BACS HW - Week 9

108070023

Question 1) 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:

i. ***(recommended)*** Download PicCollage onto your mobile from the 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?

Ans: There are 6 recommendations for each bundle.

ii. Find a single sticker bundle that is both in our limited data set and also in the app's Sticker Store

Ans: I'll choose X'mas Sketches(col: xmassketches),and recommend these 5 bundles: xmasquotes, Xmas2012StickerPack, christmassnow, newyearsparty, snowflakeee since these 5 bundles are all related to same theme(Christmas).

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

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

```
> #1
> #Import data
> #install.packages("data.table")
> library(data.table)
> ac_bundles_dt<-fread("piccollage_accounts_bundles.csv")
> ac_bundles_matrix<-as.matrix(ac_bundles_dt[,-1,with=FALSE])
> #1-b-i
> #create a matrix for top 5 recommendations for all bundles
> recom<-matrix(0,nrow=5,ncol =length(ac_bundles_matrix[1,]) ,dimnames =
list(c("rec1","rec2","rec3","rec4","rec5"),colnames(ac_bundles_matrix)))
> recom[,1:5]
     Maroon5V between pellington StickerLite saintvalentine
rec1        0       0          0           0              0
rec2        0       0          0           0              0
rec3        0       0          0           0              0
rec4        0       0          0           0              0
rec5        0       0          0           0              0
> #Write a function to calculate cosine similarity
> cossim<-function(matrix){
```

```
+   return<-matrix(0,nrow=6,ncol =length(ac_bundles_matrix[1,]) ,dimnames =
list(c("rec1","rec2","rec3","rec4","rec5","rec6"),colnames(ac_bundles_matri
x)))
+   for (i in 1:length(ac_bundles_matrix[1,])){
+     a<-matrix[,i]
+     cos<-c()#create cosine similarity matrix
+     for(j in 1:length(ac_bundles_matrix[1,])){
+       b<-matrix[,j]
+       cos<-c(cos,sum(a*b)/(sqrt(sum(a^2))*sqrt(sum(b^2))))#compute cosine
similarity
+     }
+
return[,i]=colnames(ac_bundles_matrix)[order(cos,decreasing=TRUE)[1:6]]
#put the top 1-6 bundles into recommendation matrix
+   }
+   return[] #return recommendation matrix
+ }
> top6<-cossim(ac_bundles_matrix)
> top6[,'xmassketches']
             rec1              rec2              rec3
   "pacmanholiday"      "vintagexmas"       "yummyfood"
             rec4              rec5              rec6
"watercolorywinter"    "wordstoliveby"     "helloautumn"
> recom<-top6[1:5,] #"xmassketches" itself doesn't appear in top 6
> recom[,"xmassketches"]
             rec1              rec2              rec3
   "pacmanholiday"      "vintagexmas"       "yummyfood"
             rec4              rec5
"watercolorywinter"    "wordstoliveby"
```

Ans: According to cosine similarity, the top 5 recommendations for `'xmassketches'` is:`"pacmanholiday", "vintagexmas", "yummyfood", "watercolorywinter", "wordstoliveby".`

ii. Let's create *correlation* based recommendations.

```
> #1-b-ii
> #create a matrix for top 5 recommendations for all bundles
> recom<-matrix(0,nrow=5,ncol =length(ac_bundles_matrix[1,]) ,dimnames =
list(c("rec1","rec2","rec3","rec4","rec5"),colnames(ac_bundles_matrix)))
```

```
> #Some adujstment on correlation
> col_means <- apply(ac_bundles_matrix, 2, mean)
> col_means_matrix <- t(replicate(nrow(ac_bundles_matrix),col_means))
> ac_bundles_matrix_cor <- ac_bundles_matrix - col_means_matrix
> top6 <- cossim(ac_bundles_matrix_cor)
> top6[,'xmassketches']
             rec1              rec2              rec3
   "pacmanholiday"      "vintagexmas"       "yummyfood"
             rec4              rec5              rec6
"watercolorywinter"    "wordstoliveby"     "helloautumn"
> recom<-top6[1:5,]#"xmassketches" itself doesn't appear in top 6
> recom[,"xmassketches"]
             rec1              rec2              rec3
   "pacmanholiday"      "vintagexmas"       "yummyfood"
             rec4              rec5
"watercolorywinter"    "wordstoliveby"
```

Ans: According to correlation similarity, the top 5 recommendations for 'xmassketches'is:"pacmanholiday","vintagexmas","yummyfood","watercolorywinter", "wordstoliveby". And the result is same as cosine similarity .

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

```
#1-b-iii
> #create a matrix for top 5 recommendations for all bundles
> recom<-matrix(0,nrow=5,ncol =length(ac_bundles_matrix[1,]) ,dimnames =
list(c("rec1","rec2","rec3","rec4","rec5"),colnames(ac_bundles_matrix)))
> #Some adujstment for adjusted-cosine similarity
> account_means <- apply(ac_bundles_matrix, 1, mean)
> account_means_matrix <- replicate(ncol(ac_bundles_matrix),account_means)
> ac_bundles_matrix_adjcos <- ac_bundles_matrix - account_means_matrix
> top6 <- cossim(ac_bundles_matrix_adjcos)
> top6[,'xmassketches']
             rec1              rec2              rec3
      "dayofdead"     "xmassketches"    "summergetaway"
             rec4              rec5              rec6
"watercolorywinter"    "pacmanholiday"     "helloautumn"
> recom[1,]<-top6[1,]#"xmassketches" itself appears in top 6
> recom[2:5,]<-top6[3:6,] #remove itself
> recom[,"xmassketches"]
```

```
             rec1              rec2              rec3
      "dayofdead"    "summergetaway" "watercolorywinter"
             rec4              rec5
  "pacmanholiday"      "helloautumn"
```

Ans: According to adjusted-cosine similarity, the top 5 recommendations for `'xmassketches'` is: `"dayofdead","summergetaway","watercolorywinter", "pacmanholiday", "helloautumn"` , which is different from the above two results.

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?

Ans: No, they are totally different from the five bundles I pick by intuition, while looking at the results produced by computer, I'll still consider it reasonable, since bundles like `"summergetaway","watercolorywinter","vintagexmas","yummyfood"` still relate to the events that happen in Christmas or the feature of Christmas itself.And I think the difference between computation model and my intuition may probably be the aspect of my consideration.

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

Ans: Cosine similarity doesn't take center(mean) into consideration, while correlation and adjusted-cosine do and the centers they take are different.

Question 2) Correlation is at the heart of many data analytic methods so let's explore it further.

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

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

Ans: We'll expect the slope nearly close to 0, since it is horizontal.

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

Ans: Similarly, we'll also expect the correlation is nearly close to 0, since horizonal set indicates that the movement of x is totally unrelated to y.

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

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

Ans: I'll expect the slope to be nearly 0, since the set of points fill the entire plotting area, we can't observe any trend in it.

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

Ans: Also, the correlation of x and y will be expected to be 0. For a fixed x-axis (ex.look at x=5), there will appear many points with different y and we can see that the relationship between x and y must not be 1-1 related.

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)

Ans: We'll expect the slope of the regression line to be around 1 since the set of points is diagonal(slope=1) but they don't completely lie on the same line and upward(positive).

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

Ans: I'll expect the correlation of x and y to be >0 and <1. The set of points exhibits a positive diagonal trend, which indicates certain positive relationship between x and y, while these points don't completely lie on the same line, so correlation will be smaller than 1 and bigger than 0.

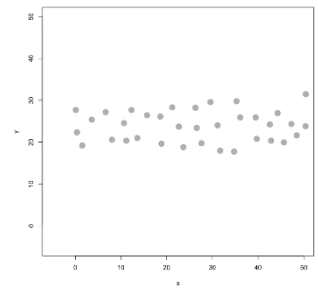
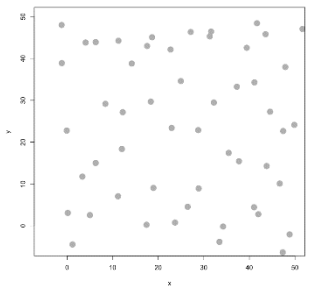
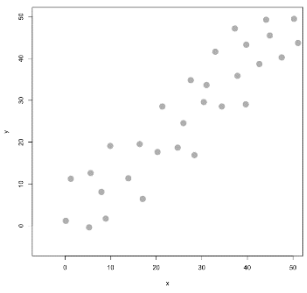
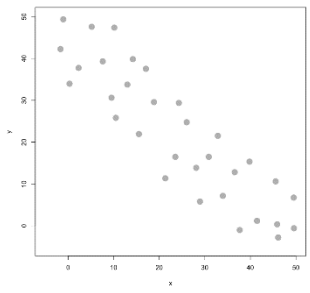
d. Create a diagonal set of random trending downwards at 45 degrees

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

Ans: I'll expect the slope of the regression line to be around -1 since the set of points is diagonal(slope=-1) but they don't completely lie on the same line.

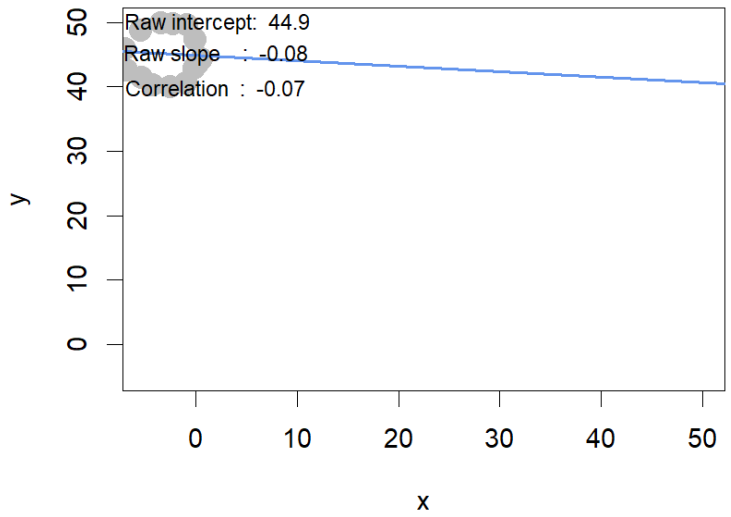
ii.What is the correlation of x and y that you would *generally* expect?

Ans: I'll expect the correlation of x and y to be >-1 and <0. The set of points exhibits a negative diagonal trend, which indicates certain negative relationship between x and y, while these points don't completely lie on the same line, so correlation will be smaller than 0 and bigger than -1.

Scenario A	*Scenario B*	*Scenario C*	*Scenario D*
			

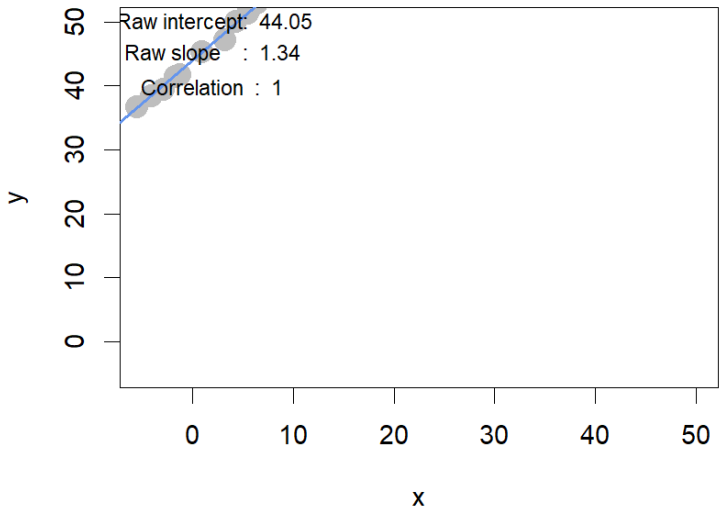
e. Apart from any of the above scenarios, find another pattern of data points with no correlation (r ≈ 0).
(can create a pattern that visually suggests a strong relationship but produces r ≈ 0*?)*

Ans: Create a pattern of points with distribution visually similar to a circle, and we can observe this relationship by eyes while its correlation of x and y is nearly 0.



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

Ans: As long as we create a set of data points that lies on the same line and the slope of line isn't 0, the correlation between x and y must be 1.



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

i. Run the simulation and record the points you create: `pts <- interactive_regression()`

(simulate either a positive or negative relationship)

```
> pts<-interactive_regression()
```

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( pts$y ~ pts$x ))`

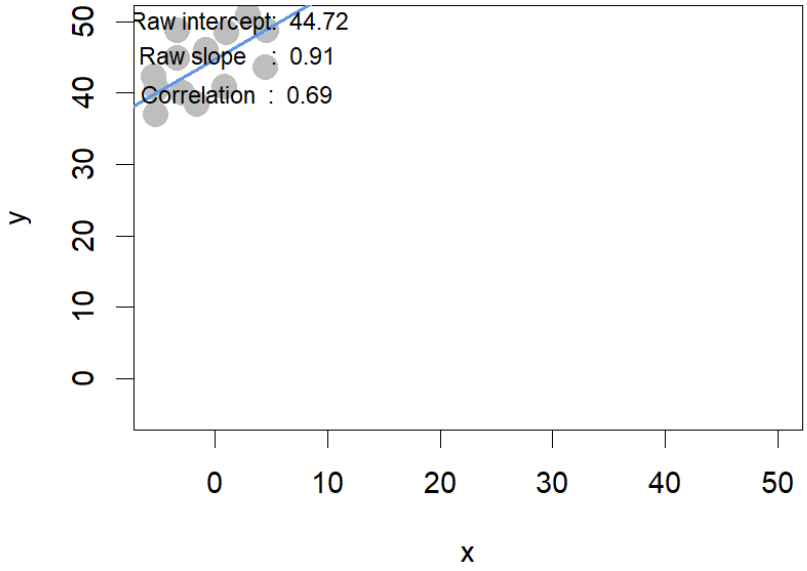
```
> #original regression line
> summary( lm( pts$y ~ pts$x ))

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

Residuals:
    Min      1Q  Median      3Q     Max
-5.0264 -3.9951  0.4059  2.9360  7.2758

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  44.7211     1.1862  37.701  7.8e-14 ***
pts$x         0.9103     0.2752   3.308  0.00625 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 4.054 on 12 degrees of freedom
Multiple R-squared:  0.4769,     Adjusted R-squared:  0.4333
F-statistic: 10.94 on 1 and 12 DF,  p-value: 0.006251
```



Ans: The regression intercept and slope of pts are same as the values reported in the simulation plot.

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 1.000000 0.690592
y 0.690592 1.000000
```

Ans: Yes, the correlation of x and y is the same as reported in the plot

iv. Now, *standardize* the values of *both* `x` and `y` from `pts` and re-estimate the regression slope

What is the relationship between *correlation* and the *standardized simple-regression estimates*?

```
#standardized
> pts$x<-scale(pts$x)
> pts$y<-scale(pts$y)
> #standardized regression line
> summary( lm( pts$y ~ pts$x ))

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

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.93335 -0.74186  0.07537  0.54519  1.35103

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.672e-17  2.012e-01   0.000  1.00000
pts$x       6.906e-01  2.088e-01   3.308  0.00625 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7528 on 12 degrees of freedom
Multiple R-squared:  0.4769,     Adjusted R-squared:  0.4333
F-statistic: 10.94 on 1 and 12 DF,  p-value: 0.006251

> cor(pts)
        x       y
x 1.000000 0.690592
y 0.690592 1.000000
```

Ans: Even the values have been standardized, the correlation of x and y still doesn’t change, and the standardized regression coefficient is the correlation and the intercept is 0.