HW 9

110077443

4/14/2022

Please note that all code in this document is presented in a grey box and the output reflected below each box

- The below code allows lengthy lines of comments to display neatly within the grey box (wrapping it)

```
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)
```

1) Automated recommendation system for the PicCollage mobile app

```
# Import files
library(data.table)
ac_bundles_dt <- fread("piccollage_accounts_bundles.csv")
ac_bundles_matrix <- as.matrix(ac_bundles_dt[, -1, with = FALSE])
dim(ac_bundles_matrix)  # Checking number of rows and columns in matrix
```

```
ANSWER ## [1] 24649   165
```

a) Exploring sticker bundles

i) Recommendations from PicCollage App

- ANSWER ## The application does not provide me with any recommendations (Android App)

ii) Single sticker bundle both in our data set and in the Sticker Store

- ANSWER ## summerlovin

Recommend 5 other bundles by intuition:

- ANSWER ## cutoutlov / justmytype / snowflakes / HeartStickerPack / saintvalentine

b) Similar bundles using geometric models of similarity:

i) Creating cosine similarity based recommendations

1. Creating a matrix of the top 5 recommendations for all bundles

```
require(lsa)  # Package required to use cosine function
sim_matrix <- cosine(ac_bundles_matrix)  # Cosine similarity matrix
dim(sim_matrix)  # Checking dimensions
```

```
ANSWER ## [1] 165 165
```

```
rec_mat <- t(apply(sim_matrix, 1, function(x) names(sort(x, decreasing = T)[2:6])))
rownames(rec_mat) <- row.names(sim_matrix)
colnames(rec_mat) <- c("1st", "2nd", "3rd", "4th", "5th")
knitr::kable(head(rec_mat, 5), caption = "Cosine Based Recommendations (Sample of 5)",
    align = "c")
```

Table 1: Cosine Based Recommendations (Sample of 5)

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

2. Creating a new function that automates the above functionality

```
rec_mat_fun <- function(ac_bundles_matrix) {
    library(lsa)
    sim_matrix <- cosine(ac_bundles_matrix)
    rec_mat <- t(apply(sim_matrix, 1, function(x) names(sort(x,
        decreasing = T)[2:6])))
    rownames(rec_mat) <- row.names(sim_matrix)
    colnames(rec_mat) <- c("1st", "2nd", "3rd", "4th", "5th")
    return(rec_mat)
}
require(dplyr)  # For piping function (%>%)
rec_mat_fun(ac_bundles_matrix) %>%
    head(1)  # Checking if first observation matches
```

```
ANSWER ##          1st          2nd          3rd    4th     5th
ANSWER ## Maroon5V "OddAnatomy" "beatsmusic" "xoxo" "alien" "word"
```

3. Top 5 recommendations (cosine similarity based) for “summerlovin”

```
knitr::kable(rec_mat["summerlovin", ], caption = "Top 5")
```

Table 2: Top 5

	x
1st	sassyhween
2nd	superherodad2
3rd	tropicalparadise
4th	mmlm
5th	julyfourth

ii.) Creating correlation based recommendations

1 & 2. Reuse the function(rec_mat_fun) with an accounts-bundles matrix where each bundle (column) has already been mean-centered

```r
bundle_means <- apply(ac_bundles_matrix, 2, mean)
bundle_means_matrix <- t(replicate(nrow(ac_bundles_matrix), bundle_means))
ac_bundle_mc_b <- ac_bundles_matrix - bundle_means_matrix
cor_rec <- rec_mat_fun(ac_bundle_mc_b)
knitr::kable(head(cor_rec, 5), caption = "Correlation Based Recommendations (Sample of 5)",
    align = "c")
```

Table 3: Correlation Based Recommendations (Sample of 5)

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

3. Top 5 recommendations (correlation based) for "summerlovin"

```r
knitr::kable(cor_rec["summerlovin", ], caption = "Top 5")
```

Table 4: Top 5

	x
1st	sassyhween
2nd	superherodad2
3rd	tropicalparadise
4th	mmlm
5th	julyfourth

iii.) Creating adjusted-cosine based recommendations

1 & 2. Reuse the function(rec_mat_fun) with an accounts-bundles matrix where each bundle (row has already been mean-centered

```
accounts_means <- apply(ac_bundles_matrix, 1, mean)
accounts_means_matrix <- replicate(ncol(ac_bundles_matrix), accounts_means)
ac_bundles_mc_b <- ac_bundles_matrix - accounts_means_matrix
ad_rec_mat <- rec_mat_fun(ac_bundles_mc_b)
knitr::kable(head(ad_rec_mat, 5), caption = "Cosine Adjusted Based Recommendations (Sample of 5)",
    align = "c")
```

Table 5: Cosine Adjusted Based Recommendations (Sample of 5)

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

3. Top 5 recommendations (adjusted-cosine based) for "summerlovin"

```
knitr::kable(ad_rec_mat["summerlovin", ], caption = "Top 5")
```

Table 6: Top 5

	x
1st	justmytype
2nd	mmlm
3rd	bestdaddy
4th	sweetmothersday
5th	julyfourth

c) Are the three sets of geometric recommendations similar in nature from intuition?

- ANSWER ## The three sets are different from intuition perhaps because I related the similarity based on key words and the themes related to love, when in fact the bundles were not matched solely on this.
- ANSWER ## We can also see that the cosine and correlation based recommendations provided the same top 5 result, whereas the adjust cosine based recommendations had a few differences.

d) The conceptual difference in cosine similarity, correlation, and adjusted-cosine.

- ANSWER ## The cosine similarity computes the similarity between two samples, whereas correlation computes the similarity between two jointly distributed random variables.
- ANSWER ## Correlation is also referred to as a mean centered cosine.

2) Exploring correlation by running a simulation

- Source: "demo_simple_regression.R"
- Function: interactive_regression()

a) Creating a horizontal set of random points, with a relatively narrow but flat distribution

```
knitr::include_graphics("Rplot_2a.pdf")  # Importing plot
```

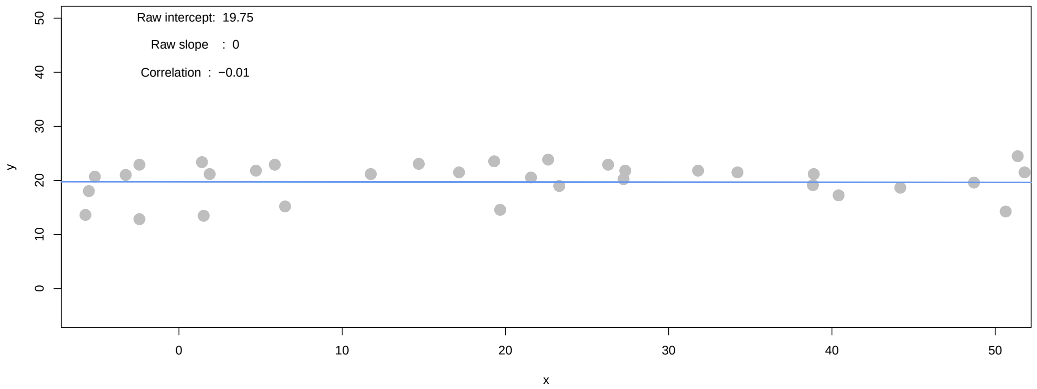

Figure 1: Flat Distribution

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

- ANSWER ## We would expect a **raw slope** to be **0** or close to 0 as shown in *figure 1*.

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

- ANSWER ## We would expect the **correlation** to be **0** or close to 0 as shown in *figure 1* since the points lie horizontally.

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

```
knitr::include_graphics("Rplot_2b.pdf")  # Importing plot
```

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

- ANSWER ## We would expect a **raw slope** to be **0** or close to 0 as shown in *figure 2*.

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

- ANSWER ## We would expect the **correlation** to be **0** or close to 0 as shown in *figure 2* since the points are randomly filled all along the x and y-axis.

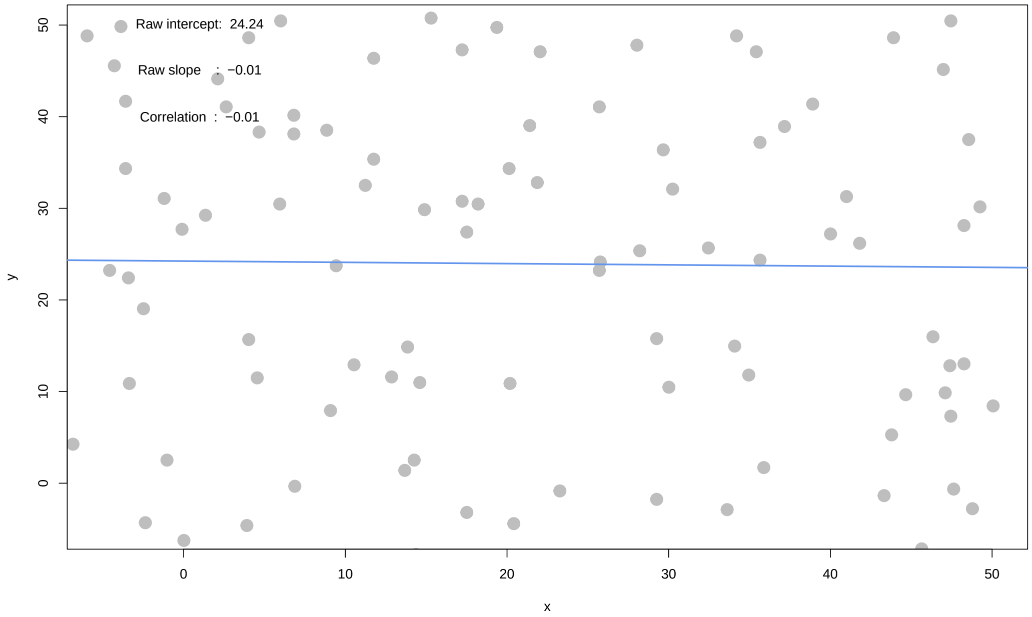

Figure 2: Random (filled) Distribution

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

```
knitr::include_graphics("Rplot_2c.pdf")  # Importing plot
```

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

- ANSWER ## We would expect a **raw slope** to be **1** or close to 1 as shown in *figure 3*.

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

- ANSWER ## We would expect the **correlation** to be **1** or close to 1 as shown in *figure 3* since the points of x have the same positive trend with y.

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

```
knitr::include_graphics("Rplot_2d.pdf")  # Importing plot
```

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

- ANSWER ## We would expect a **raw slope** to be **-1** or close to -1 as shown in *figure 4*.

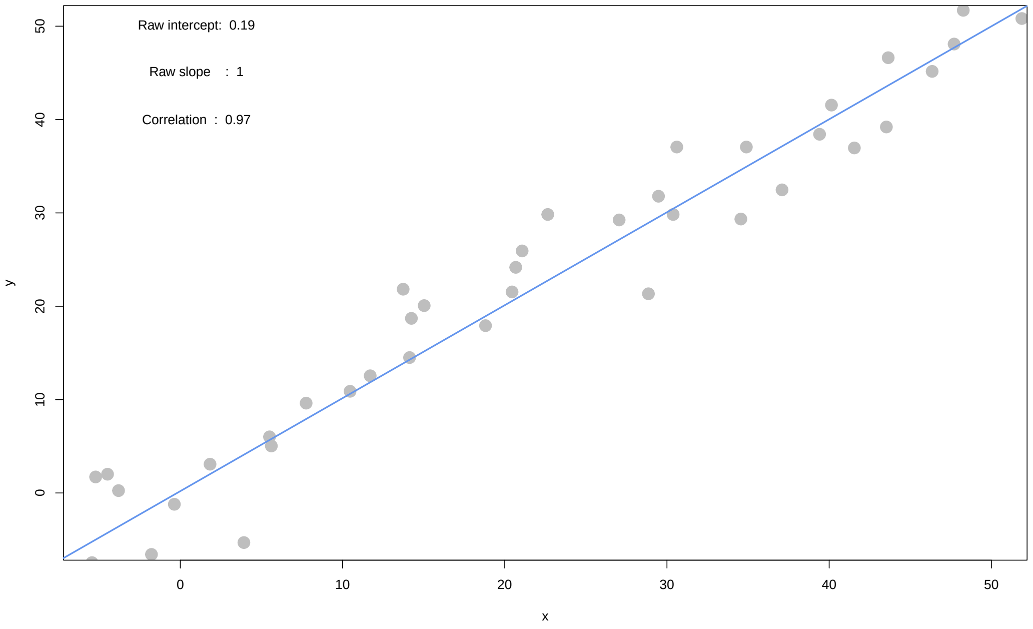

Figure 3: Diagonal Distribution (45 degrees upwards)

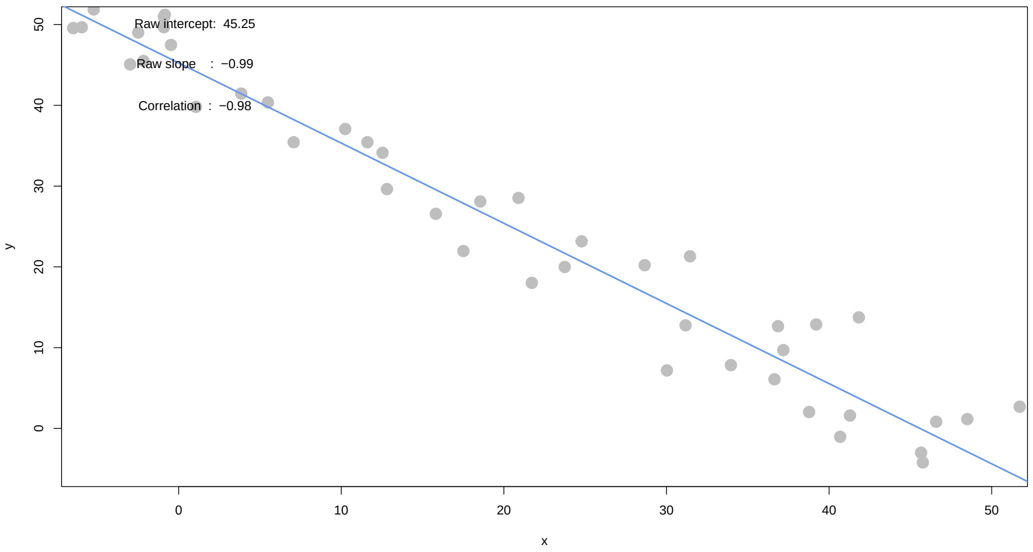

Figure 4: Diagonal Distribution (45 degrees downwards)

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

- ANSWER ## We would expect the **correlation** to be **-1** or close to -1 as shown in *figure 4* since the points of x have the same positive trend with y.

e) Creating a pattern of data points with no correlation, but visually suggests a strong relationship

```r
knitr::include_graphics("Rplot_2e.pdf")  # Importing plot
```

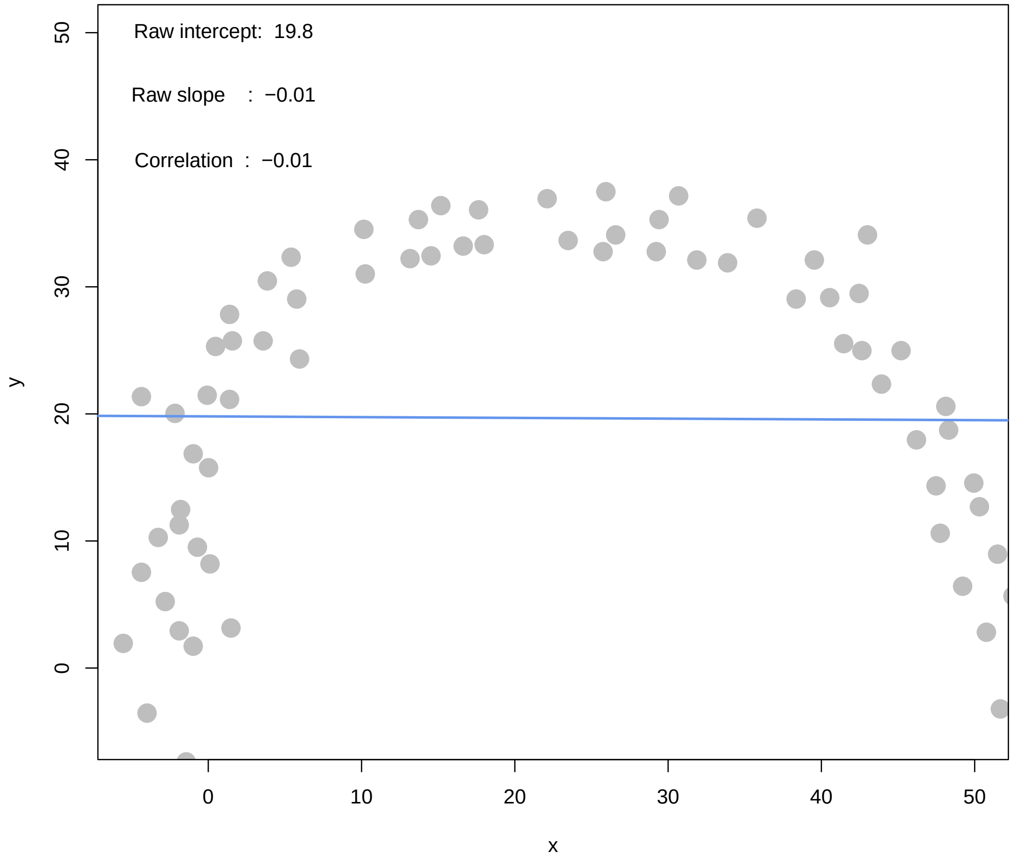

Figure 5: Scenario E

- ANSWER ## The correlation is close to **0**, but visually suggests a strong relationship in *figure 5*.

f) Creating a pattern of data points with perfect correlation, but visually suggests a different relationship

```
knitr::include_graphics("Rplot-2f.pdf")  # Importing plot
```

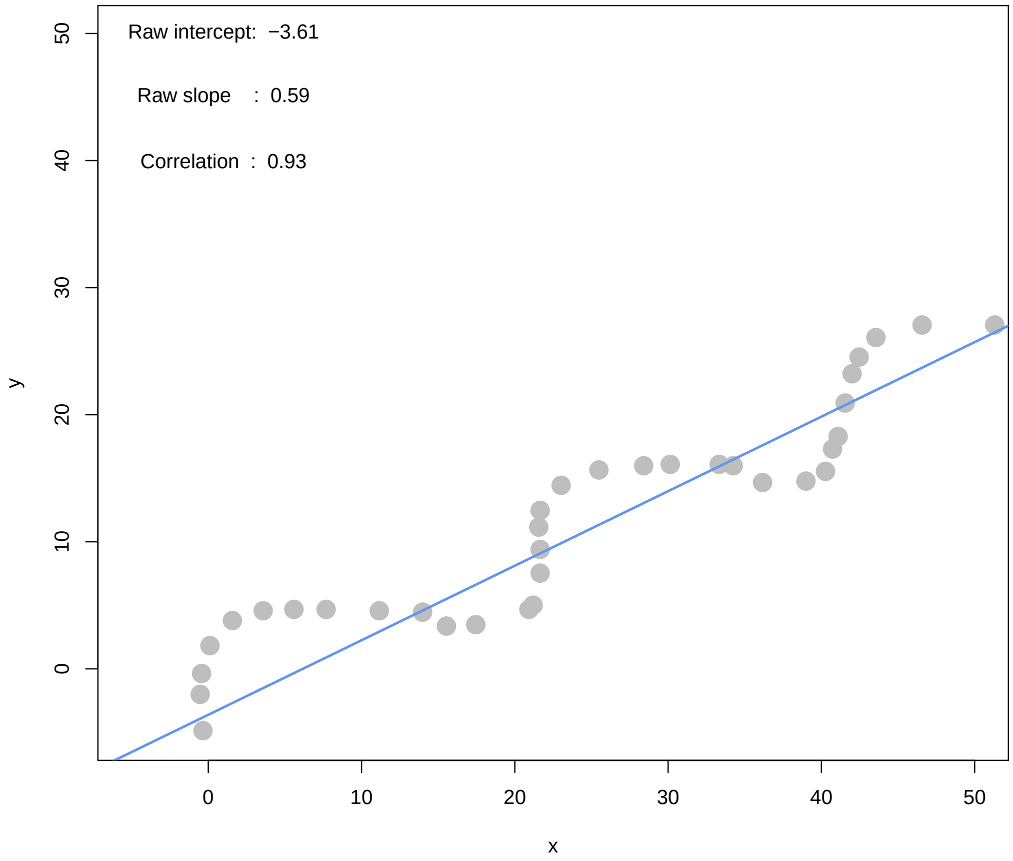

Figure 6: Scenario F

- ANSWER ## The correlation is close to **1**, but visually suggests a different relationship in *figure 6*.

g) Let's see how correlation relates to simple regression by simulating a linear relationship

i) Run the simulation and show a record of the points

```
knitr::include_graphics("Rplot_2g plot.pdf")  # Importing plot
```

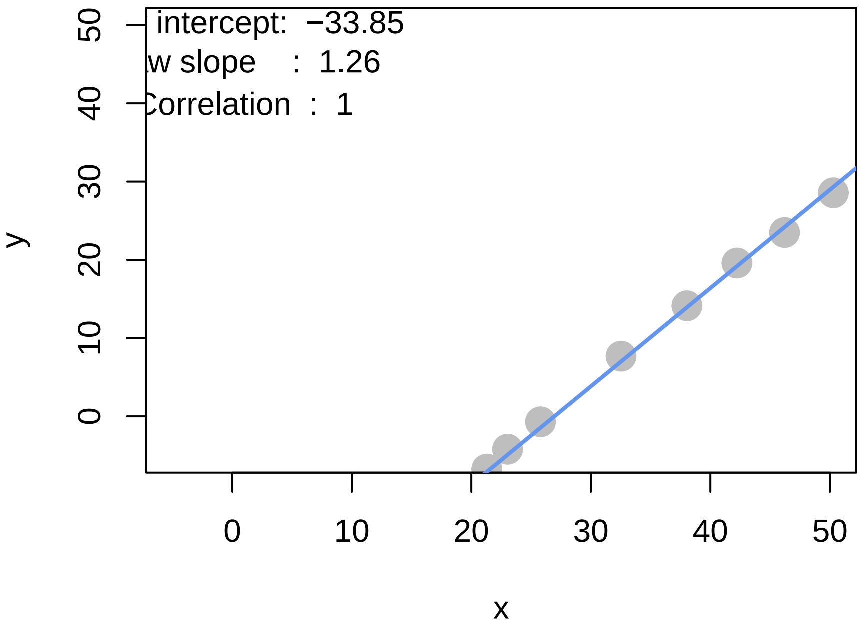

Figure 7: Scenario G

```
# source('demo_simple_regression.R')

# Showing value points of graphs
pts <- data.frame(x = c(16.5992, 18.13055, 21.29534, 23.03087,
    25.7873, 32.52523, 38.03809, 42.22377, 46.20528, 50.28888,
    54.16829, 54.67874), y = c(-13.7713835, -12.2104848, -6.7473394,
    -4.210879, -0.6988569, 7.6909736, 14.1296807, 19.5928262,
    23.4950729, 28.5679937, 34.2262515, 35.0067008))
```

ii) Estimating the regression intercept and slope of pts to ensure they are the same as the values reported in the simulation plot (Scenario G)

```
pts_summary <- summary(lm(pts$y ~ pts$x))  # Running regression and adding variable
pts_summary$coefficients  # Printing only coefficients to show intercept and slope
```

```
ANSWER ##                Estimate Std. Error   t value     Pr(>|t|)
ANSWER ## (Intercept) -33.847546 0.55776086 -60.68469 3.588485e-14
ANSWER ## pts$x         1.255974 0.01476308  85.07529 1.231170e-15
```

- ANSWER ## The values reported in the simulation approximately **match**.

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

```
cor(pts)  # Checking correlation
```

```
ANSWER ##           x         y
ANSWER ## x 1.0000000 0.9993099
ANSWER ## y 0.9993099 1.0000000
```

- ANSWER ## The values reported in the simulation approximately **match**.

iv) Standardizing the values of both x and y from “pts” and re-estimating the regression slope

```
pts_sd <- data.frame(x = scale(pts$x), y = scale(pts$y))
pts_scaled_summary <- summary(lm(pts_sd$y ~ pts_sd$x))
pts_scaled_summary$coefficients  # Printing only coefficients to show intercept and slope
```

```
ANSWER ##                   Estimate Std. Error       t value    Pr(>|t|)
ANSWER ## (Intercept) -2.563066e-16 0.01124611 -2.279068e-14 1.00000e+00
ANSWER ## pts_sd$x     9.993099e-01 0.01174618  8.507529e+01 1.23117e-15
```

- ANSWER ## Standardizing “pts” **changed** the intercept and slope values

```
cor(pts_sd)  # Print
```

```
ANSWER ##           x         y
ANSWER ## x 1.0000000 0.9993099
ANSWER ## y 0.9993099 1.0000000
```

- The correlation of x and y **did not** change.

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

- ANSWER ## They have a 1:1 relationship with an intercept of 0