BACS - HW (Week 10)

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

Download the CSV file `piccollage_accounts_bundles.csv` from Canvas.
You may either use `read.csv()` and `data.frame` to load the file as before, or you can try learning how to use `data.table` – a high performance package for reading, writing, and managing large data sets.
Note: It will take time to fully learn data.table — but here's some code to get you started:

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

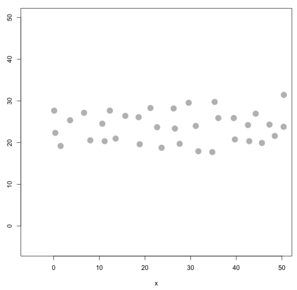
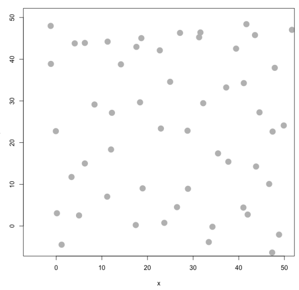
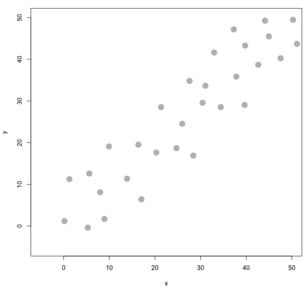
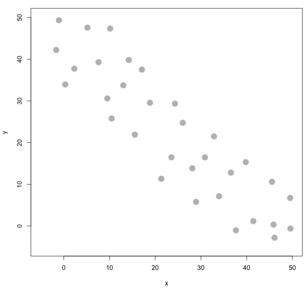
a. Let's explore to see if any sticker bundles seem intuitively similar:
   i. ***(recommended)*** Download PicCollage onto your mobile from the iOS/Android app store and take a look at the style and content of various bundles in their Sticker Store: how many recommendations does each bundle have?
   ii. Find a single sticker bundle that is both in our limited data set and also in the app's Sticker Store (e.g., "sweetmothersday"). Then, use your intuition to recommend (guess!) five other bundles in our dataset that might have *similar* usage patterns as this bundle.
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
      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.
      3. What are the top 5 recommendations for the bundle you chose to explore earlier?
   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 already been mean-centered in advance.
      3. Now what are the top 5 recommendations for the bundle you chose to explore earlier?
   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 already been mean-centered in advance.
      3. What are the top 5 recommendations for the bundle you chose to explore earlier?
c. *(not graded)* 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?
d. *(not graded)* What do you think is the conceptual difference in cosine similarity, correlation, and adjusted-cosine?

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

Download `demo_simple_regression.R` from Canvas – it has a function called `interactive_regression()` that runs a simulation. You can click to add points to the plotting area, to see a corresponding regression line (hitting ESC will stop the simulation). You will also see three numbers: regression intercept – where the regression line crosses the y-axis; regression coefficient – the slope of x on y; correlation - correlation of x and y.

For each of the scenarios below, create the described set of points in the simulation. You might have to create each scenario a few times to get a general sense of them. Visual examples of scenarios a-d are shown below.

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

Scenario A	*Scenario B*	*Scenario C*	*Scenario D*
			

e. Apart from any of the above scenarios, find another pattern of data points with no correlation ($r \approx 0$). *(optionally: can create a pattern that visually suggests a strong relationship but produces* $r \approx 0$*?)*
f. Apart from any of the above scenarios, find another pattern of data points with perfect correlation ($r \approx 1$). *(optionally: can you find a scenario where the pattern visually suggests a different relationship?)*
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: `pts <- interactive_regression()`
   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 ))`
   iii. Estimate the correlation of `x` and `y` to see it is the same as reported in the plot: `cor(pts)`
   iv. Now, re-estimate the regression using *standardized* values of both `x` and `y` from `pts`
   v. What is the relationship between *correlation* and the *standardized simple-regression estimates*?