Business Analytics using Forecasting Assignment 9

Question 1

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

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

```
## Warning: package 'data.table' was built under R version 3.2.5
```

```
ac_bundles <- fread("../9-piccollage_accounts_bundles.csv")
ac_bundles <- as.matrix(ac_bundles[, -1, with=FALSE])
```

i. Download PicCollage onto your mobile from the iOS/Android appstores and take a look at the style and content of various bundles in their Sticker Store: how many recommendations does each bundle have?

6

ii. Find a single sticker bundle that is both in our limited data set and also in the app's Store (e.g., "sweetmothersday") — use your intuition to recommend (guess!) five other bundles that might have similar usage patterns as this bundle.

sweetmothersday => (bestdaddy, Dad2013, Mom2013, CampusLife, HeartStickerPack)

b. Let's find similar bundles using geometric methods:

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

```
library(lsa)
```

```
## Warning: package 'lsa' was built under R version 3.2.5
```

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

```
cos_sim_matrix <- cosine(ac_bundles)
diag(cos_sim_matrix) <- NA
bundle_reco <- function(bundle_cos) names(sort(bundle_cos, decreasing = TRUE))
cos_all_recos <- t(apply(cos_sim_matrix, 1, bundle_reco))
cos_top5_recos <- cos_all_recos[, 1:5]
head(cos_top5_recos)
```

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

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.

```
top5_recos <- function(ac_bundles_matrix) {
  sim_matrix <- cosine(ac_bundles_matrix)
  diag(sim_matrix) <- NA
  bundle_reco <- function(bundle_cos) names(sort(bundle_cos, decreasing = TRUE))
  all_recos <- t(apply(sim_matrix, 1, bundle_reco))
  top5_recos <- all_recos[, 1:5]

  return(top5_recos)
}

test_function <- top5_recos(ac_bundles)
identical(cos_top5_recos, test_function)
```

```
## [1] TRUE
```

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

```
cos_top5_recos['sweetmothersday', ]
```

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

ii. Let's create correlation based recommendations.

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

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

```
col_mean <- sapply(as.data.frame(ac_bundles), mean)
col_mean_matrix <- t(replicate(nrow(ac_bundles), col_mean))
bundles_col_mc <- as.matrix(ac_bundles - col_mean_matrix)
cor_top5_recos <- top5_recos(bundles_col_mc)
head(cor_top5_recos)
```

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

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

```
cor_top5_recos['sweetmothersday', ]
```

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

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

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

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

```
row_mean <- apply(as.data.frame(ac_bundles), 1, mean)
row_mean_matrix <- replicate(ncol(ac_bundles), row_mean)
bundles_row_mc <- as.matrix(ac_bundles - row_mean_matrix)
adj_cos_top5_recos <- top5_recos(bundles_row_mc)
head(adj_cos_top5_recos)
```

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

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

```
adj_cos_top5_recos['sweetmothersday', ]
```

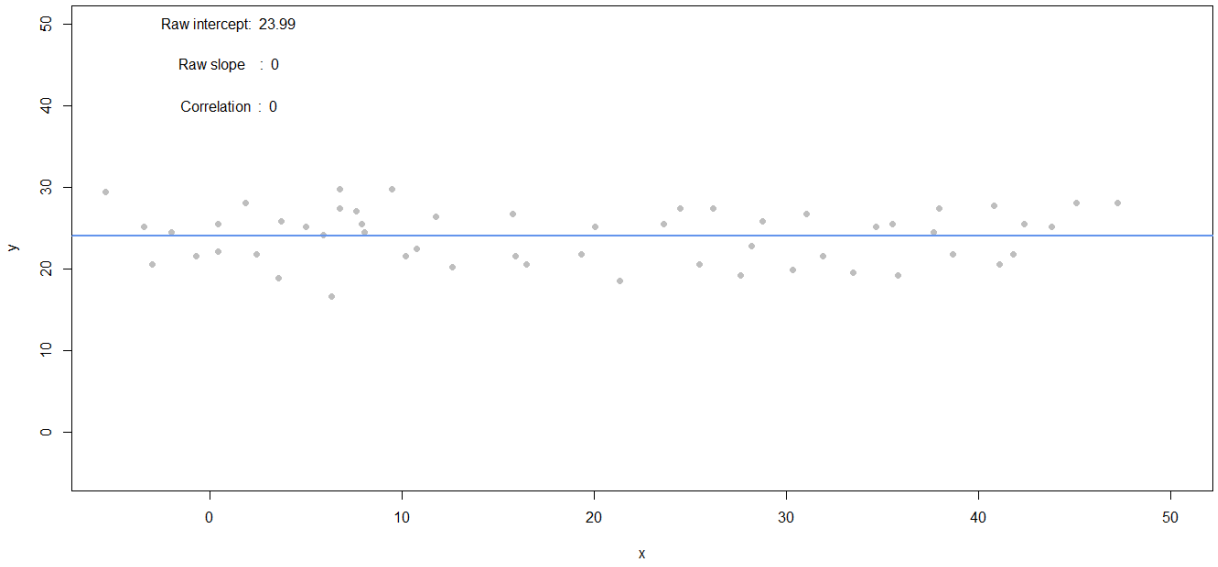
```
## [1] "justmytype" "julyfourth" "gudetama"   "mmlm"       "bestdaddy"
```

c. (not graded)

Question 2

For each of the scenarios below, create a set of points matching the description. You might have to create each scenario a few times to get a general sense of each. Visual examples of the first four scenarios is shown below.

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



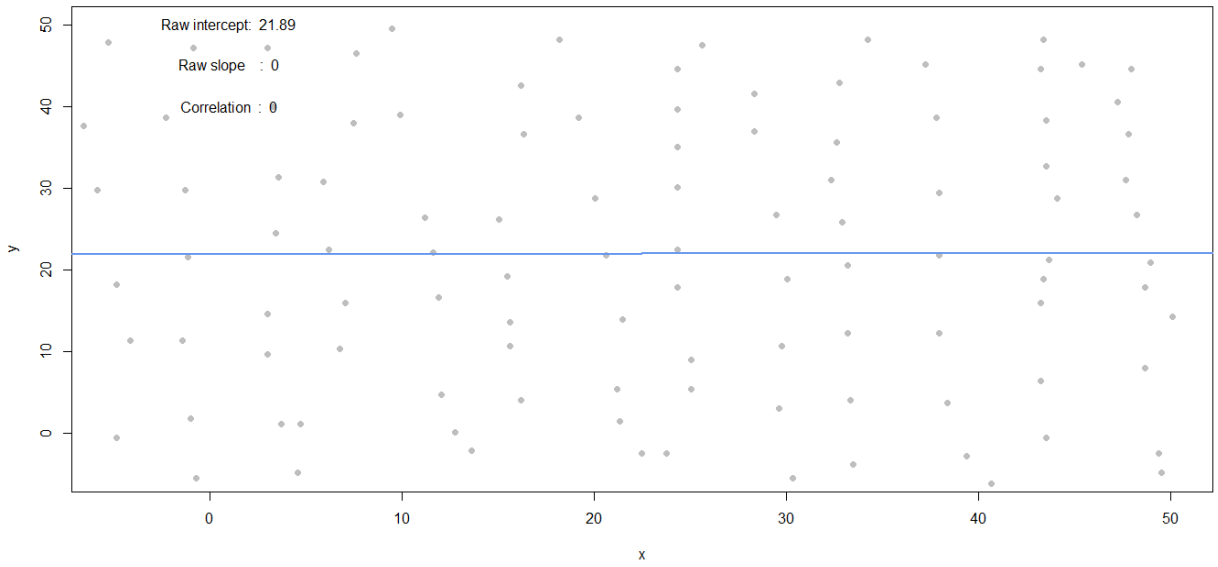
i. What raw slope of the x and y would you generally expect?

0

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

0

b. Create a completely random set of points ranging all along the entire x-axis and y-axis (i.e., fill the plot)



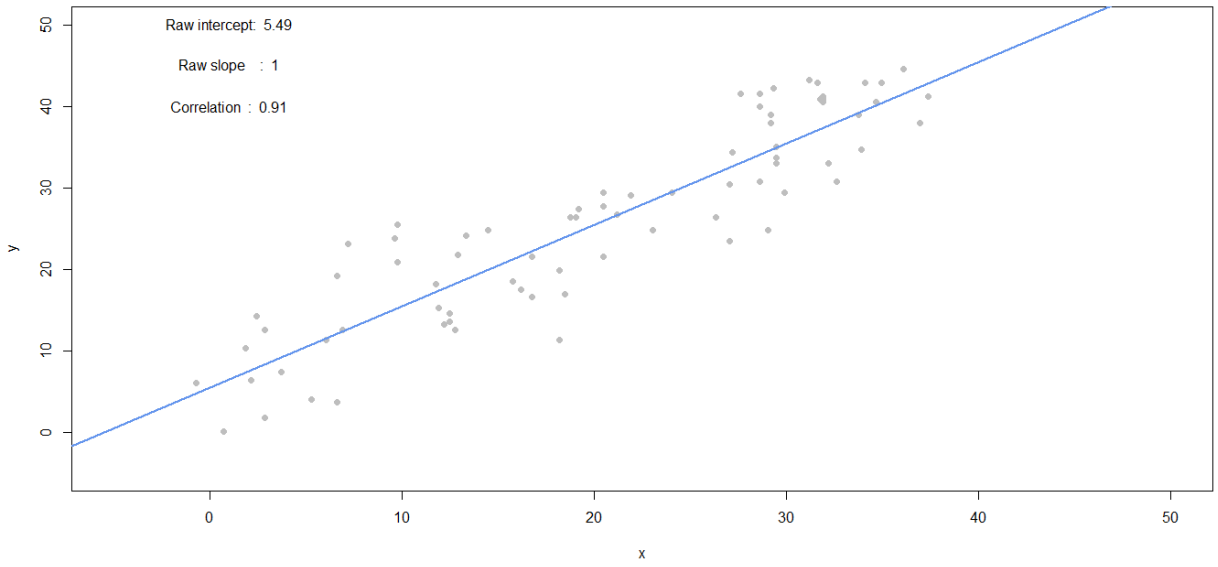
i. What raw slope of the x and y would you generally expect?

0

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

0

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



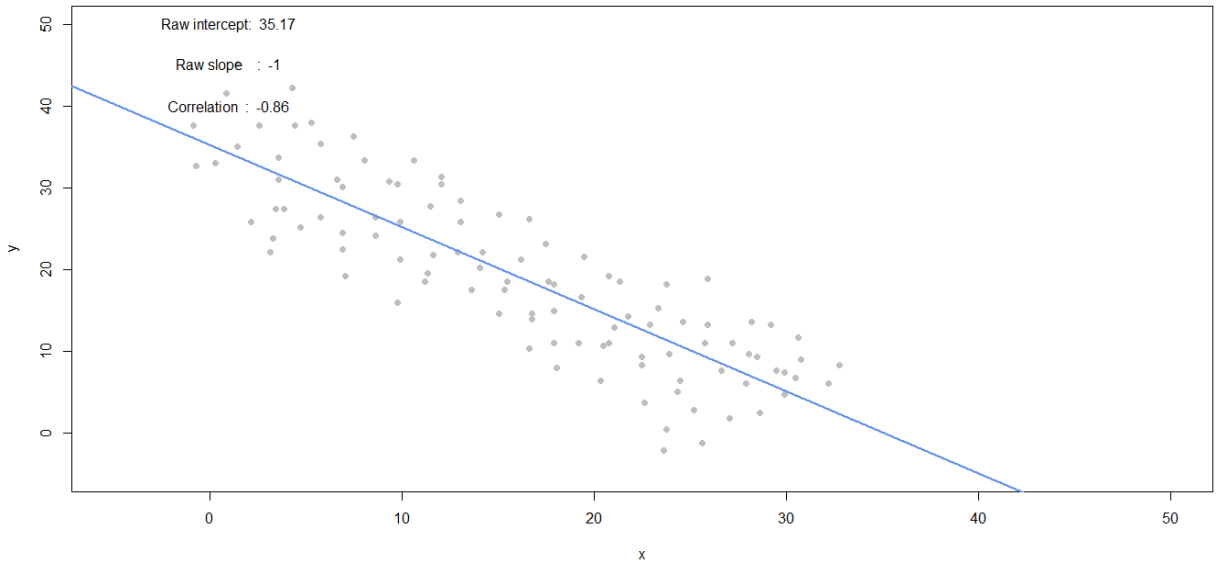
i. What raw slope of the x and y would you generally expect? (note that x, y have the same scale)

1

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

1

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



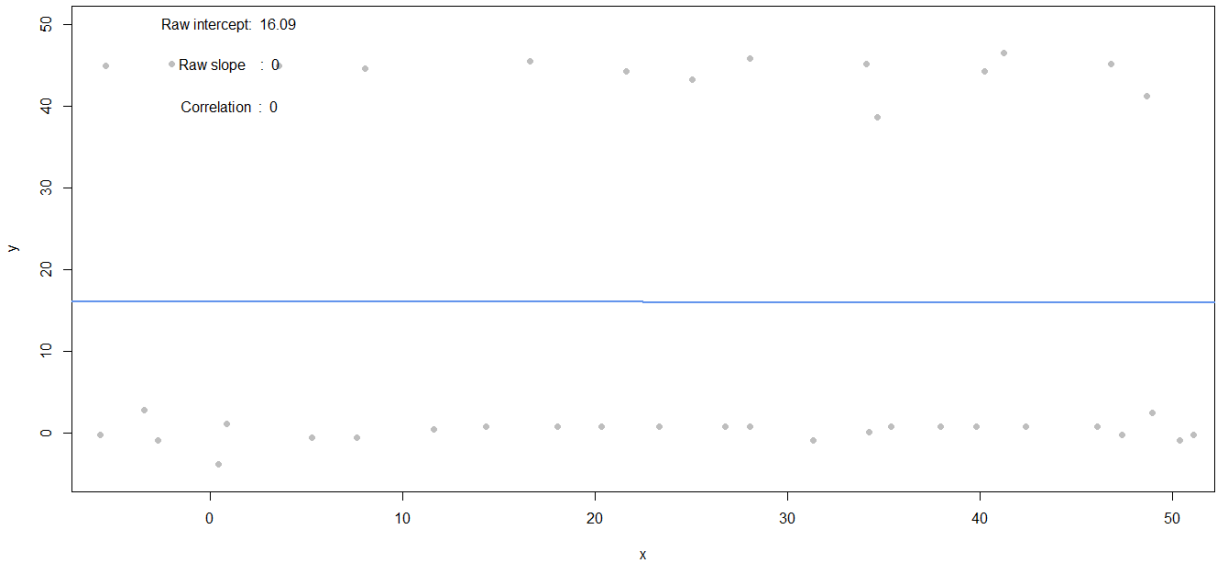
i. What raw slope of the x and y would you generally expect? (note that x, y have the same scale)

-1

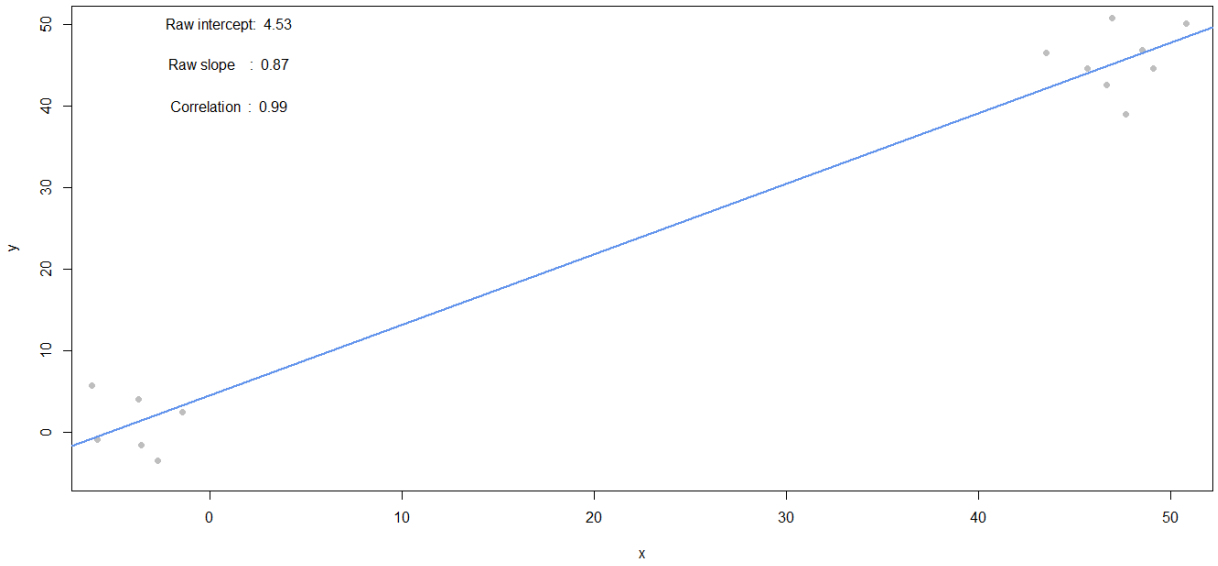
ii. What is the correlation of x and y that you would generally expect?

-1

e. Apart from any of the above scenarios, find another pattern of data points with no correlation (r 0). (challenge: can you find a scenario where the pattern visually suggests a relationship?)



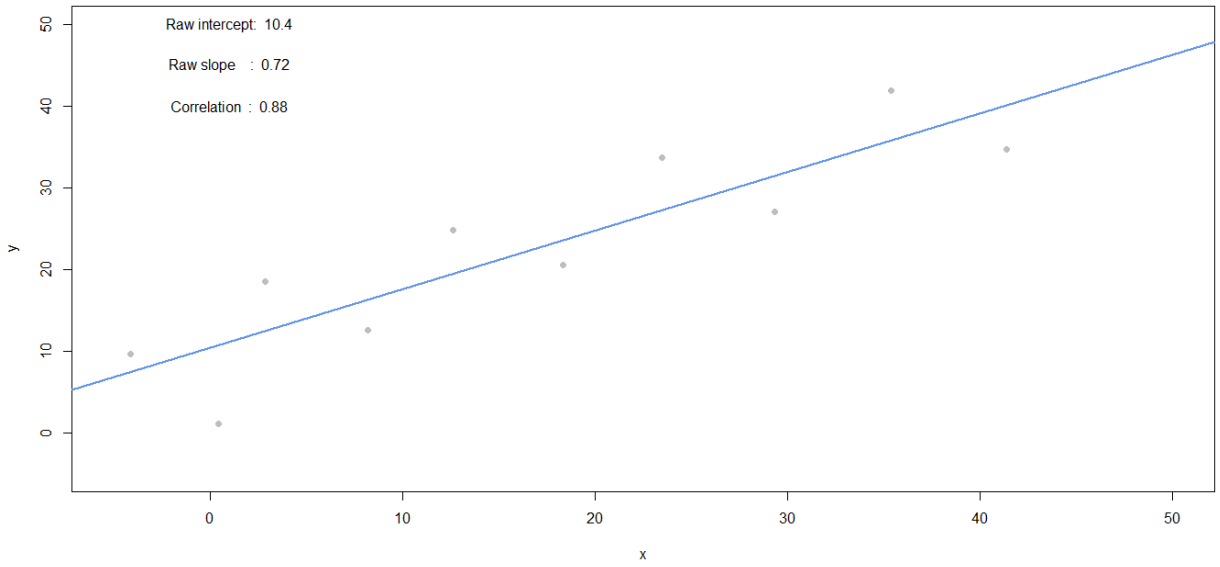
f. Apart from any of the above scenarios, find another pattern of data points with perfect correlation (r 1). (challenge: can you find a scenario where the pattern visually suggests a different relationship?)



g. Let's find the relationship between correlation and regression

i. Run the simulation and capture the points you create

```
pts <- interactive_regression()
```



ii. 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
-9.6430 -4.1975 -0.4227  5.8838  6.4230

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.3966     3.0088   3.455 0.008626 **
pts$x         0.7176     0.1351   5.312 0.000718 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.258 on 8 degrees of freedom
Multiple R-squared:  0.7791,    Adjusted R-squared:  0.7515
F-statistic: 28.21 on 1 and 8 DF,  p-value: 0.0007183
```

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

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

[1] 0.8826578
```

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

```
pts_std <- as.data.frame(pts_std)
summary(lm(pts_std$y ~ pts_std$x))

Call:
lm(formula = pts_std$y ~ pts_std$x)

Residuals:
     Min       1Q   Median       3Q      Max
-0.76813 -0.33436 -0.03367  0.46868  0.51164

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.374e-17  1.576e-01   0.000 1.000000
pts_std$x   8.827e-01  1.662e-01   5.312 0.000718 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4985 on 8 degrees of freedom
Multiple R-squared:  0.7791,    Adjusted R-squared:  0.7515
F-statistic: 28.21 on 1 and 8 DF,  p-value: 0.0007183
```

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

The correlation and the standardized regression estimates are the same.