HW7

113078506

2025-04-01

- Gen AI Usage: I use Gen AI to refine my grammar and verify my reasoning.
- Students who helped: 113078505, 113078502, and 113078514, helped with conclusion reasoning.

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

```
library(data.table)
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 the App

- Done

ii. Choose a bundle and use intuition to recommend 5 other bundels in dataset.

- I choose "WinterWonderland", and I would recommend "snowflakes", "betweenspring", "retrosummer", "simplyautumn", and "fallinlovewiththefall". I think these bundles are all describing "seasons", which might all contain similar element.

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

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

```
library("lsa")
```

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

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

```r
cosine_ac_bundle <- cosine(ac_bundles_matrix)

top5rec <- apply(cosine_ac_bundle, 1, function(x) { # 1 = apply by row
    x_sorted <- sort(x, decreasing = TRUE)
    names(x_sorted)[2:6]  # skip the first one (self-similarity)
  })

# Convert to a data.frame for easy viewing
top5_df <- as.data.frame(t(top5rec))
colnames(top5_df) <- paste0("Rec_", 1:5)
head(top5_df)
```

```
##                            Rec_1            Rec_2      Rec_3          Rec_4
## 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
##                               Rec_5
## Maroon5V                       word
## between         AccessoriesStickerPack
## pellington           tropicalparadise
## StickerLite                    Random
## saintvalentine         lovestinks2016
## HipsterChicSara           StickerLite
```

```r
get_top_5_rec <- function(input_matrix) {
  cosine_matrix <- cosine(input_matrix)

  top5 <- apply(cosine_matrix, 1, function(x) { # 1 = apply by row
    x_sorted <- sort(x, decreasing = TRUE)
    names(x_sorted)[2:6]  # skip the first one (self-similarity)
  })

  # Convert to a data.frame for easy viewing
  top5_dataframe <- as.data.frame(t(top5))
  colnames(top5_dataframe) <- paste0("Rec_", 1:5)
  return(top5_dataframe)
}

top5_recommendations <- get_top_5_rec(ac_bundles_matrix)
top5_recommendations["WinterWonderland", ]
```

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.

```
##                          Rec_1       Rec_2       Rec_3     Rec_4      Rec_5
## WinterWonderland PhotoboothFest DecktheHall StickerLite japan2015 CampusLife
```

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

- The top 5 recommendations for "WinterWonderland" are "PhotoboothFest", "DecktheHall", "StickerLite", "japan2015", and "CampusLife", which is completely different from my intuitive. :D

i. Let's create correlation based recommendations.

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

```
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_top5_rec <- get_top_5_rec(ac_bundles_mc_b)
```

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

```
cor_top5_rec["WinterWonderland", ]
```

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

```
##                            Rec_1       Rec_2     Rec_3       Rec_4      Rec_5
## WinterWonderland PhotoboothFest DecktheHall japan2015 StickerLite CampusLife
```

- The top 5 recommendations for "WinterWonderland" using correlation are "PhotoboothFest", "DecktheHall", "japan2015", "StickerLite", and "CampusLife". The ranking of "japan2015" and "StickerLite" are switched.

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

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

```
account_means <- apply(ac_bundles_matrix, 1, mean)
account_means_matrix <- matrix(account_means, nrow = length(account_means),  ncol = ncol(ac_bundles_mat
ac_bundles_mc_b_r <- ac_bundles_matrix - account_means_matrix

adj_cosine_top5_rec <- get_top_5_rec(ac_bundles_mc_b_r)
```

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

```
adj_cosine_top5_rec["WinterWonderland", ]
```

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

```
##                            Rec_1       Rec_2       Rec_3   Rec_4       Rec_5
## WinterWonderland PhotoboothFest DecktheHall StickerLite Mom2013 CatpixCubie
```

- The top 5 recommendations for "WinterWonderland" using correlation are "PhotoboothFest", "DecktheHall", "StickerLite", " Mom2013" and "CatpixCubie". The last 2 recommendations are completely changed.

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?

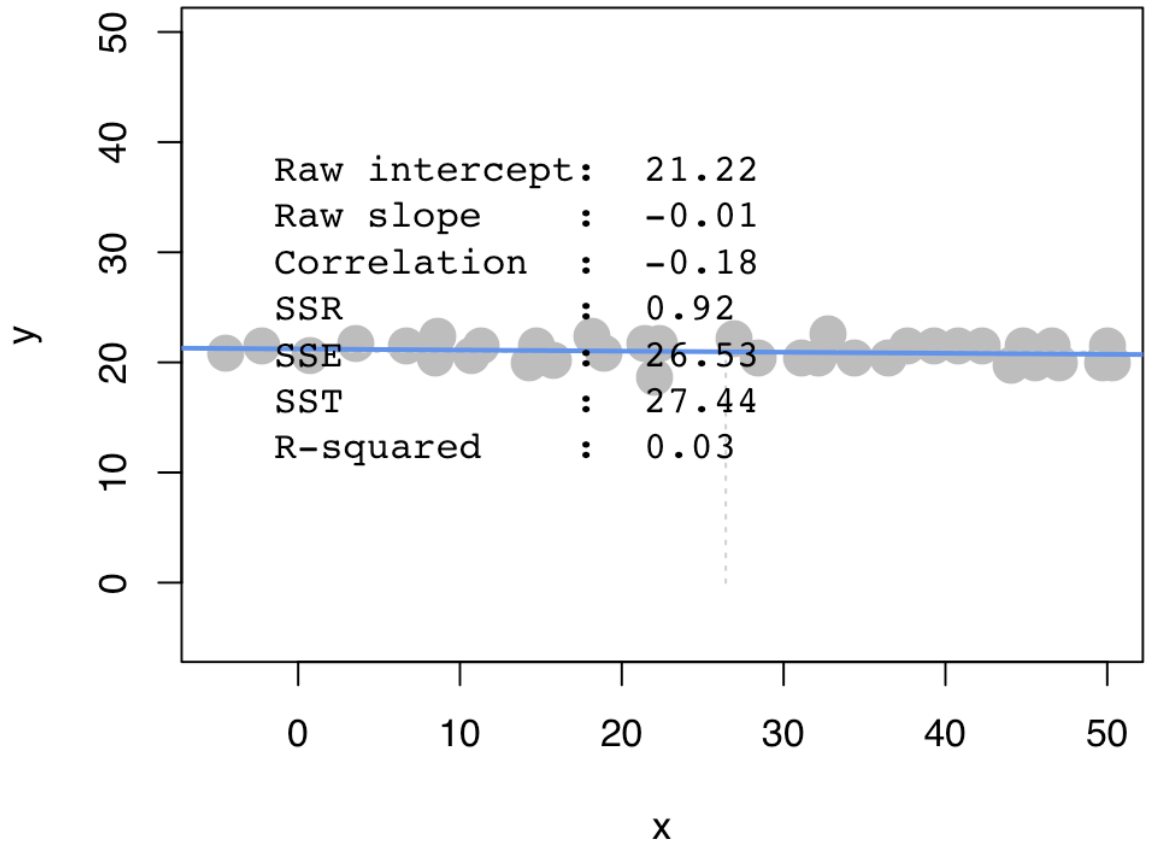
- No, the three sets of geometric recommendations are quite different from the ones I picked using intuition. My intuitive picks, like "snowflakes", "betweenspring", "retrosummer", "simplyautumn", and "fallinlovewiththefall", were based on seasonal or thematic keywords, assuming that bundles with similar visual or semantic content would be used in similar ways. However, the cosine-based recommendations like "PhotoboothFest", "DecktheHall", "StickerLite", and "CampusLife" seem to reflect actual usage patterns from users, which are the potentially bundles that are often purchased or used together by similar users, even if they have no obvious thematic overlap. I think this similarity could also be affected by popularity bias, where commonly used bundles show up as similar just because they appear in many accounts. In conclusion, my intuitive picks are based on visual/surface similarity, while the model reflects underlying usage correlations.

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

- The conceptual differences between cosine similarity, correlation, and adjusted-cosine lie in how they handle the scale and centering of data. Cosine similarity measures the angle between two vectors without centering, focusing on the direction of usage patterns rather than the amount. It's useful when we care about whether two bundles are used in similar ways, regardless of how often. Correlation, on the other hand, compares how two bundles vary relative to their own means, so it accounts for popularity differences and highlights co-variation in usage. Adjusted-cosine goes a step further by centering usage data by users (rows) instead of bundles (columns), helping to control for users who tend to use more (or fewer) bundles overall. This adjustment provides a more personalized measure of similarity by focusing on the relative preferences of individual users.

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

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



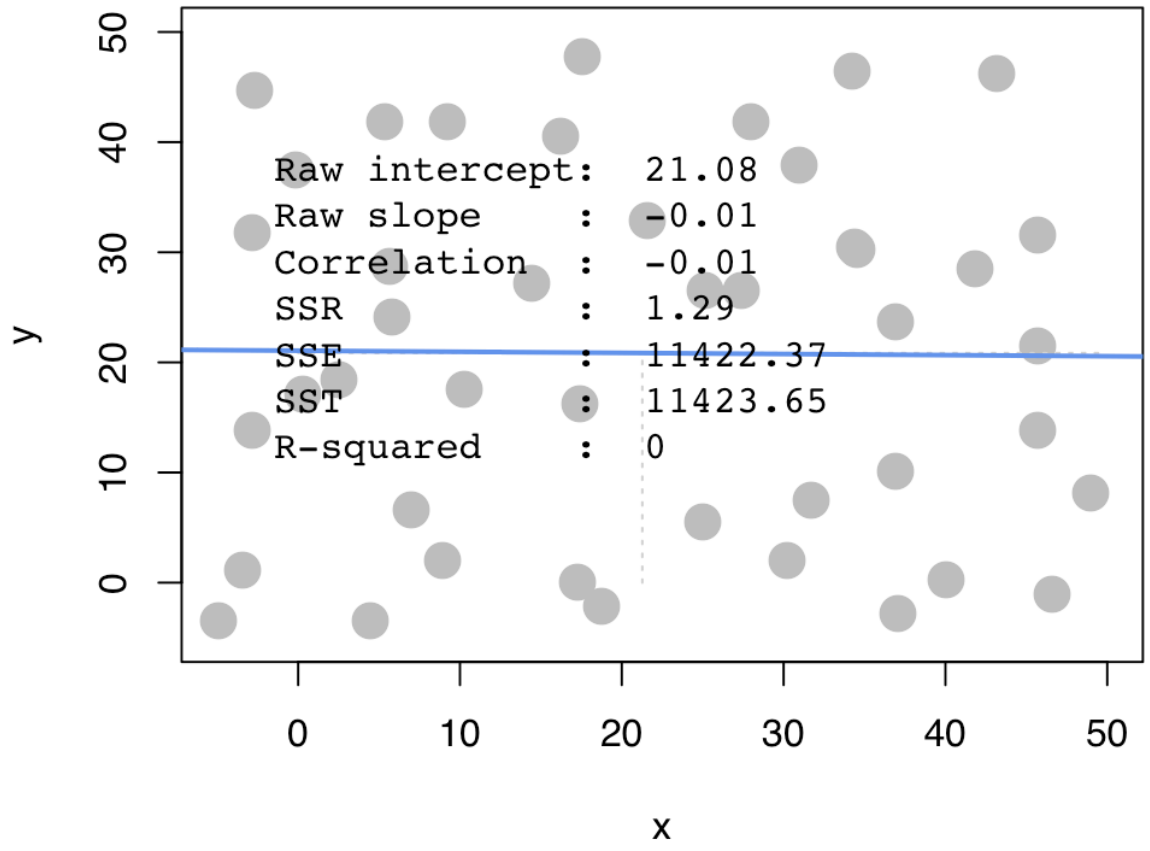
i. What raw slope of x and y would you generally expect?

- I would expect the raw slope be close to 0. In my simulation it's -0.01, which is exactly close to 0.

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

- I would expect the correlation of x and y be close to 0. In my simulation it's -0.18, which is not quite close to 0.

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



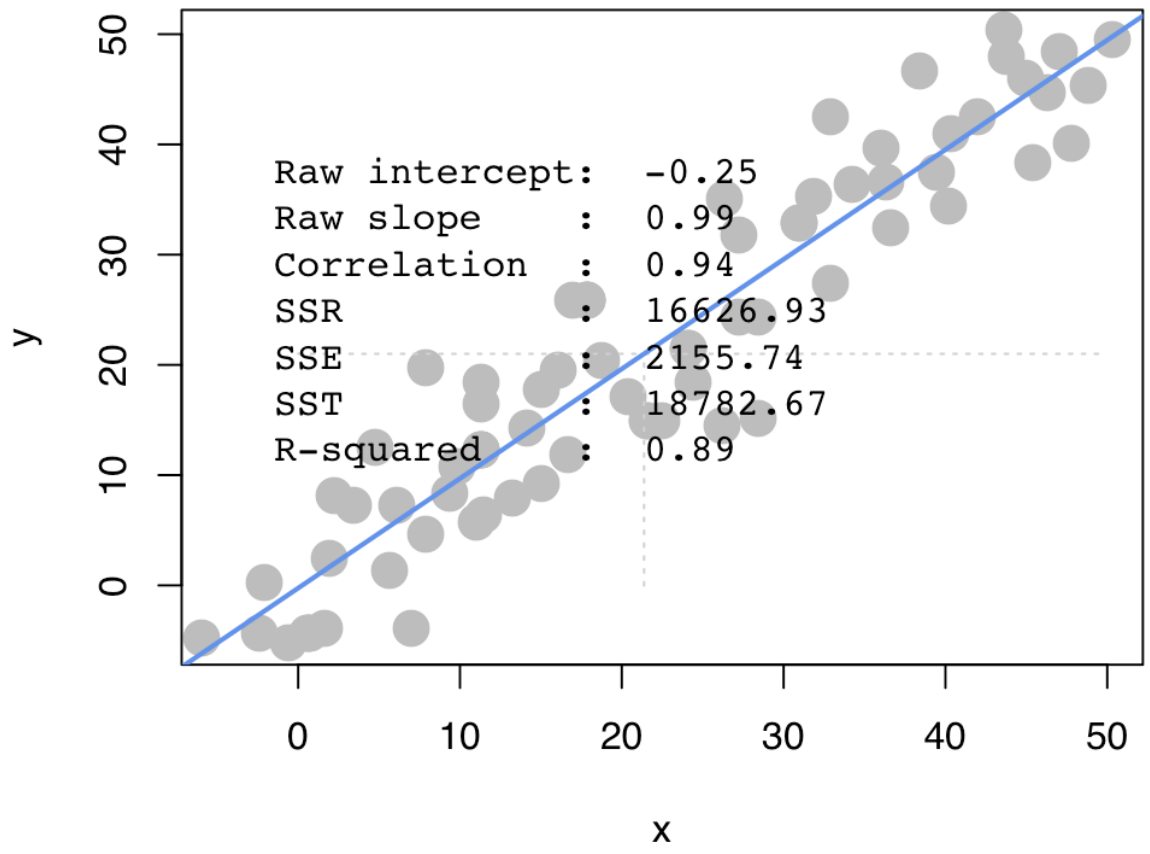
i. What raw slope of the x and y would you generally expect?

- The raw slope would generally be close to zero, because the points are scattered randomly without any linear trend, which means increasing in x don't predict increases or decreases in y. In my simulation, the raw slope is -0.01, which is exactly close to 0.

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

- The correlation would also generally be close to zero, as there is no linear association between x and y. In my simulation, the correlation is -0.01, which is also exactly close to 0.

c. Scenario C: Create a diagonal set of random points trending upwards at 45 degrees

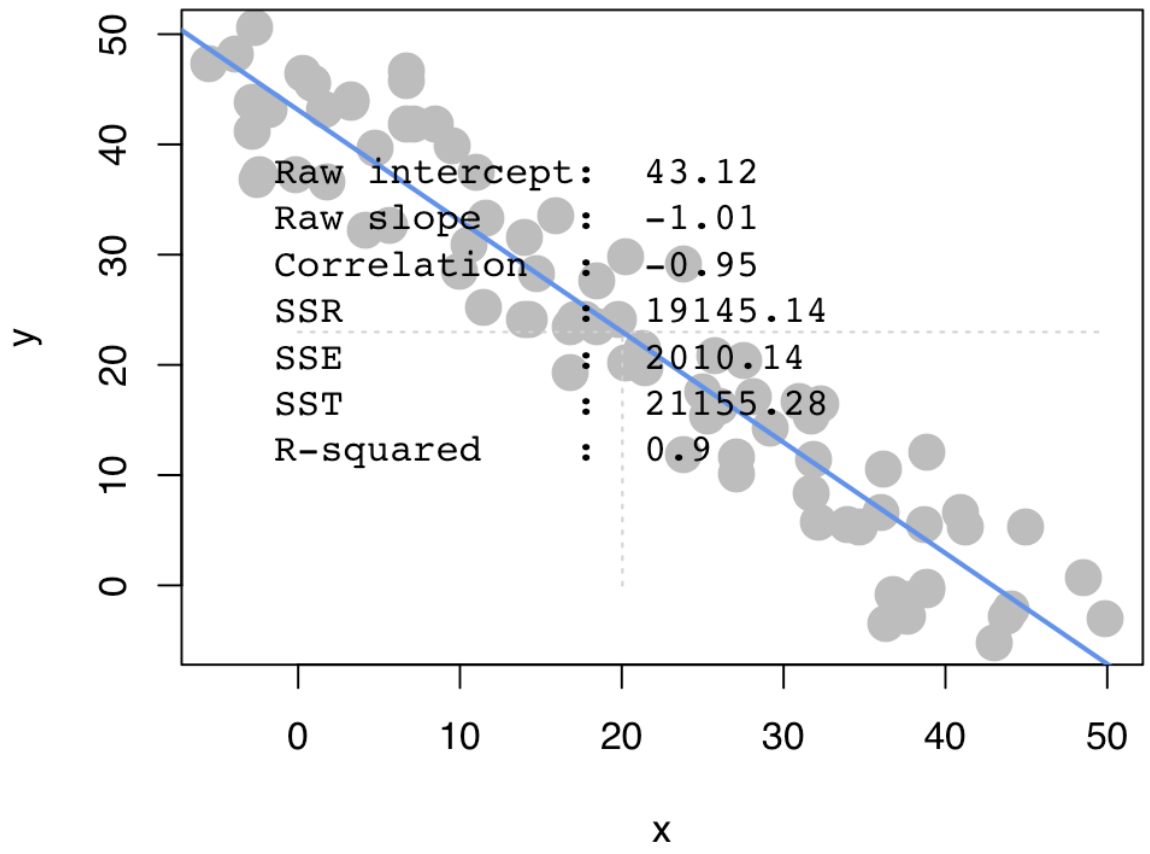


i. What raw slope of the x and y would you generally expect? (note that x, y have the same scale) -The raw slope would generally be around 1, because for every unit increase in x, y increases by about the same amount. In my simulation, the raw slope is 0.99, which is close to 1.

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

- The correlation would generally be close to 1, indicating a strong positive linear relationship. In my simulation, the correlation is 0.94, which is close to 1.

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



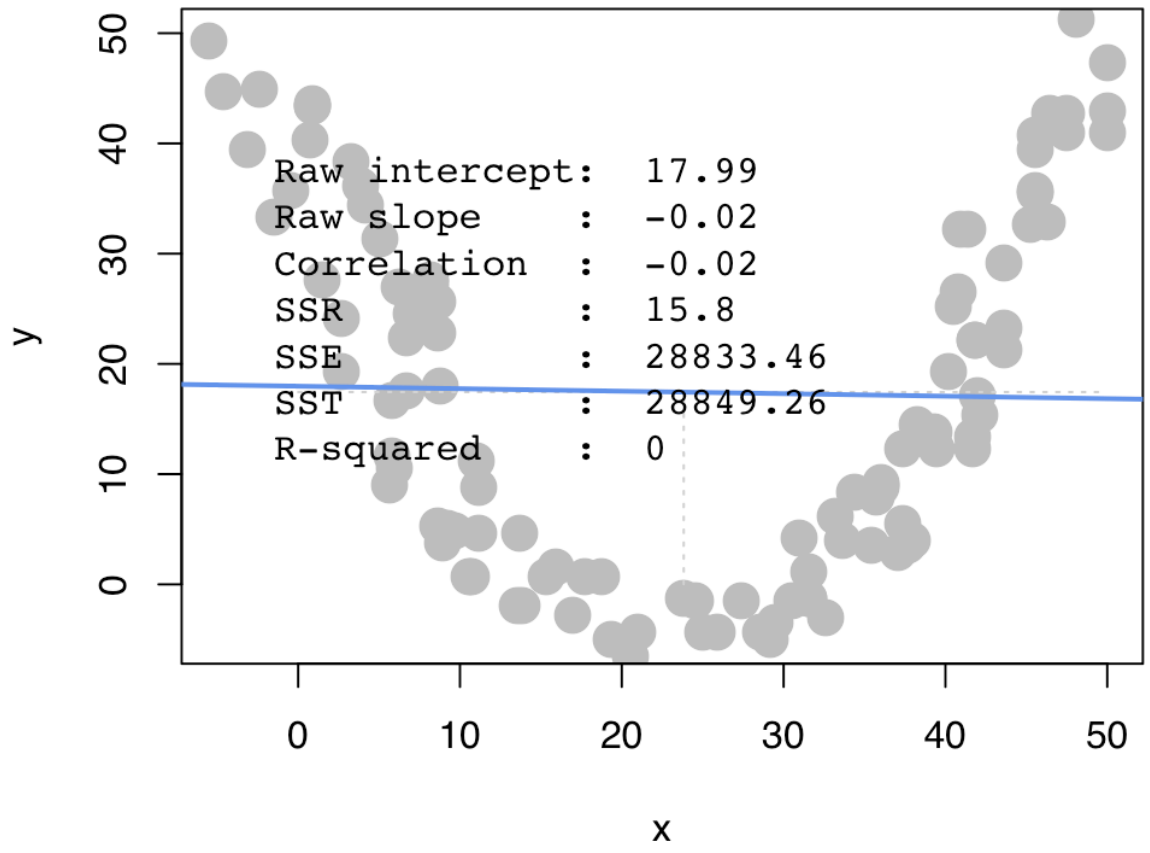
i. What raw slope of the x and y would you generally expect? (note that x, y have the same scale)

- The raw slope would generally be around –1, since y decreases by roughly the same amount that x increases. In my simulation, the raw slope is -1.01, which is close to -1.

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

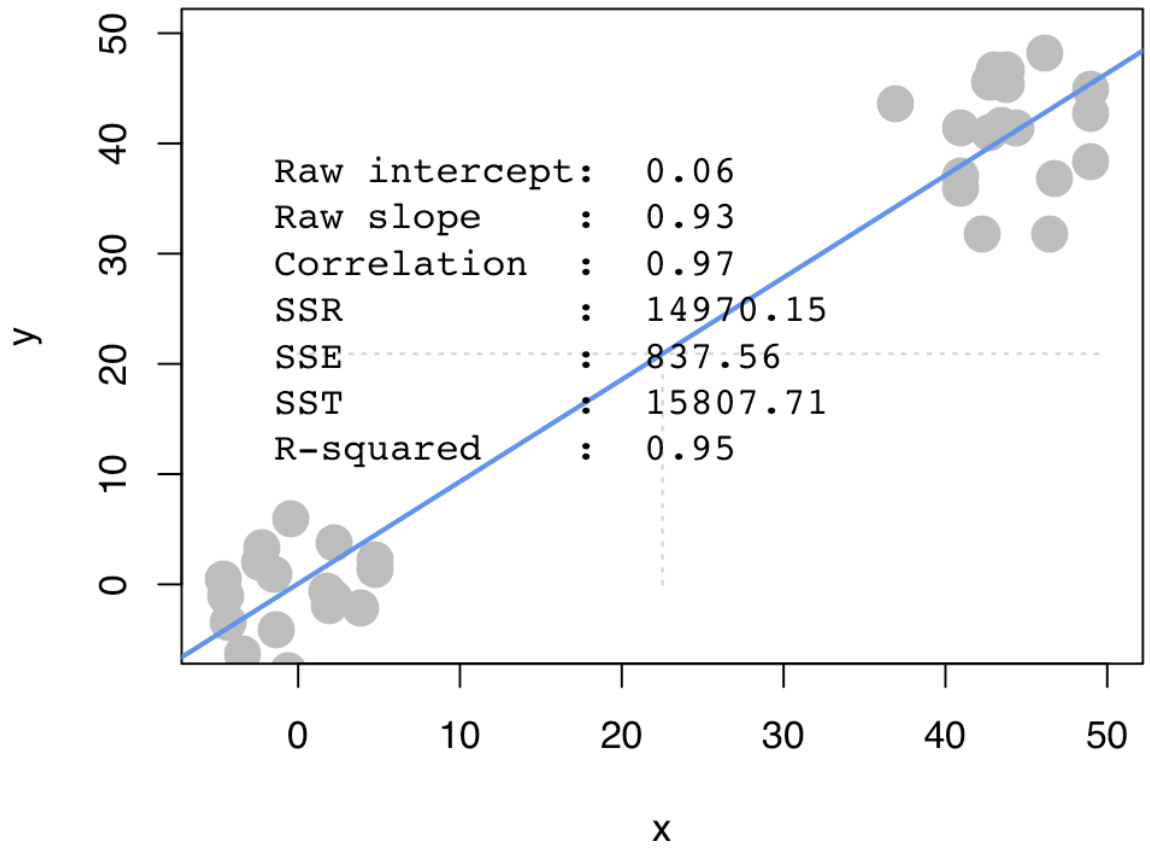
- The correlation would generally be close to –1, indicating a strong negative linear relationship. In my simulation, the correlation is -0.95, which is also close to -1.

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



- One example is a U-shaped pattern (e.g., $y = x^2$). Visually, it shows a strong relationship, y clearly depends on x, but since the trend is nonlinear, the linear correlation is near zero. That's because correlation only measures linear association, and the positive and negative trends cancel each other out.

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



- An example is a plot where two separate groups of data have no internal trend, but when combined, they form an overall diagonal pattern. For instance, in the "Two groups" plot, each group individually shows little to no correlation, yet when viewed together, the combined data forms a strong negative slope. As a result, the computed correlation might be high in magnitude, but it misrepresents the relationship within each group. This is an example of how correlation can be misleading when data contains distinct subgroups or clusters.

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

```
library("compstatslib")
#pts <- interactive_regression()
```

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

- I can't save pts when knitting the file, so I wrote it in a csv file and read it for further calculation.

```
pts <- read.csv("pts.csv")
summary(lm(pts$y ~ pts$x))
```

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 ptsx))

```
##
## Call:
## lm(formula = pts$y ~ pts$x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.7767  -5.8381  -0.5981   4.3781  18.1837
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.00172    1.79807  18.354  < 2e-16 ***
## pts$x       -0.26483    0.06938  -3.817 0.000387 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.661 on 48 degrees of freedom
## Multiple R-squared:  0.2329, Adjusted R-squared:  0.2169
## F-statistic: 14.57 on 1 and 48 DF,  p-value: 0.0003865
```

```
cor(pts$x, pts$y)
```

iii. Estimate the correlation of x and y to see it is the same as reported in the plot: cor(pts)

```
## [1] -0.4825721
```

```
x_std <- scale(pts$x)
y_std <- scale(pts$y)
summary(lm(y_std ~ x_std))
```

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

```
##
## Call:
## lm(formula = y_std ~ x_std)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.81634 -0.59651 -0.06111  0.44734  1.85793
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.362e-15  1.251e-01   0.000 1.000000
## x_std       -4.826e-01  1.264e-01  -3.817 0.000387 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8849 on 48 degrees of freedom
## Multiple R-squared:  0.2329, Adjusted R-squared:  0.2169
## F-statistic: 14.57 on 1 and 48 DF,  p-value: 0.0003865
```

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

- The slope of the regression line using standardized variables (-0.4825721) is exactly equal to the correlation coefficient between x and y (-0.4826). This is because when variables are standardized (mean = 0, SD = 1), the regression slope becomes a measure of how many standard deviations y changes per standard deviation of x — which is the definition of correlation.