BACS HW9

Student ID: 110065508

Due: 2022/04/17

Question 1. (a) Each bundles have 6 recommendations.

I choose the bundle "Between Spring". Since it is a bundle related to seasons and flower, I guess the top five recommendations are:

1. The bouqs.
2. Spring rose.
3. Hello spring.
4. Autumn.
5. 2014 summer.

(b) First, I use `read_csv` to read the file since I can't install the package `data.table` due to a `R` version issue.

```r
library(tidyverse)
library(lsa) # cosine()

# Question 1
ac_bundles_dt <- read_csv('piccollage_accounts_bundles.csv')
ac_bundles_matrix <- as.matrix(ac_bundles_dt[, -1, with=FALSE])
rm(ac_bundles_dt)
```

(i) The cosine recommendation matrix can be computed by the following code:

```r
# Question 1 (b-i)
top_5_recommend_cos <- function(ac_bundles_matrix){
  cos_matrix <- cosine(ac_bundles_matrix) # Obtain cosine similarity
  sorted_names_matrix <- c() # construct a empty matrix
```

```

  for (i in colnames(cos_matrix)){ # extract every column names
    temp_vector <- cos_matrix[,i] # extract a column of cos matrix
    # sort the similarities decreasingly
    temp_vector_sorted <- data.frame(sort(temp_vector, decreasing=TRUE))
    # the rownames are sorted according to the cosine similarity, too
    names_vector <- rownames(temp_vector_sorted)
    # combine the result to get a full recommendation matrix
    sorted_names_matrix <- cbind(sorted_names_matrix, names_vector)
  }

  # assign the column names to the sorted names matrix
  colnames(sorted_names_matrix) <- colnames(cos_matrix)
  # We only want top 5 (omit each bundle itself)
  recommand_matrix <- sorted_names_matrix[2:6,]

  return(recommand_matrix)
}

recommand_matrix_cos <- top_5_recommend_cos(ac_bundles_matrix)
```

Use the command

`> recommand_matrix_cos[,"betweenspring"]`

The console returns the following bundles:

`"OddAnatomy" "supersassy" ‘word" "KLL" "xoxo"`

(ii) The correlation recommendation matrix can be computed by the following code:

```
# Question 1 (b-ii)
mean_centering_col <- function(ac_bundles) {
  bundle_means <- apply(ac_bundles, 2, mean)
  bundle_means_matrix <- t(replicate(nrow(ac_bundles), bundle_means))
  # Subtract each row with its mean
  ac_bundles_mc_b <- ac_bundles - bundle_means_matrix

  return(ac_bundles_mc_b)
}

ac_bundles_matrix_centered <- mean_centering_col(ac_bundles_matrix)
recommand_matrix_cor <- top_5_recommend_cos(ac_bundles_matrix_centered)
rm(ac_bundles_matrix_centered)
```

Use the command

`> recommand_matrix_cor[,"betweenspring"]`

The console returns the following bundles:

`"OddAnatomy" "supersassy" "word" "xoxo" "KLL"`

(iii) The adjusted-cosine based recommendation matrix can be computed by the following code:

```
# Question 1 (b-iii)
mean_centering_row <- function(ac_bundles) {
  bundle_means <- apply(ac_bundles, 1, mean)
  bundle_means_matrix <- t(replicate(ncol(ac_bundles), bundle_means))
  # Subtract each row with its mean
  ac_bundles_mc_b <- ac_bundles - t(bundle_means_matrix)

```

```
  return(ac_bundles_mc_b)
}

# for adjust cosine
ac_bundles_matrix_ad <- mean_centering_row(ac_bundles_matrix)
recommand_matrix_ad <- top_5_recommend_cos(ac_bundles_matrix_ad)
rm(ac_bundles_matrix_ad)
```

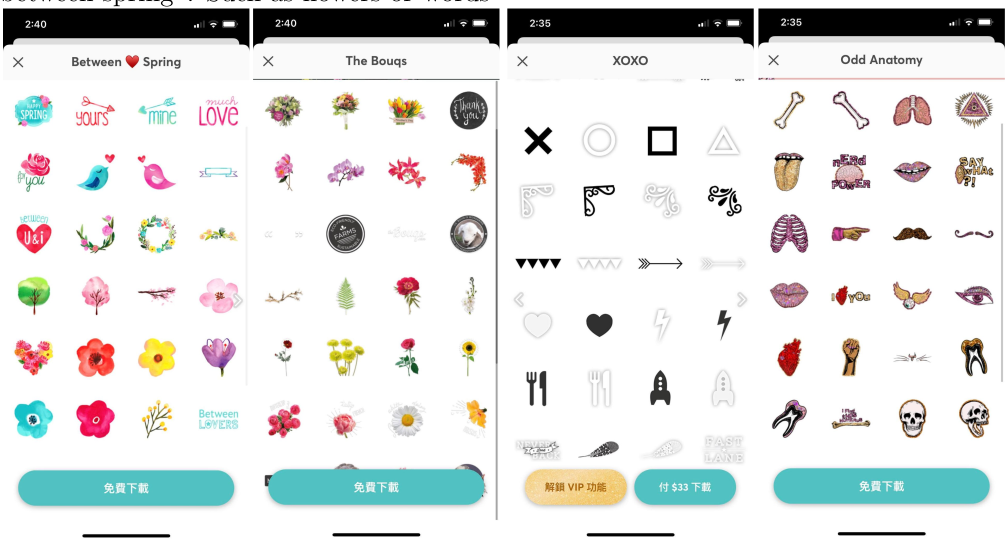
Use the command

`> recommand_matrix_ad[,"betweenspring"]`

The console returns the following bundles:

`"OddAnatomy" "thebouqs" "xoxo" "word" "between"`

(c) However, I have found some of the recommended bundles which is in the dataset. They share some features with the bundle "between spring". Such as flowers or words



(d) Basically, I think cosine similarity, correlation, and adjusted-cosine are the same things. We try to define similarities in a inner product space ($\mathbb{R}^n$) by generalized the cos function. The main difference is that we view data as vectors centralized at different points: cosine similarity is centered at 0, adjusted-cosine is centered at the mean of all data, and correlation is centered at 0, but we subtract each vector with the mean of its components. ∎

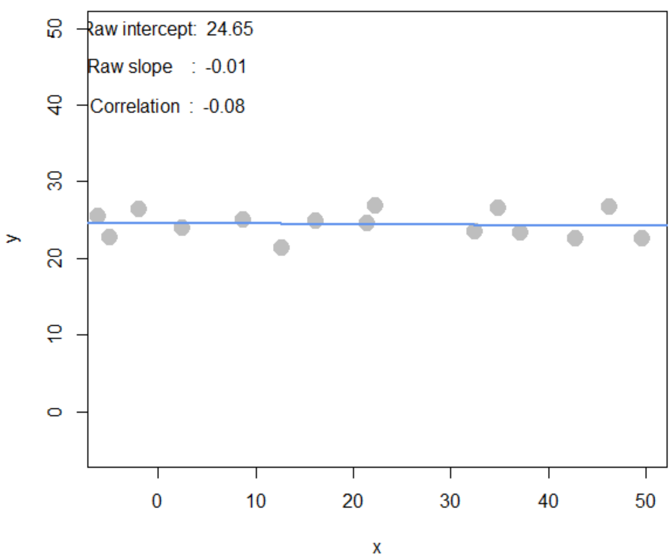
Question 2. Use the command

`source("demo_simple_regression.R")`

`interactive_regression()`

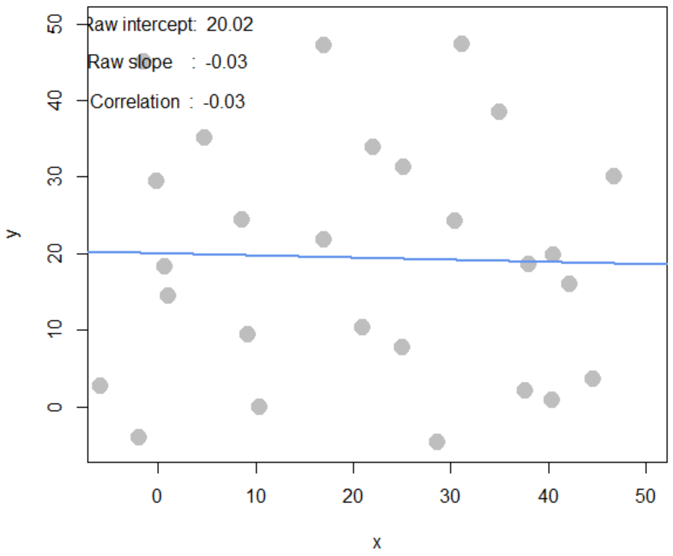
to conduct simulations.

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



- The slope of x and y I expect $m = 0$
- The correlation of x and y that I expect $r(x, y) = 0$

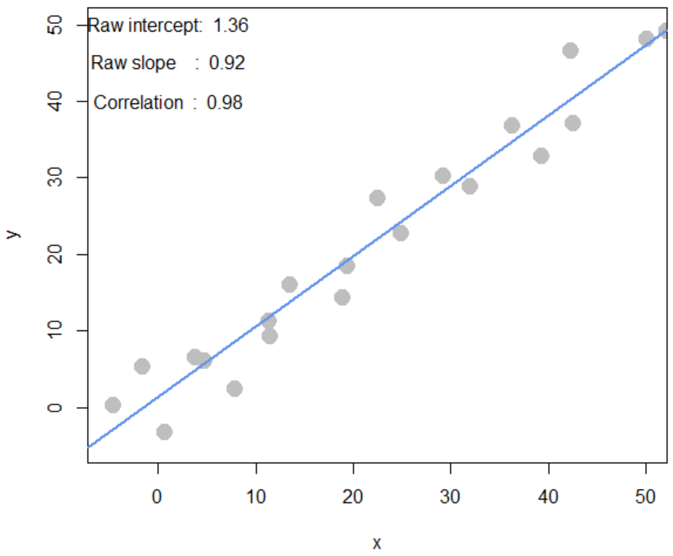
(b) Create a completely random set of points to fill the entire plotting area, along both x-axis and y-axis



- The slope of x and y I expect $m = 0$
- The correlation of x and y that I expect $r(x, y) = 0$

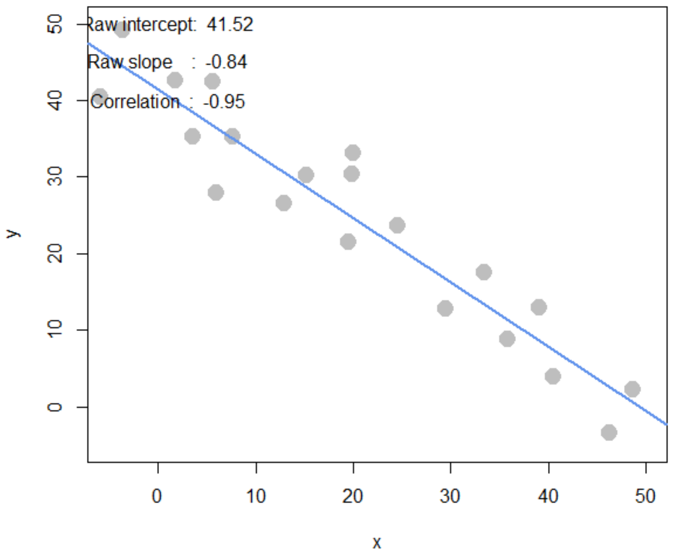
(c) Create a diagonal set of random points trending upwards at 45 degrees

- The slope of x and y I expect $m = 1$
- The correlation of x and y that I expect $r(x, y) = 1$

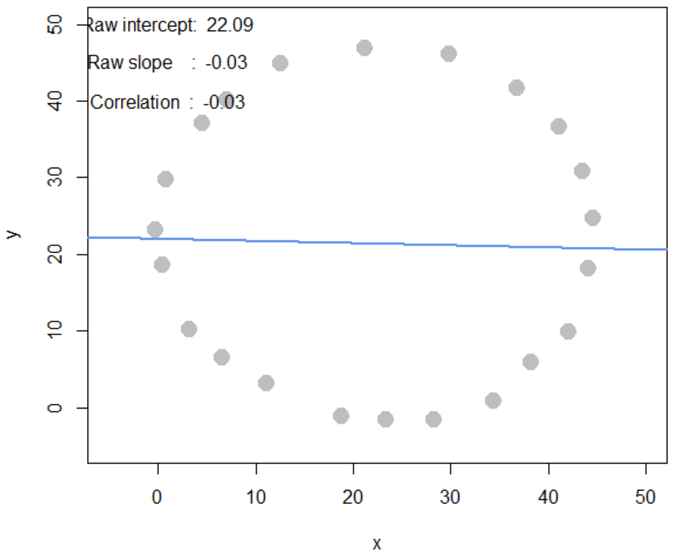


(d) Create a diagonal set of random trending downwards at 45 degrees

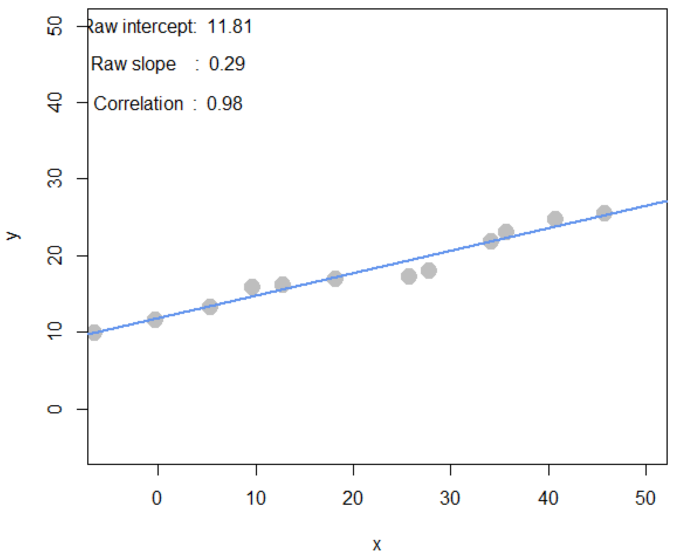
- The slope of x and y I expect $m = -1$
- The correlation of x and y that I expect $r(x, y) = 1$



(e) I found that when all data points are on a circle, the correlation would be 0, too.



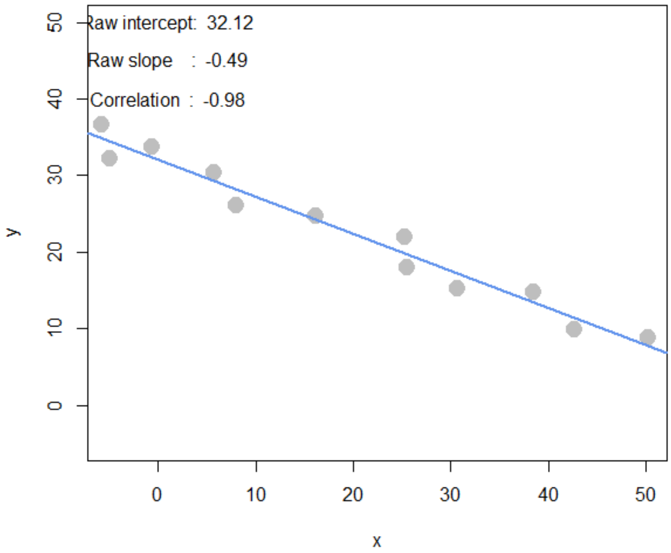
(f) I found that when all data points are on a straight line whose slope is nonzero, then the correlation would be 1.



(g) The code of this problem:

```
# Question 2 (g)
source("demo_simple_regression.R")
pts <- interactive_regression() # run the simulation and record the points
slope <- summary(lm(pts$y~pts$x)) # estimate the regression intercept and slope
cor_pts <- cor(pts) # estimate the correlation
pts_std <- scale(pts) # standardize
slope_std <- summary(lm(pts_std[,"y"]~pts_std[,"x"])) # regression slope
cor_pts_std <- cor(pts_std) # correlation
```

(i) The points generated:



(ii) The regression intercept is $k = 32.12312$ and slope $m = -0.48548$.

(iii) $r = -0.9834927$.

(iv) The regression intercept of the standardized values is $k = 0$ and slope $m = -0.9835$.

(v) It suggests that the correlation is the slope of the regression model of standardized values.

■