STAT 33B Homework 2

Gunnar Mayer (3034535154)

This assignment is due **Feb 19, 2020** by 11:59pm.

The purpose of this assignment is to practice working with data frames, including loading tabular data, taking subsets, and making plots.

Edit this file, knit to PDF, and:

- Submit the Rmd file on bCourses.
- Submit the PDF file on Gradescope.

If you think you'll need help with submission, please ask in office hours *before* the assignment is due.

Answer all questions with complete sentences, and put code in code chunks. You can make as many new code chunks as you like. Please do not delete the exercises already in this notebook, because it may interfere with our grading tools.

Working with Data

1. In lecture, you saw that the `readRDS()` function can read data stored in R's built-in RDS format. Tabular data is often distributed online as tab-separated value (TSV) or comma-separated value (CSV) files.

   In a TSV file, each row of the data set is one one line, with entries in the columns separated by tabs.

   For this assignment, you'll use the Datasaurus Dozen data set, which is available on the bCourse (`DatasaurusDozen.tsv`).

   Read the documentation for `read.delim()` to figure out how to load the Datasaurus Dozen data set into R.

   Assign the data set to the `dsaur` variable.

```
dsaur = read.delim(file = "DatasaurusDozen.tsv")
```

2. Now that you've loaded the data set, print out summary information, including:

   - Number of columns
   - Number of rows
   - Classes of the columns
   - Levels in the `dataset` column

"Number of Columns"

```
ncol(dsaur)
```

```
## [1] 3
```

"Number of Rows"

```
nrow(dsaur)
```

```
## [1] 1846
```

"Classes of the Columns"

```
sapply(dsaur, class)
```

```
##   dataset         x         y
##  "factor" "numeric" "numeric"
```

"Levels of the dsaur$dataset column"

```
levels(dsaur$dataset)
```

```
##  [1] "away"       "bullseye"   "circle"     "dino"       "dots"
##  [6] "h_lines"    "high_lines" "slant_down" "slant_up"   "star"
## [11] "v_lines"    "wide_lines" "x_shape"
```

3. The Datasaurus Dozen is actually a collection of 12 data sets stacked together. The `dataset` column indicates which data set each row comes from.

    a. Use subsetting to extract only the rows in the `dino` data set. Assign those rows to the `dino` variable.

    b. Compute the mean and standard deviation for the `x` and `y` columns in the `dino` data set.

    Repeat these two steps for the `star` dataset.

    Based on these statistics, are the two data sets similar?

```
#Dino Code
# a.
dino = subset(dsaur, dsaur$dataset == 'dino')
head(dino)
```

```
##   dataset       x       y
## 1    dino 55.3846 97.1795
## 2    dino 51.5385 96.0256
## 3    dino 46.1538 94.4872
## 4    dino 42.8205 91.4103
## 5    dino 40.7692 88.3333
## 6    dino 38.7179 84.8718
```

```
# b.
"Dino x mean"
```

```
## [1] "Dino x mean"
```

```r
d_x_mean = mean(dino$x)
d_x_mean
```

```
## [1] 54.26327
```

```r
"Dino x standard deviation"
```

```
## [1] "Dino x standard deviation"
```

```r
d_x_sd = sd(dino$x)
d_x_sd
```

```
## [1] 16.76514
```

```r
"Dino y mean"
```

```
## [1] "Dino y mean"
```

```r
d_y_mean = mean(dino$y)
d_y_mean
```

```
## [1] 47.83225
```

```r
"Dino y standard deviation"
```

```
## [1] "Dino y standard deviation"
```

```r
d_y_sd = sd(dino$y)
d_y_sd
```

```
## [1] 26.9354
```

```r
#Star Code
# a.
star = subset(dsaur, dsaur$dataset == 'star')
head(star)
```

```
##     dataset        x        y
## 711    star 58.21361 91.88189
## 712    star 58.19605 92.21499
## 713    star 58.71823 90.31053
## 714    star 57.27837 89.90761
## 715    star 58.08202 92.00815
## 716    star 57.48945 88.08529
```

```r
# b.
"Star x mean"
```

```
## [1] "Star x mean"
```

```r
s_x_mean = mean(star$x)
s_x_mean
```

```
## [1] 54.26734
```

```r
"Star x standard deviation"
```

```
## [1] "Star x standard deviation"
```

```r
s_x_sd = sd(star$x)
s_x_sd
```

```
## [1] 16.76896
```

```r
"Star y mean"
```

```
## [1] "Star y mean"
```

```r
s_y_mean = mean(star$y)
s_y_mean
```

```
## [1] 47.83955
```

```r
"Star y standard deviation"
```

```
## [1] "Star y standard deviation"
```

```r
s_y_sd = sd(star$y)
s_y_sd
```

```
## [1] 26.93027
```

Your written answer goes here: Based on these statistics, are the two data sets similar?

Yes the data sets are similar. Based on the means and standard deviations found above it appears that the star and dino data sets are very similar.

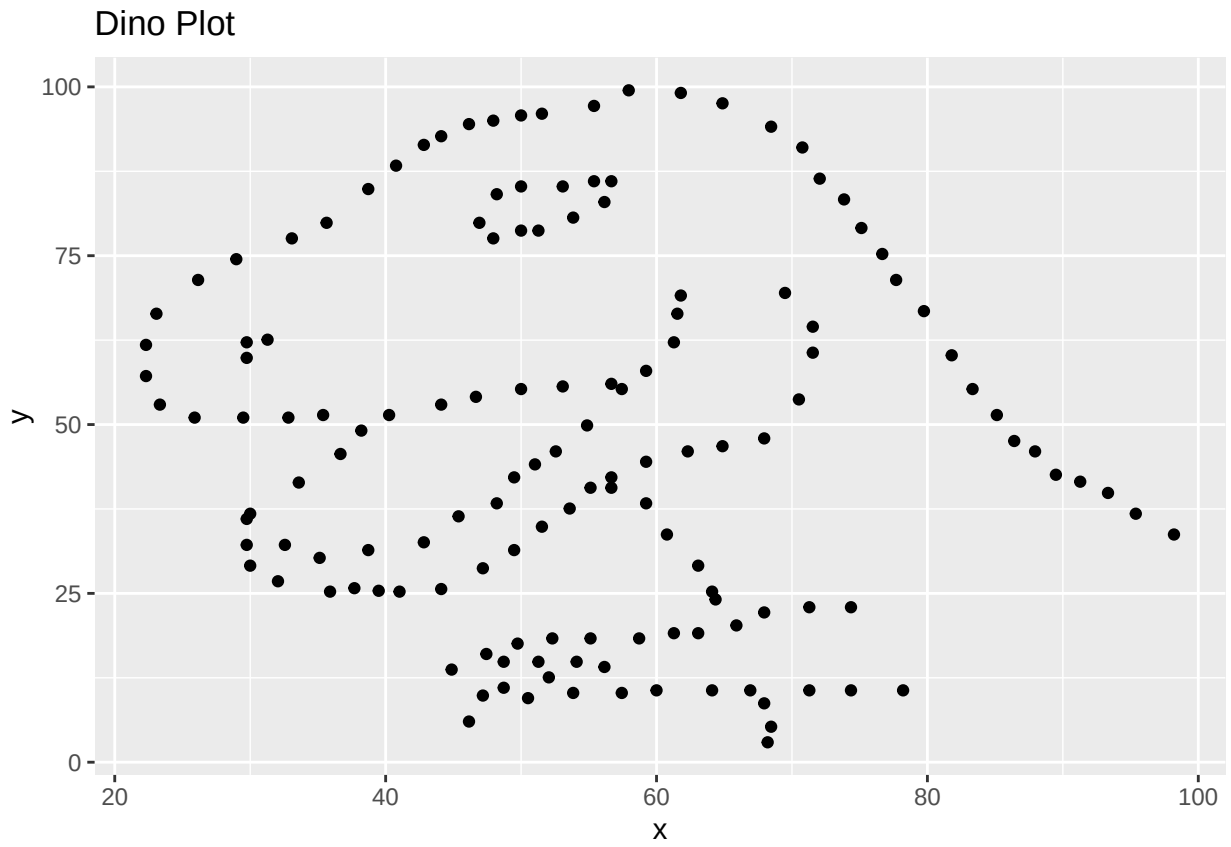
~~~~~~~~~~~~~~

4. Use `ggplot2` to make a scatter plot of `x` versus `y` for the `dino` data set. Make sure your plot includes a title.

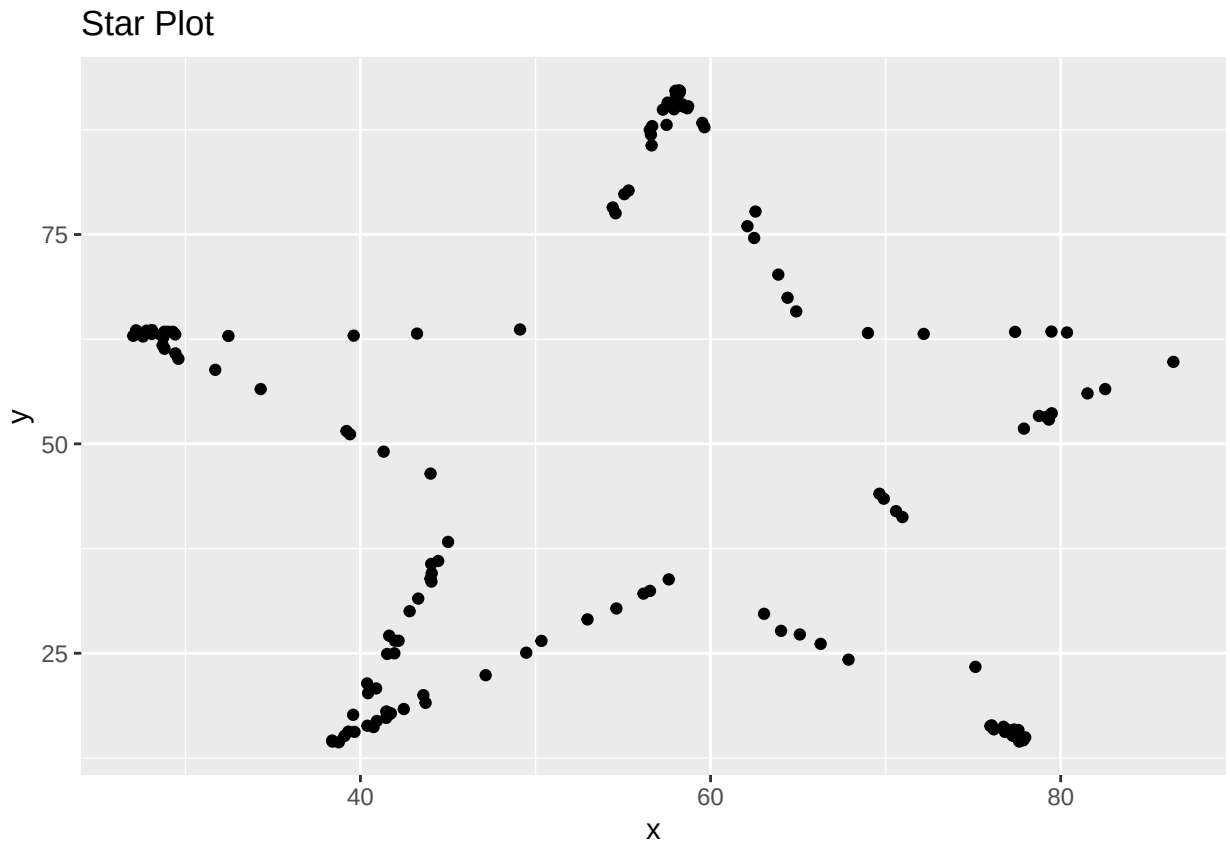
   Repeat for the `star` data set.

   Based on these plots, are the two data sets similar?

```r
# Dino Code
library(ggplot2)
ggplot(data = dino, aes(x = x, y = y)) +
  geom_point() +
  labs(title = "Dino Plot")
```
~~~~~~~~~~~~~~




```
ggplot(data = star, aes(x = x, y = y)) +
  geom_point() +
  labs(title = "Star Plot")
```



Your written answer goes here: Based on these plots, are the two data sets similar?

No, after visually inspecting the scatter plots of the 'dino' and 'star' data sets it is apparent that they are not similar. However, they both share the fact that they look cool.