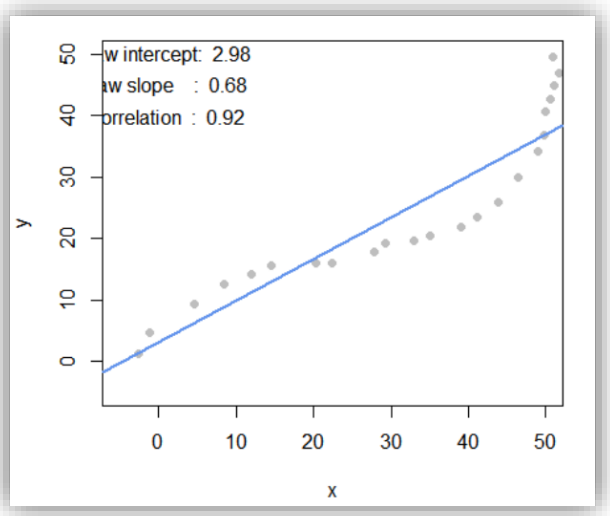
x and y have an opposite trend. Therefore, the correlation might close to negative one.

e). Apart from any of the above scenarios, find another pattern of data points with no correlation ($r \approx 0$).(challenge: can you find a scenario where the pattern visually suggests a relationship?)



The correlation is close to zero but it's easily to find out its pattern.

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



The correlation is close to 1. Also, the pattern visually suggests a different relationship.

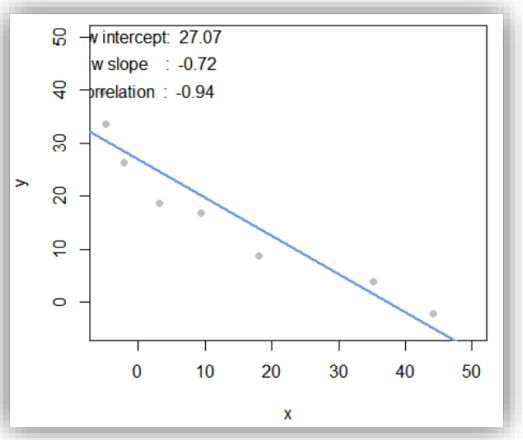
g). Let's find the relationship between correlation and regression

i. Run the simulation and capture the points you create:

pts <- interactive_regression()

```
#pts <- interactive_regression()
#for better format the report here I hard code the data point.
```

```
pts <- data.frame(x = c(-5.393789,-4.913414,-2.191287,9.337717,17.984473,35.117856,44.084861,3.09284),
                  y = c(39.791928,33.717480,26.428143,16.911508,8.912244,3.952686,-2.121762,18.733842))
```



ii. Estimate the regression intercept and slope of pts to ensure they are the same as the values reported in the simulation plot: summary(lm(pts*y* ~ pts*x*))

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

```
Call:
lm(formula = pts$y ~ pts$x)

Residuals:
    Min      1Q  Median      3Q     Max
-6.1092 -3.8429  0.0308  2.8115  8.8026

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  27.0830     2.3536  11.507 2.59e-05 ***
pts$x        -0.7242     0.1101  -6.577 0.000592 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.479 on 6 degrees of freedom
Multiple R-squared:  0.8782,    Adjusted R-squared:  0.8579
F-statistic: 43.26 on 1 and 6 DF,  p-value: 0.0005925
```

Yes, they're the same value as shown in above figure.

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 -0.9371249
## y -0.9371249  1.0000000
```

Yes, it is almost the same as reported in the plot.

iv. Now, re-estimate the regression using standardized values of both x and y from pts

```
pts_sd <- data.frame(x=scale(pts$x), y=scale(pts$y))
summary( lm( pts_sd$y ~ pts_sd$x ))
```

```
Call:
lm(formula = pts_sd$y ~ pts_sd$x)

Residuals:
     Min       1Q   Median       3Q      Max
-1.38110 -0.41535 -0.08625  0.46280  1.63343

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.860e-18  1.090e-01   0.000        1
pts_sd$x    -7.163e-01  1.103e-01  -6.492 9.62e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7065 on 40 degrees of freedom
Multiple R-squared:  0.5131,    Adjusted R-squared:  0.5009
F-statistic: 42.14 on 1 and 40 DF,  p-value: 9.624e-08
```

The Intercept and slope have changed using standardized data.

```
cor(pts_sd)
```

```
##             x           y
## x  1.0000000 -0.9371249
## y -0.9371249  1.0000000
```

The correlation of x and y remain the same.

v. What is the relationship between correlation and the standardized regression estimates?

The regression coefficient is the correlation and the intercept is 0.