# BACS HW9

108071037

2022/04/15

## Q1

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

```
#install.packages('data.table')
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.0.2
```

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

**i) How many recommendations does each bundle have?**

6 recommendations

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

I choose 'sweetmothersday'

#recommendations guesses

'Mom2013' 'toMomwithLove' 'supersweet' 'lovestinks2016' 'happybday'

by searching on key words related to mom, love, sweet and happy.

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

**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**

```
#install.packages('lsa')
library(lsa)
```

```
## Warning: package 'lsa' was built under R version 4.0.2
## Loading required package: SnowballC
## Warning: package 'SnowballC' was built under R version 4.0.2
```

```
bundles_cos <- round(cosine(ac_bundles_matrix),2)
originname <- colnames(bundles_cos)

func <- function(a){
  order(bundles_cos[,a], decreasing = TRUE)
  order <- bundles_cos[order(bundles_cos[,a], decreasing = TRUE),]
  rec5 <- row.names(order)[2:6]
  return(rec5)
```

```r
}

for (i in 1:165){
  if (i == 1){
    x <- func(i)
    newdata <- matrix(x,nrow = 5)
  }else{
    x <- func(i)
    newdata <- cbind(newdata,x)
  }
}
colnames(newdata) <- originname
as.data.frame(newdata[,1:10]) #Since printing out the whole data takes too much space, I only printed o
```

```
##     Maroon5V                between       pellington      StickerLite
## 1 OddAnatomy       BlingStickerPack       springrose HeartStickerPack
## 2      alien                   xoxo            8bit2  HipsterChicSara
## 3 beatsmusic                   gwen             mmlm            Emome
## 4       xoxo             OddAnatomy       julyfourth          Mom2013
## 5       word AccessoriesStickerPack tropicalparadise          between
##   saintvalentine  HipsterChicSara       OddAnatomy      wonderland
## 1       nashnext           Random            alien          Random
## 2     givethanks HeartStickerPack             xoxo HipsterChicSara
## 3 togetherwerise       wonderland          between        Maroon5V
## 4      teenwitch            Emome              KLL       supercute
## 5 lovestinks2016      StickerLite BlingStickerPack            gwen
##                        V10 lovestinks2016
## 1                  Mom2013       nashnext
## 2         HeartStickerPack      teenwitch
## 3              DecktheHall     givethanks
## 4               CampusLife togetherwerise
## 5 Halloween2012StickerPack  bubbleletters
```

**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.**

```r
library(lsa)
bundles_cos <- round(cosine(ac_bundles_matrix),2)

func <- function(a){
  order(bundles_cos[,a], decreasing = TRUE)
  order <- bundles_cos[order(bundles_cos[,a], decreasing = TRUE),]
  rec5 <- row.names(order)[2:6]
  return(rec5)
}
for (i in 1:165){
  if (i == 1){
    x <- func(i)
    newdata <- matrix(x,nrow = 5)
  }else{
    x <- func(i)
    newdata <- cbind(newdata,x)
  }
}
```

```r
colnames(newdata) <- originname
as.data.frame(newdata[,1:5])
```

```
##     Maroon5V                between       pellington      StickerLite
## 1 OddAnatomy       BlingStickerPack       springrose HeartStickerPack
## 2      alien                   xoxo            8bit2  HipsterChicSara
## 3 beatsmusic                   gwen             mmlm            Emome
## 4       xoxo             OddAnatomy       julyfourth          Mom2013
## 5       word AccessoriesStickerPack tropicalparadise          between
##   saintvalentine
## 1       nashnext
## 2     givethanks
## 3 togetherwerise
## 4      teenwitch
## 5 lovestinks2016
```

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

```r
num <- which(colnames(newdata) == "sweetmothersday")
newdata[,num]
```

```
## [1] "mmlm"             "julyfourth"       "tropicalparadise" "bestdaddy"
## [5] "justmytype"
```

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

**1. Reuse the function you created above**

```r
library(lsa)
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)
originname <- colnames(bundles_cos)

func_cor <- function(a){
  order(cor_sim[,a], decreasing = TRUE)
  order_cor <- cor_sim[order(cor_sim[,a], decreasing = TRUE),]
  rec5_cor <- row.names(order_cor)[2:6]
  return(rec5_cor)
}
for (i in 1:165){
  if (i == 1){
    x_cor <- func_cor(i)
    newdata_cor <- matrix(x_cor,nrow = 5)
  }else{
    x_cor <- func_cor(i)
    newdata_cor <- cbind(newdata_cor,x_cor)
  }
}
colnames(newdata_cor) <- originname
as.data.frame(newdata_cor[,1:10])
```

```
##     Maroon5V                between       pellington                StickerLite
## 1 OddAnatomy       BlingStickerPack       springrose           HeartStickerPack
## 2 beatsmusic                   xoxo            8bit2 AnimalFriendsStickerPack
## 3       xoxo                   gwen tropicalparadise                    between
```

```
## 4      alien             OddAnatomy            mmlm                    Emome
## 5       word AccessoriesStickerPack       julyfourth          HipsterChicSara
##   saintvalentine  HipsterChicSara       OddAnatomy      wonderland
## 1       nashnext           Random            alien          Random
## 2     givethanks HeartStickerPack             xoxo HipsterChicSara
## 3      teenwitch       wonderland          between        Maroon5V
## 4 togetherwerise            Emome              KLL       supercute
## 5 lovestinks2016      StickerLite BlingStickerPack            gwen
##               V10 lovestinks2016
## 1         Mom2013       nashnext
## 2 HeartStickerPack      teenwitch
## 3       CampusLife     givethanks
## 4      DecktheHall togetherwerise
## 5 BlingStickerPack  bubbleletters
```

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

```
library(lsa)
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)

func_cor <- function(a){
  order(cor_sim[,a], decreasing = TRUE)
  order_cor <- cor_sim[order(cor_sim[,a], decreasing = TRUE),]
  rec5_cor <- row.names(order_cor)[2:6]
  return(rec5_cor)
}
for (i in 1:165){
  if (i == 1){
    x_cor <- func_cor(i)
    newdata_cor <- matrix(x_cor,nrow = 5)
  }else{
    x_cor <- func_cor(i)
    newdata_cor <- cbind(newdata_cor,x_cor)
  }
}
colnames(newdata_cor) <- originname
as.data.frame(newdata_cor[,1:5])
```

```
##     Maroon5V                between       pellington              StickerLite
## 1 OddAnatomy       BlingStickerPack       springrose         HeartStickerPack
## 2 beatsmusic                   xoxo            8bit2 AnimalFriendsStickerPack
## 3       xoxo                   gwen tropicalparadise                  between
## 4      alien             OddAnatomy             mmlm                    Emome
## 5       word AccessoriesStickerPack       julyfourth          HipsterChicSara
##   saintvalentine
## 1       nashnext
## 2     givethanks
## 3      teenwitch
## 4 togetherwerise
## 5 lovestinks2016
```

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

```r
num_cor <- which(colnames(newdata_cor) == "sweetmothersday")
func_cor(num)
```

```
## [1] "mmlm"       "julyfourth" "bestdaddy"  "justmytype" "gudetama"
```

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

**1.Reuse the function you created above**

```r
library(lsa)
bundle_means_adj <- apply(ac_bundles_matrix, 1, mean)
bundle_means_adj_matrix<-replicate(ncol(ac_bundles_matrix),bundle_means_adj)
ac_bundles_mc<-ac_bundles_matrix-bundle_means_adj_matrix
cos_adj<-cosine(ac_bundles_mc)
originname <- colnames(bundles_cos)

func_adj <- function(a){
  order(cos_adj[,a], decreasing = TRUE)
  order_adj <- cos_adj[order(cos_adj[,a], decreasing = TRUE),]
  rec5_adj <- row.names(order_adj)[2:6]
  return(rec5_adj)
}
for (i in 1:165){
  if (i == 1){
    x_adj <- func_adj(i)
    newdata_adj <- matrix(x_adj,nrow = 5)
  }else{
    x_adj <- func_adj(i)
    newdata_adj <- cbind(newdata_adj,x_adj)
  }
}
colnames(newdata_adj) <- originname
as.data.frame(newdata_adj[,1:10])
```

```
##     Maroon5V          between       pellington      StickerLite saintvalentine
## 1 OddAnatomy BlingStickerPack       springrose HeartStickerPack togetherwerise
## 2       word             xoxo            8bit2          Mom2013     givethanks
## 3       xoxo             gwen       backtocool  HipsterChicSara      teenwitch
## 4 beatsmusic      Monsterhigh tropicalparadise            Emome    mrcurlsport
## 5  supercute       OddAnatomy       julyfourth           Random         arrows
##    HipsterChicSara OddAnatomy      wonderland           V10 lovestinks2016
## 1           Random       xoxo          Random christmassnow      teenwitch
## 2 HeartStickerPack      alien HipsterChicSara       cny2017     givethanks
## 3       wonderland    between            food    frombierun togetherwerise
## 4            Emome        KLL        Maroon5V floralwedding    mrcurlsport
## 5      StickerLite       word      supersweet chicchristmas       kungfood
```

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

```r
library(lsa)
bundle_means_adj <- apply(ac_bundles_matrix, 1, mean)
bundle_means_adj_matrix<-replicate(ncol(ac_bundles_matrix),bundle_means_adj)
ac_bundles_mc<-ac_bundles_matrix-bundle_means_adj_matrix
cos_adj<-cosine(ac_bundles_mc)
originname <- colnames(bundles_cos)
```

```r
func_adj <- function(a){
  order(cos_adj[,a], decreasing = TRUE)
  order_adj <- cos_adj[order(cos_adj[,a], decreasing = TRUE),]
  rec5_adj <- row.names(order_adj)[2:6]
  return(rec5_adj)
}
for (i in 1:165){
  if (i == 1){
    x_adj <- func_adj(i)
    newdata_adj <- matrix(x_adj,nrow = 5)
  }else{
    x_adj <- func_adj(i)
    newdata_adj <- cbind(newdata_adj,x_adj)
  }
}
colnames(newdata_adj) <- originname
as.data.frame(newdata_adj[,1:5])
```

```
##     Maroon5V         between       pellington      StickerLite saintvalentine
## 1 OddAnatomy BlingStickerPack       springrose HeartStickerPack togetherwerise
## 2       word             xoxo            8bit2          Mom2013     givethanks
## 3       xoxo             gwen       backtocool  HipsterChicSara      teenwitch
## 4 beatsmusic      Monsterhigh tropicalparadise            Emome    mrcurlsport
## 5  supercute       OddAnatomy       julyfourth           Random         arrows
```

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

```r
num_adj <- which(colnames(newdata_adj) == "sweetmothersday")
func_adj(num)
```

```
## [1] "justmytype" "julyfourth" "gudetama"   "mmlm"       "bestdaddy"
```

**c. 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?**

The results using geometric recommendation methods are not the same as my guesses, because we can only "guess" the results instead of calculating all the relations and compare between them.

**d. What do you think is the conceptual difference in cosine similarity, correlation, and adjusted-cosine?**

Correlation and adjusted-cosine uses the mean-centered cosine. The difference is that correlation uses the column mean while adjusted-cosine uses the row mean.

## Q2

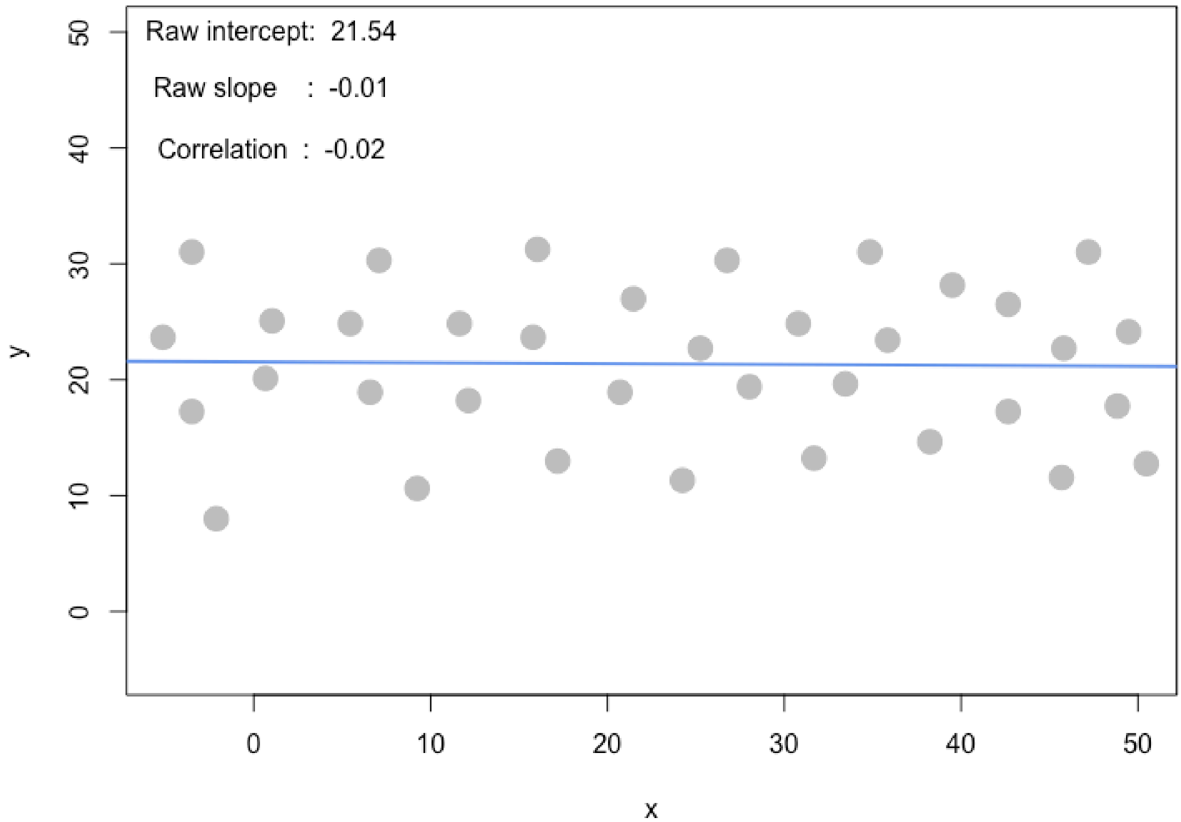
**a. Create a horizontal set of random points, with a relatively narrow but flat distribution.**

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

We expect the slope close be to 0.

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

We expect the correlation be close to 0.



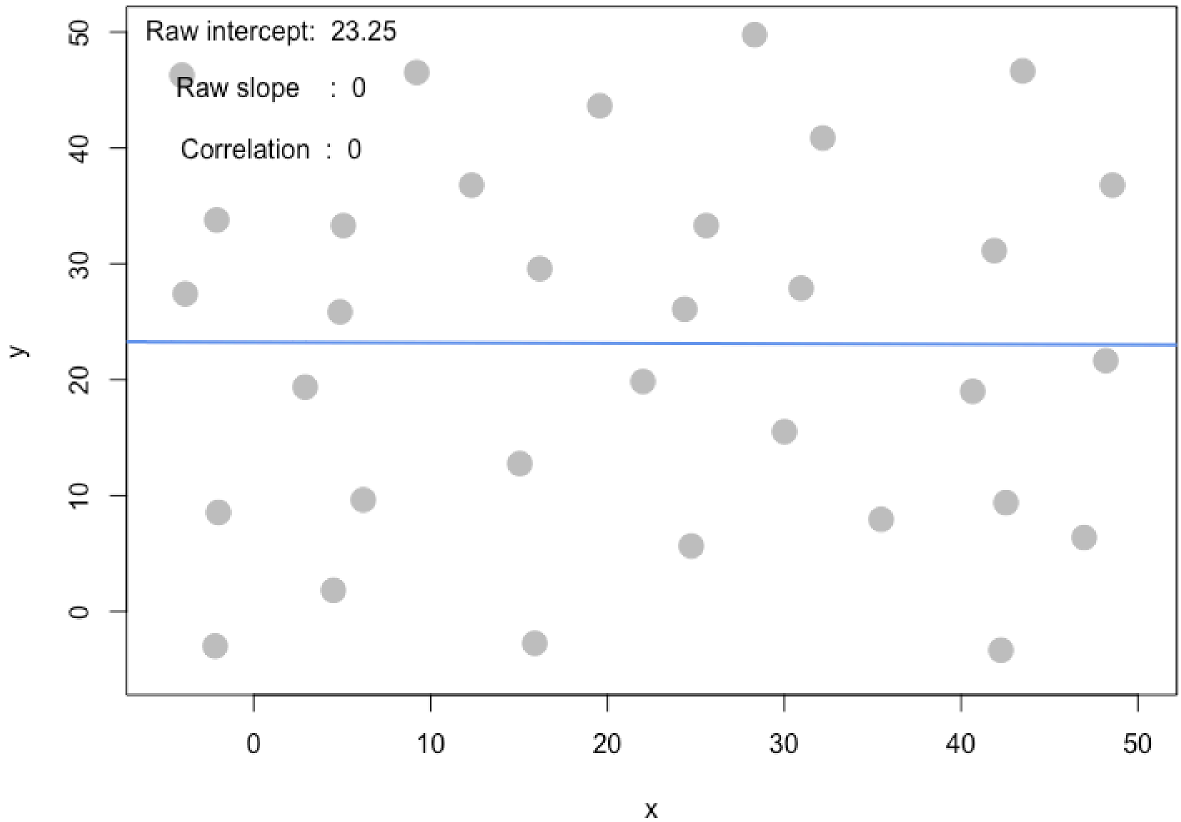

**b. Create a completely random set of points to fill the entire plotting area, along both x-axis and y-axis**

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

We expect the slope close be to 0.

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

We expect the correlation be close to 0.




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

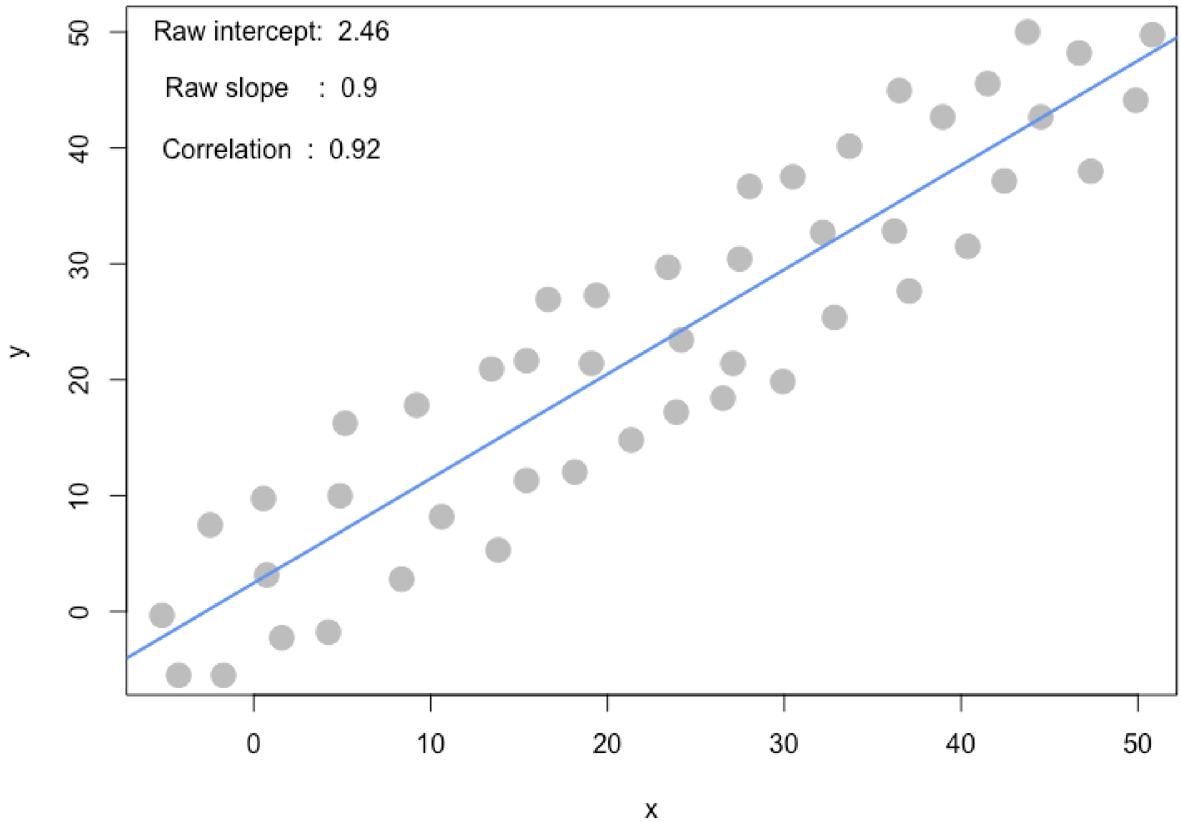
**i) What raw slope of x and y would you generally expect? (note that x, y have the same scale)**

We expect the slope close be to 1.

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

If x and y are linear, we can expect the correlation be close to 1.

If x and y are nonlinear, we can expect the correlation be close to 0.




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

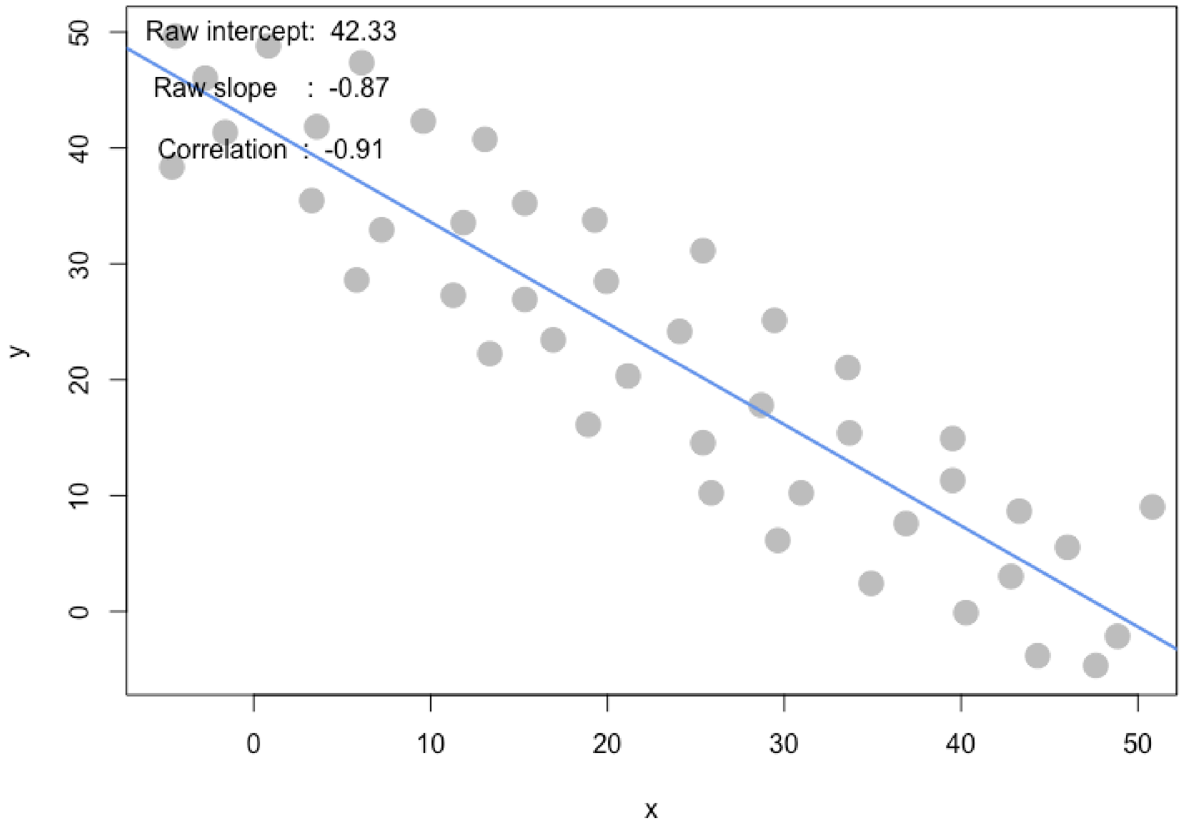
**i) What raw slope of x and y would you generally expect? (note that x, y have the same scale)**

We expect the slope close be to -1.

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

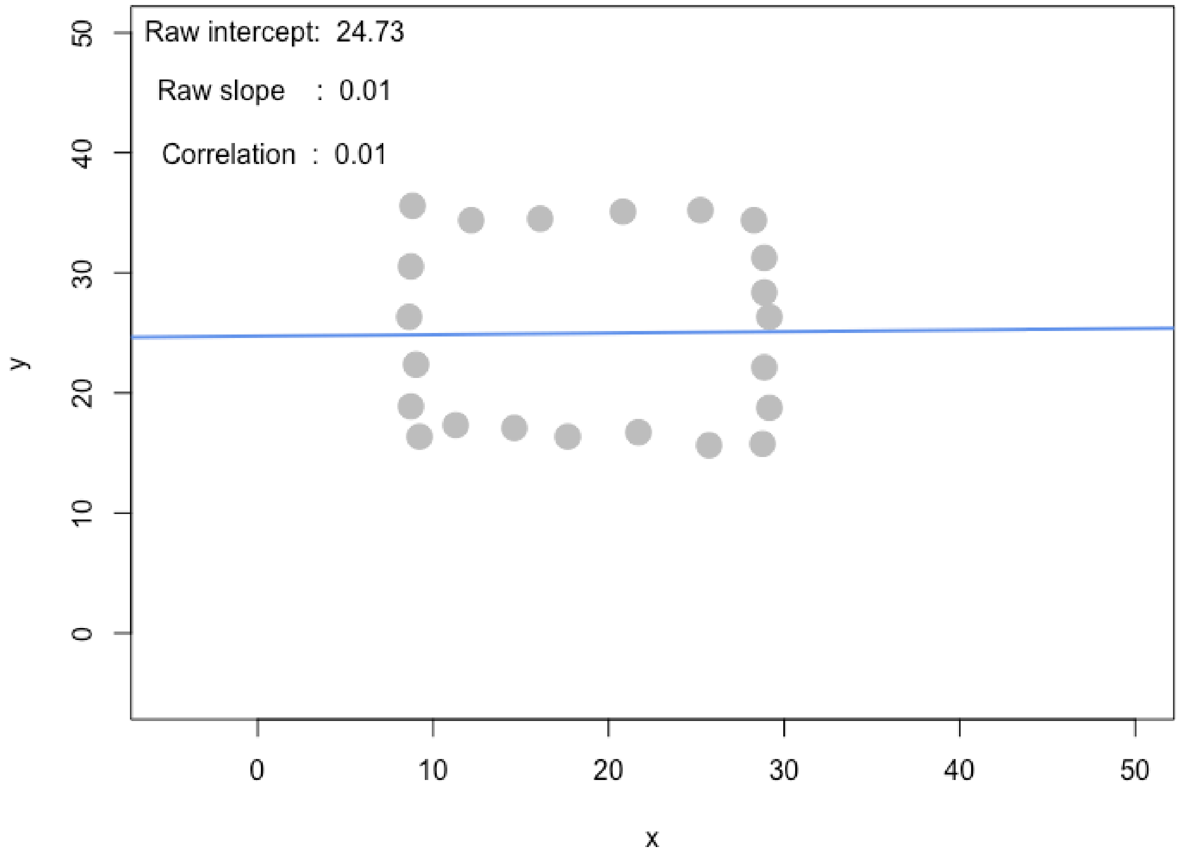
If x and y are linear, we can expect the correlation be close to -1.

If x and y are nonlinear, we can expect the correlation be close to 0.



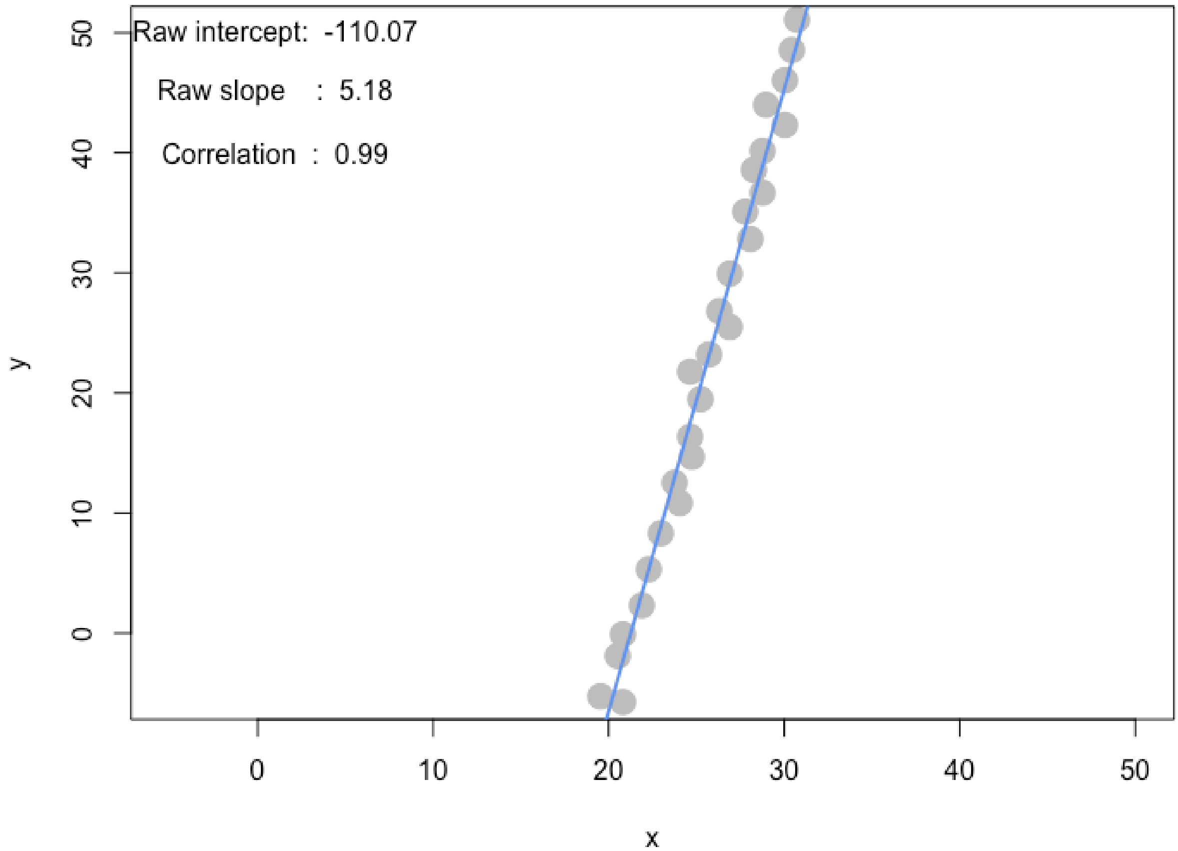

**e. Apart from any of the above scenarios, find another pattern of data points with no correlation (r   0).**

We create a symmetric pattern.



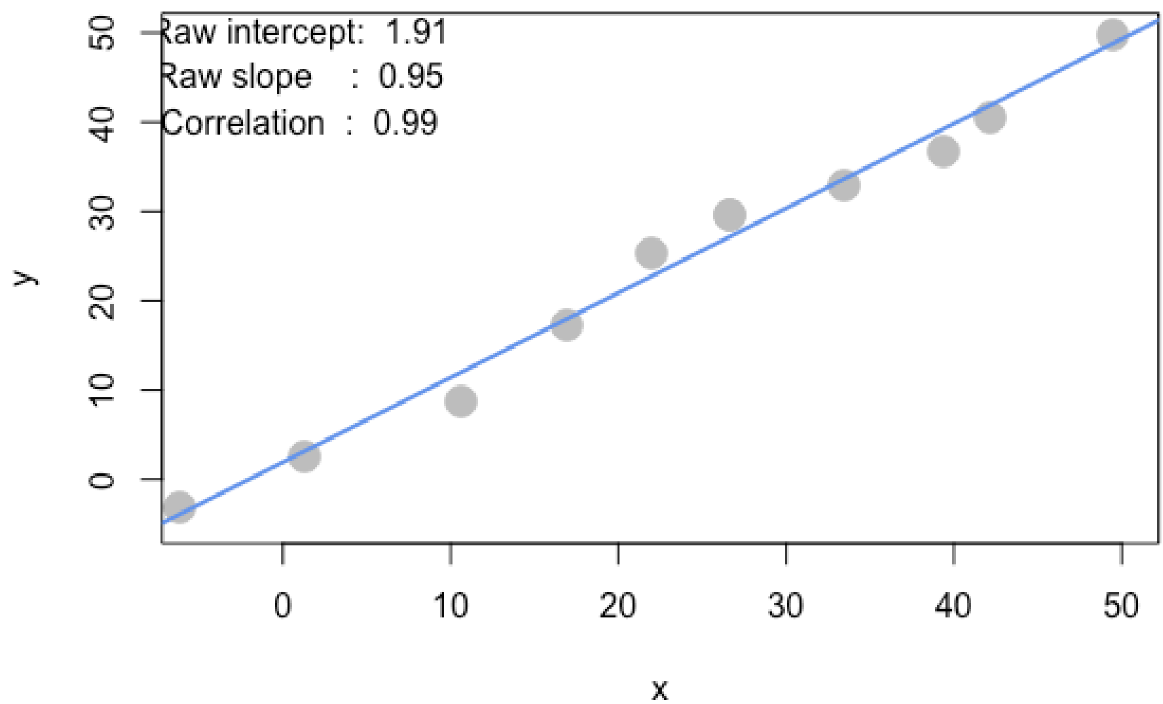

**f. Apart from any of the above scenarios, find another pattern of data points with perfect correlation (r 1).**

We create a set of points highly centralized into a positive steep slope line (almost vertical line).




**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:**




```
##             x         y
## 1  -6.155682 -3.147627
## 2   1.285714  2.546573
## 3  10.618991  8.715290
## 4  16.925259 17.256591
## 5  26.636912 29.594025
## 6  26.636912 29.594025
## 7  33.447681 32.915641
## 8  42.150331 40.507909
## 9  49.465602 49.760984
## 10 21.970273 25.323374
## 11 39.375573 36.711775
```

**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))
```

```
##
## Call:
## lm(formula = PTS$y ~ PTS$x)
##
## Residuals:
##    Min      1Q Median     3Q    Max
## -3.266 -1.037 -0.586  1.689  2.581
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.91370    1.12288   1.704    0.123
## PTS$x        0.94805    0.03876  24.461 1.53e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.114 on 9 degrees of freedom
## Multiple R-squared:  0.9852, Adjusted R-squared:  0.9835
## F-statistic: 598.3 on 1 and 9 DF,  p-value: 1.528e-09
```

The intercept is the same as the plot(1.91).

**iii) Estimate the correlation of x and y to see it is the same as reported in the plot:**

```r
cor(PTS)
```

```
##           x         y
## x 1.0000000 0.9925631
## y 0.9925631 1.0000000
```

The correlation of x and y is the same as reported in the plot($r = 0.99$).

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

```r
X <- (PTS$x-mean(PTS$x))/sd(PTS$x)
Y <- (PTS$y-mean(PTS$y))/sd(PTS$y)
summary(lm(Y ~ X))
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19820 -0.06295 -0.03557  0.10253  0.15663
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.595e-17  3.869e-02    0.00        1
## X           9.926e-01  4.058e-02   24.46 1.53e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1283 on 9 degrees of freedom
## Multiple R-squared:  0.9852, Adjusted R-squared:  0.9835
## F-statistic: 598.3 on 1 and 9 DF,  p-value: 1.528e-09
```

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

```r
cor(X,Y)
```

```
## [1] 0.9925631
```

The covariance of standardized simple-regression is equal to correlation (0.99).