BACS HW10 10670020

The classmate that help me: 106070004, 106070038

2021年4月29日

Question 1

```
#Q1(a)
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.0.5
```

```
data <- fread("C:/Users/eva/Desktop/作業 上課資料(清大)/大四下/BACS/HW10 BACS/piccollage_accounts_bundles.csv", header =
T)
dat_mat <- as.data.frame(data[, -1, with=FALSE])
```

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

(i) How many recommendations does each bundle have?

> Answer: Six recommendations.

(ii) Use your intuition to recommend five other bundles in our dataset that might have similar usage patterns as this bundle.

```
head(dat_mat$sweetmothersday)
```

```
## [1] 0 0 3 0 0 0
```

```
dat_mat[3, 1:20]
```

```
##   Maroon5V between pellington StickerLite saintvalentine HipsterChicSara
## 3       11       1          7          79              2               8
##   OddAnatomy wonderland V10 lovestinks2016 Random supercute retrosummer Emome
## 3          4         15   0              0     20         7           3     2
##   toMomwithLove thebouqs HeartStickerPack bubbleletters gwen food
## 3             5        4                9             0   39    8
```

> For sweet mother's day, I will recommend Mom2013, toMomwithLove, supersweet, HeartStickerPack and cutoutluv, as they are all about love, and related to the mother's day.

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

(i) Let’s create cosine similarity based recommendations for all bundles:

```
#1
# install.packages("lsa")
# install.packages("SnowballC")
library(SnowballC)
library(lsa)
```

```
## Warning: package 'lsa' was built under R version 4.0.5
```

```
data_matrix<-as.matrix(dat_mat)
cos_similar <- cosine(data_matrix)
sim_mat <- apply(cos_similar, 1, mean)
sim_mat_rank <- sim_mat[order(sim_mat, decreasing = TRUE)]
sim_mat_rank[1:5]
```

```
##     springrose eastersurprise         bemine     watercolor hipsterholiday
##      0.1578966      0.1459645      0.1383451      0.1375165      0.1368757
```

> Answer: The Top 5 recommendation are springrose, eastersurprise, bemine, watercolor and hipsterholiday.

```
#2
top5 <- function (name,data) {
  target <- data[name,]
  recom <- target[order(target, decreasing = TRUE)]
  return (recom[2:6])
}
```

```
#3

top5("sweetmothersday",cos_similar)
```

```
##             mmlm       julyfourth tropicalparadise        bestdaddy
##        0.9486833        0.9486833        0.9486833        0.9486833
##       justmytype
##        0.9486833
```

> The top 5 recommendations are mmlm, julyfourth, tropicalparadise, bestdaddy and justmytype. However, this result is totally different with my prediction.

(ii) Let’s create **correlation** based recommendations.

```
#1
bundle_means <- apply(data_matrix, 2, mean)
bundle_means_matrix <- t(replicate(nrow(data_matrix), bundle_means))
ac_bundles_mc_b <- data_matrix - bundle_means_matrix
row.names(ac_bundles_mc_b) <- row.names(data)
new_cor_similar <- cosine(ac_bundles_mc_b)
top5("sweetmothersday", new_cor_similar)
```

```
##       mmlm julyfourth  bestdaddy justmytype   gudetama
##   0.948682   0.948682   0.948682   0.948682   0.948682
```

> Answeer: The top 5 recommendations are mmlm, julyfourth, bestdaddy, justmytype, gudetama, which is similar with the answer in the last question but the order is different.

(iii) Let’s create **adjusted-cosine** based recommendations.

```
#2
library(data.table)

bundle_means <- apply(data_matrix, 1, mean)
bundle_means_matrix <- replicate(ncol(data_matrix), bundle_means)
ac_bundles_mc_b <- data_matrix - bundle_means_matrix
ad_cor_sim <- cosine(ac_bundles_mc_b)
top5("sweetmothersday", ad_cor_sim)
```

```
## justmytype julyfourth   gudetama       mmlm  bestdaddy
##  0.9984446  0.9984391  0.9984391  0.9961341  0.9961341
```

> The top 5 recommendations are justmytype, julyfourth, gudetama, mmlm, bestdaddy, the bundles are samw with the last two questions, while the orders are different.

(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?

> Answer: No, they are not similar at all. I think it is because I recommend the top 5 bundles by realizing their names' meanings.

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

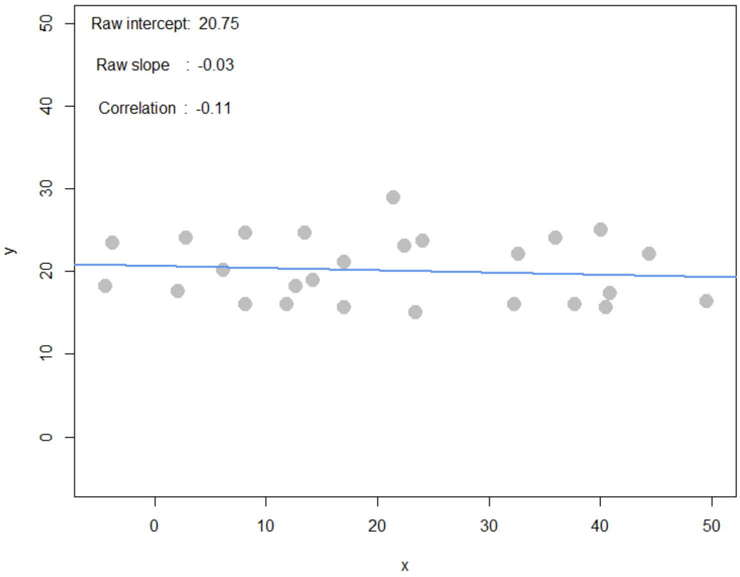
- Cosine similarity : Calculate the Cosine angle between the two vectors. The larger the angle, the more unlike the two vectors are; the smaller the angle, the more similar the two vectors are.
- correlation coefficient: It is widely used to measure the degree of linear dependence between two variables X and Y, with values between -1 and 1.
- adjusted-cosine: It is used to do the calculation for the content similarity, but in order to consider the problem of scale difference, so each will deduct an average of the score given by the user for each rating.

Q2

(a)

Expected Raw slope = 0

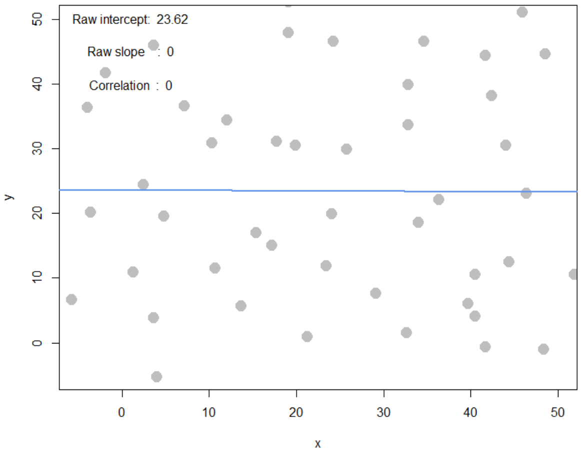
Expected correlation = 0



(b)

Expected Raw Slope = 0

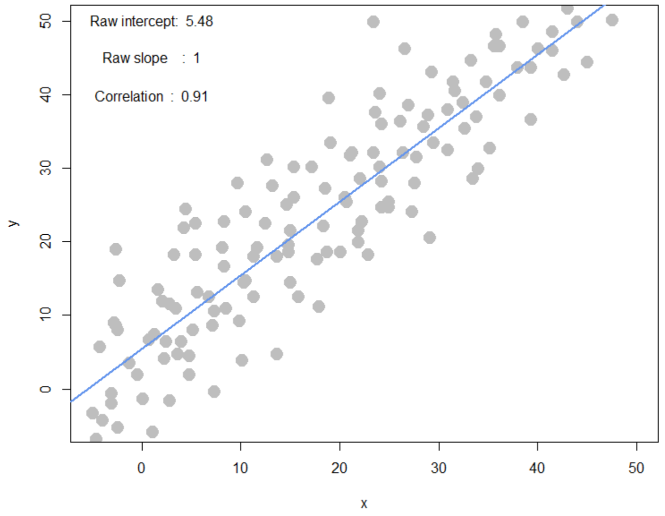
Expected correlation = 0



(c)

Expected Raw Slope = 1

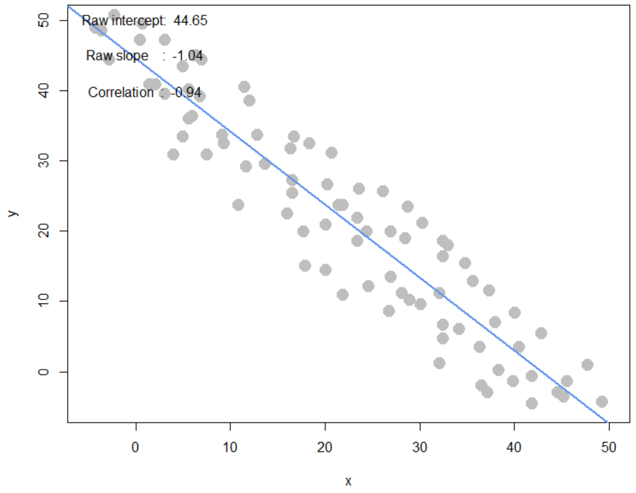
Expected correlation = 1



(d)

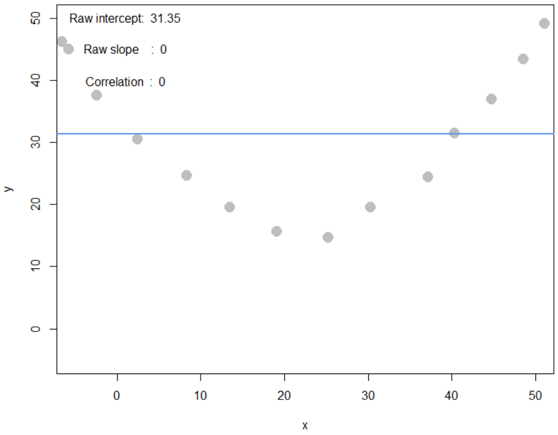
Expected Raw Slope = =-1

Expected correlation = -1



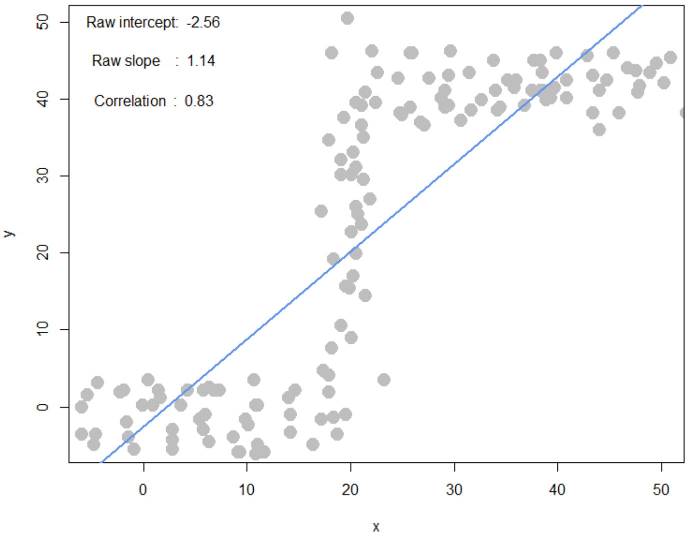
(e)

Expected correlation = 0

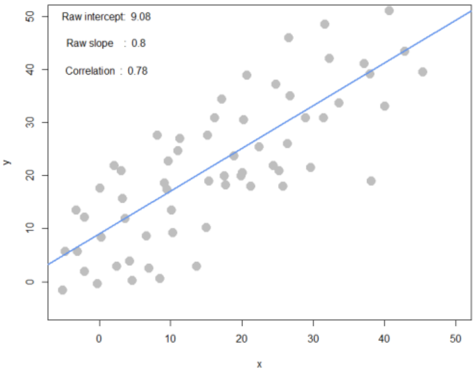


(f)

Expected correlation = 1



(g)



(i)

```
pts <- interactive_regression()
pts<-as.data.frame(pts)
```

(i)

(ii)

```
linear_model<-lm(y~x, data=pts)
summary(linear_model)
```

```
> summary(linear_model)

Call:
lm(formula = y ~ x, data = pts)

Residuals:
    Min      1Q  Median      3Q     Max 
-20.8511 -6.0679 -0.5571  7.0800 15.5215 

Coefficients:
          Estimate Std. Error t value Pr(>|t|)    
(Intercept)  9.08018    1.64868   5.508 6.23e-07 ***
x           0.80466    0.07772  10.354 1.54e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’
1

Residual standard error: 8.683 on 67 degrees of freedom
Multiple R-squared:  0.6154,    Adjusted R-squared:  0.6096 
F-statistic: 107.2 on 1 and 67 DF,  p-value: 1.538e-15
```

(iii)

```
cor(pts)
cor.test(pts$x,pts$y,method="pearson")
```

```
> cor(pts)
        x         y
x 1.0000000 0.7844672
y 0.7844672 1.0000000
> cor.test(pts$x,pts$y,method="pearson")

   Pearson's product-moment correlation

data:  pts$x and pts$y
t = 10.354, df = 67, p-value = 1.538e-15
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6726822 0.8612425
sample estimates:
     cor
0.7844672
```

In cor(pts), the correlation is 0.7844642. If using the cor.test, the correlation is also the same. Both are same with the correlation in plot.

(iv)

```
x<-(pts[,1]-mean(pts[,1]))/sd(pts[,1])
y<-(pts[,2]-mean(pts[,2]))/sd(pts[,2])
sta<-cbind(x,y)
sta<-as.data.frame(sta)
lm_sta<-lm(y~x, data=sta)
```

```
> summary(lm_sta)

Call:
lm(formula = y ~ x, data = sta)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.50039 -0.43663 -0.04009  0.50946  1.11689 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 6.706e-17  7.521e-02    0.00        1    
x           7.845e-01  7.577e-02   10.35 1.54e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 
1

Residual standard error: 0.6248 on 67 degrees of freedom
Multiple R-squared:  0.6154,    Adjusted R-squared:  0.6096 
F-statistic: 107.2 on 1 and 67 DF,  p-value: 1.538e-15
```

(v)

```
cor(sta)
cor(pts)
```

```
> cor(sta)
          x         y
x 1.0000000 0.7844672
y 0.7844672 1.0000000
> cor(pts)
          x         y
x 1.0000000 0.7844672
y 0.7844672 1.0000000
```

After the standardization, the correlation will not change.