XXXXX

STA 325: Homework 3

General instructions for homeworks: Your code must be completely reproducible and must compile. No late homeworks will be accepted.

Advice: Start early on the homeworks and it is advised that you not wait until the day of as these homeworks are meant to be longer and treated as case studies.

Commenting code Code should be commented. See the Google style guide for questions regarding commenting or how to write code https://google.github.io/styleguide/Rguide.xml.

R Markdown Test

0. Open a new R Markdown file; set the output to HTML mode and "Knit". This should produce a web page with the knitting procedure executing your code blocks. You can edit this new file to produce your homework submission.

Total points on assignment: 5 (reproducibility) + XXX points for the assignment.

Download/install the RecordLinkage and tree libraries. You will use the RLdata500 data set in R which you can load into your workspace by (after typing the command library(RecordLinkage)) typing data(RLdata500). Typing ls( ) then shows two objects – the actual text data RLdata500 and identity.RLdata500, which are the unique identifiers. See help(RLdata500) for additional details.

Our goal is to build supervised models to predict whether or not pairs of records match.

To do so, we need labels (which we have, luckily, using identity.RLdata500) and similarity metrics for the 124,750 pairs of records.

1. Create a matrix with 124,750 rows and 6 columns. The rows correspond to pairs of records. Columns correspond to the similarity metric for the different fields for the pair of records. For example, the first row would be for the record pair (1,2); the second row for (1,3), etc. The first column is for fname c1, the second for fname c2, etc. The last column is a binary indicator of whether or not that pair of records matches (1 = yes; 0 = no).

Fill in the first seven columns of your matrix with the Jaro-Winkler scores for each field; the last column should indicate match/non-match. In this problem, treat all fields as text strings, even the birthdate information.

The JW score requires character strings; you may need to use as.character( ) on the field values. Also, if one or both of the strings is NA, JW = NA.

One coding suggestion would be to first create a 500 x 500 matrix of JW scores for each field and then build the larger matrix by extracting the upper triangular part of the matrices for each field.

```
data(RLdata500)
true = identity.RLdata500
my.data <- RLdata500[, -c(2,4)]
my.data = cbind(my.data, true)
dtf = calc.pcs(my.data)
#dtf = calc.pcs(my.data, type = c(rep("l", ncol(my.data)-1), "e"))
head(dtf)
```

```
##   fname_c1.comp lname_c1.comp   by.comp bm.comp bd.comp true.comp index1 index2
## 1     0.4642857     0.6000000 0.7333333       1    0.70 0.0000000      1      2
## 2     0.5317460     0.4416667 0.7333333       0    0.00 0.0000000      1      3
## 3     0.4365079     0.0000000 0.7333333       0    0.85 0.0000000      1      4
## 4     0.4285714     0.6761905 0.7333333       0    0.00 0.0000000      1      5
## 5     0.6190476     0.4555556 0.8666667       1    0.00 0.6111111      1      6
## 6     0.4642857     0.6583333 0.7333333       0    0.00 0.0000000      1      7
```

```
pred <- dtf$true.comp
#choose(500,2)
```

2. Use your matrix to fit a logistic regression model predicting whether or not the pairs of records are a match using all fields except fname c2, lname c2 as predictor variables. Which fields were significant predictors of being a match? Does anything surprise you? Can you think of why this might have happened by digging more into the data set and thinking about if the model is appropriate for this data set.

```
#tail(true <- calc.pcs(data.frame)[,c(6)])
logistic.regression <- glm(pred ~ ., data = dtf, family = "binomial")
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
xtable(summary(logistic.regression)$coef)
```

```
## % latex table generated in R 4.3.1 by xtable 1.8-4 package
## % Thu Aug 15 19:50:14 2024
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrr}
##   \hline
##  & Estimate & Std. Error & z value & Pr($>$$|$z$|$) \\
##   \hline
## (Intercept) & -5.89 & 0.08 & -76.96 & 0.00 \\
##   fname\_c1.comp & 0.01 & 0.05 & 0.26 & 0.80 \\
##   lname\_c1.comp & -0.01 & 0.04 & -0.19 & 0.85 \\
##   by.comp & 0.04 & 0.04 & 0.98 & 0.33 \\
##   bm.comp & -0.01 & 0.03 & -0.36 & 0.72 \\
##   bd.comp & 0.00 & 0.03 & 0.04 & 0.96 \\
##   true.comp & 10.07 & 0.09 & 111.43 & 0.00 \\
##   index1 & 0.00 & 0.00 & 0.45 & 0.65 \\
##   index2 & 0.00 & 0.00 & 3.92 & 0.00 \\
##    \hline
## \end{tabular}
## \end{table}
```

```
warnings()
```

The model does not converge, and thus, we cannot interpret any of the results of it. Thinking more about the data set, it is unbalanced, where there are few duplicated records, thus, logistic regression is likely to be a poor choice for such situations. In short, we're going to need something more clever, even for such a small data set.

3. In the above, we chose to treat the numeric birthday variables as text (reasonable assumption). What if instead of using a JW score, we just used exact matching on birth year, month, and day? How would that change our results from 2)?