BACS_HW_Week10_106071041

106071041

2021/5/2

```
library(data.table)
```

Question 1

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

a. Explore PicCollage

i. How many recommendations does each bundle have?

31 recommendations

ii. Use your intuition to recommend (guess!)

For “sweetmothersday”, I think the top 5 would be “lovestinks2016”, “toMomwithLove”, “HeartStickerPack”, “Mom2013”, “springrose”.

b. Find similar bundles using geometric models of similarity

i. **Cosine similarity** based recommendations for all bundles

1. Dataframe of top 5 for all bundles

```
library(lsa)
```

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

```
## Loading required package: SnowballC
```

```
index_top5 <- apply(cosine(ac_bundles_matrix), 2, function(x) sort(x, decreasing = TRUE, inde
x.return = TRUE)$ix)[2:6,]
```

```
df_top5 <- as.data.frame(apply(index_top5, 2, function(x) colnames(ac_bundles_matrix)[x]))
```

2. Create a function that automates the above funtionality

```
top5 <- function(x) {
  index_top5 <- apply(cosine(x), 2, function(y) sort(y, decreasing = TRUE, index.return = TRU
E)$ix)[2:6,]
  df_top5 <- as.data.frame(apply(index_top5, 2, function(y) colnames(x)[y]))
  df_top5
}
```

```
top5(ac_bundles_matrix)[1]
```

Maroon5V <chr>
OddAnatomy
beatsmusic
xoxo
alien
word

5 rows

3. top 5 for the bundle I chose to explore earlier

```
df_top5["sweetmothersday"]
```

sweetmothersday <chr>
mmlm
julyfourth
tropicalparadise
bestdaddy
justmytype

5 rows

ii. **Correlation** based recommendations for all bundles: What are the top 5 this time?

```
bundle_means <- apply(ac_bundles_matrix, 2 , mean)
```

```
bundle_means_matrix <- t(replicate(nrow(ac_bundles_matrix), bundle_means))
```

```
ac_bundles_mc_b <- ac_bundles_matrix - bundle_means_matrix
```

```
cor_sim <- cosine(ac_bundles_mc_b)
```

```
top5(cor_sim)["sweetmothersday"]
```

sweetmothersday <chr>
mmlm
julyfourth
bestdaddy
justmytype
gudetama

5 rows

iii. **Adjusted-cosine** based recommendations for all bundles: What are the top 5 this time?

```
bundle_means_row <- apply(ac_bundles_matrix, 1 , mean)
```

```
bundle_means_matrix_row <- replicate(ncol(ac_bundles_matrix), bundle_means_row)
```

```
ac_bundles_mc_b_row <- ac_bundles_matrix - bundle_means_matrix_row
```

```
cor_sim_row <- cosine(ac_bundles_mc_b_row)
```

```
top5(cor_sim_row)["sweetmothersday"]
```

sweetmothersday <chr>
justmytype
julyfourth
gudetama
mmlm
bestdaddy

5 rows

c. Three above-utilized recommendations method vs. Initial guess for the earlierly-picked bundles

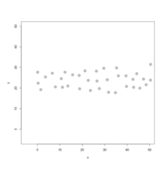
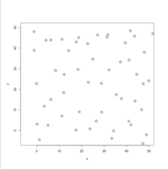
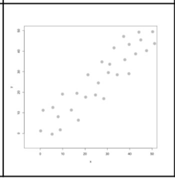
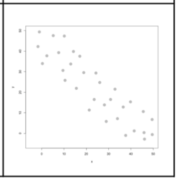
Totally different.
When just using intuition, we only pick those that we feel it related associated with the picked one, but geometric recommendations may also take the unsimilarity into consideration as well.

d. Conceptual difference in cosine similarity, correlation, and adjusted-cosine

cosine is more geometric which would care the distance of two spots while correlation and adjusted-cosine

Question 2

`demo_simple_regression.R`

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

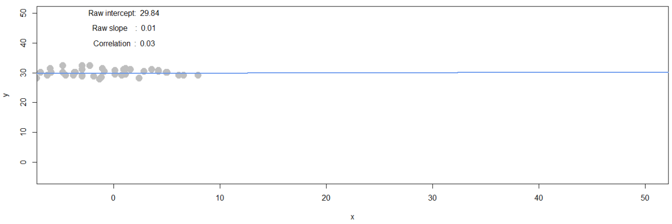
a. Scenario A

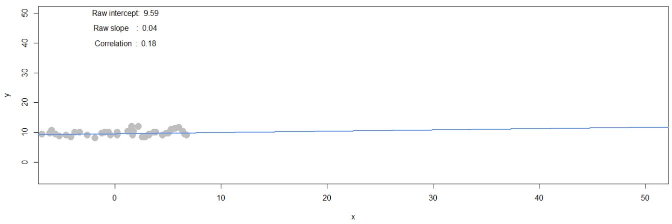
i. expected raw slope of x and y

around zero.

ii. expected correlation of x and y

around zero.





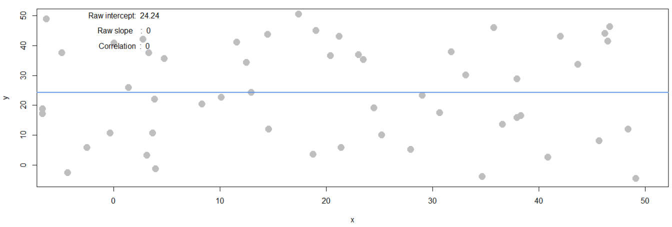
b. Scenario B

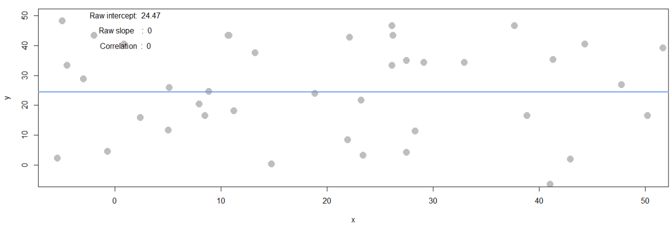
i. expected raw slope of x and y

around zero.

ii. expected correlation of x and y

around zero.





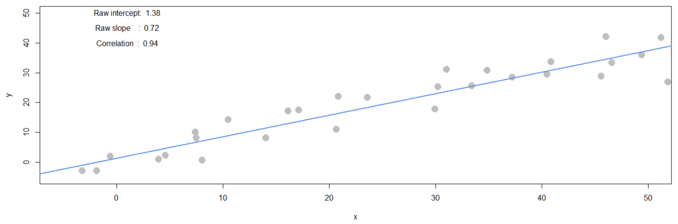
c. Scenario C

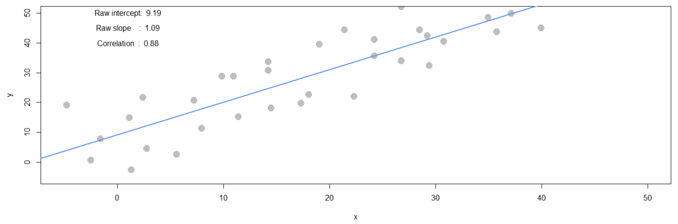
i. expected raw slope of x and y

positive slope.

ii. expected correlation of x and y

around 1.





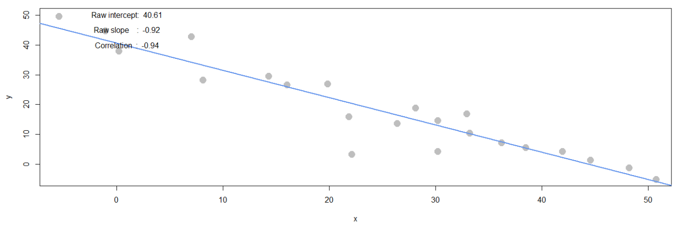
d. Scenario D

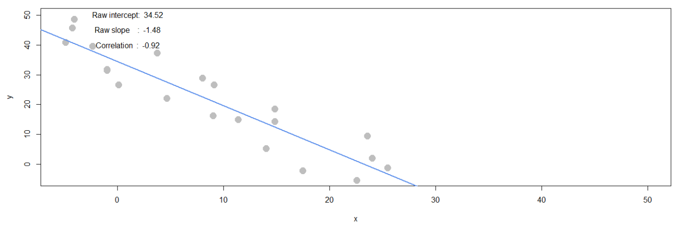
i. expected raw slope of x and y

negative slope.

ii. expected correlation of x and y

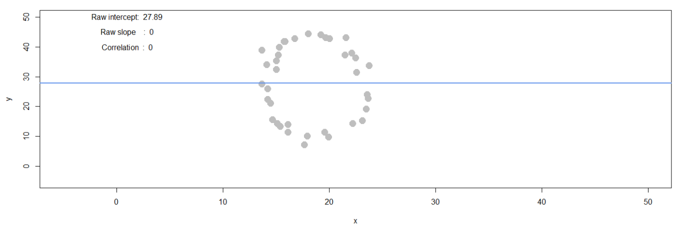
around -1.

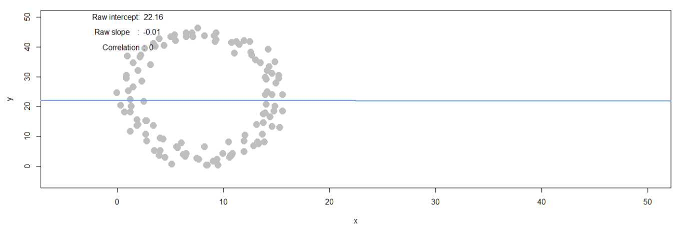




e. Find another pattern of data points with no correlation (r ≈ 0)

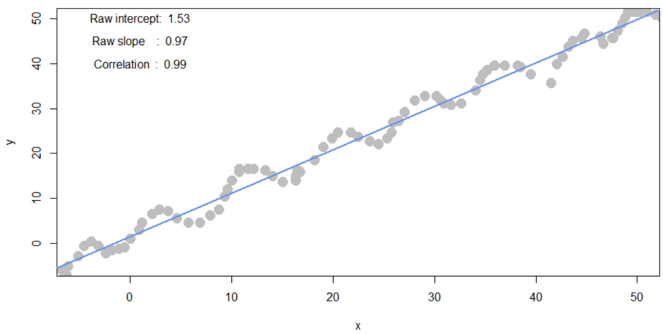
When the distribution looks like a circle.





f. Find another pattern of data points with perfect correlation (r ≈ 1)

The distribution may fluctuate but is still in the same direction.



g. Simulate wished linear relationship

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

type `pts <- interactive_regression()` in the console: `> pts <- interactive_regression()`
`Click on the plot to create data points; hit [esc] to stop`

```
pts
```

	x <dbl>	y <dbl>
	4.720359	31.94294
	8.265379	23.53484
	15.900808	33.23650
	17.809665	11.24608
	35.625666	33.23650
	36.171053	33.55989
	38.170809	26.76873
	43.897380	37.11716
	46.624319	29.03245

9 rows

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))

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

```
##
## Call:
## lm(formula = pts$y ~ pts$x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.948  -3.112   2.982   5.441   6.998
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  24.1342     5.4172   4.455  0.00295 **
## pts$x         0.1718     0.1733   0.991  0.35449
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.763 on 7 degrees of freedom
## Multiple R-squared:  0.1231, Adjusted R-squared:  -0.00213
## F-statistic: 0.983 on 1 and 7 DF,  p-value: 0.3545
```

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 0.3509071
## y 0.3509071 1.0000000
```

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

```
means <- apply(pts, 2, mean)
std <- apply(pts, 2, sd)
```

```
means
```

```
##        x        y
## 27.46505 28.85279
```

```
std
```

```
##         x         y
## 15.839069  7.754771
```

```
means_matrix <- t(replicate(nrow(pts), means))
```

```
sd_matrix <- t(replicate(nrow(pts), std))
```

```
means_matrix
```

```
##              x        y
##  [1,] 27.46505 28.85279
##  [2,] 27.46505 28.85279
##  [3,] 27.46505 28.85279
##  [4,] 27.46505 28.85279
##  [5,] 27.46505 28.85279
##  [6,] 27.46505 28.85279
##  [7,] 27.46505 28.85279
##  [8,] 27.46505 28.85279
##  [9,] 27.46505 28.85279
```

```
sd_matrix
```

```
##              x        y
##  [1,] 15.83907 7.754771
##  [2,] 15.83907 7.754771
##  [3,] 15.83907 7.754771
##  [4,] 15.83907 7.754771
##  [5,] 15.83907 7.754771
##  [6,] 15.83907 7.754771
##  [7,] 15.83907 7.754771
##  [8,] 15.83907 7.754771
##  [9,] 15.83907 7.754771
```

```
standardized <- (pts - means_matrix)/sd_matrix
```

```
standardized
```

x <dbl>	y <dbl>
-1.4359865	0.39848459
-1.2121716	-0.68576418
-0.7301086	0.56529209
-0.6095929	-2.27043545
0.5152207	0.56529209
0.5496538	0.60699397
0.6759084	-0.26874542
1.0374556	1.06571460
1.2096210	0.02316771

9 rows

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

```
##
## Call:
## lm(formula = standardized$y ~ standardized$x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0565 -0.4013  0.3845  0.7017  0.9024
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.655e-17  3.337e-01   0.000    1.000
## standardized$x 3.509e-01  3.539e-01   0.991    0.354
##
## Residual standard error: 1.001 on 7 degrees of freedom
## Multiple R-squared:  0.1231, Adjusted R-squared:  -0.00213
## F-statistic: 0.983 on 1 and 7 DF,  p-value: 0.3545
```

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

The correlation or slope would be the same but the scale are different.