Example Solution to Homework 2

Olivier Binette

February 25, 2020

```
set.seed(1)
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE,
                      fig.width = 4, fig.height = 3, fig.align = "center")

if (!require(pacman)) install.packages("pacman")
pacman::p_load(tidyverse, RecordLinkage, kableExtra, visdat, cowplot, ggrepel)
pacman::p_load_gh("OlivierBinette/pretty")
```

The goal of this homework is to investigate how "representative" samples can be obtained in the context of entity resolution, for the purpose of evaluating performance metrics. The four tasks of the homework use the `RLdata500` dataset to walk us through an exploration of the issue, the proposal of a solution, and its evaluation.

Task 1

Start by doing an exploratory analysis of the data set. What do you find?

Solution.

Table 1 shows the structure of the `RLdata500` dataset and its first few rows, when sorted by last name.

```
RLdata500 %>%
  arrange(lname_c1) %>%
  head(5) %>%
  kbl(caption = "First five rows of the \\texttt{RLdata500} dataset when sorted by last
        name.",
        booktabs = TRUE, position = "h") %>%
  add_header_above(header = c("First name" = 2, "Last name" = 2, "Birth date" = 3), bold=TRUE) %>%
  row_spec(0, monospace = TRUE)
```

First name		Last name		Birth date		
fname_c1	fname_c2	lname_c1	lname_c2	by	bm	bd
RAINER	NA	ALBRECHT	NA	1976	11	10
UWE	NA	ALBRECHT	NA	1967	1	8
ANNA	NA	ALBRECHT	NA	1964	10	22
HANNELORE	NA	ALBRECHT	NA	1963	12	4
ANNA	GUDRUN	ALBRECHT	NA	1948	10	30

Table 1: First five rows of the `RLdata500` dataset when sorted by last name.

The first and last names are each separated in two parts and birth year, month, and day are separately recorded.

Next, in Figure 1 we look at the frequency distribution of the first and last names (first components only) and of the birth date fields. Note that there are no missing values among these attributes. As for secondary name components, only 28 records have a second first name, and only 8 records have a second last name.

```
fields = c("fname_c1", "lname_c1", "by", "bm", "bd")

n_labels = 5
data = RLdata500 %>%
  select(!!!fields) %>%
  mutate(fname_c1 = fct_infreq(fname_c1),
         lname_c1 = fct_infreq(lname_c1)) %>%
  mutate_all(as.integer) %>%
  pivot_longer(everything(), names_to="Field", values_to = "Value") %>%
  add_column(labels = apply(., 1, function(x) {
    field = x[["Field"]]; value = as.integer(x[["Value"]])
    if ((field == "fname_c1") & (value <= n_labels)) {
      return(levels(fct_infreq(RLdata500$fname_c1))[[value]])
    } else if ((field == "lname_c1") & (value <= n_labels)) {
      return(levels(fct_infreq(RLdata500$lname_c1))[[value]])
    } else {
      return(NA)
    }
  }))

label_data = data %>%
  group_by(Value, Field) %>%
  summarize(label=first(labels), count = n()) %>%
  ungroup()

ggplot(data, aes(x = Value, label = labels)) +
  geom_histogram(stat = "count", fill=pretty::cmap.knitr(1)) +
  ggrepel::geom_label_repel(data = label_data,
                    mapping = aes(x=Value, y=count, label=label),
                    size = 2, label.padding=0.2, min.segment.length = 0, na.rm=TRUE,
                    seed=1) +
  xlab("") +
  cowplot::theme_minimal_hgrid(font_size = 12, font_family = "serif") +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05))) +
  theme(strip.text.x = element_text(family = "mono")) +
  facet_wrap(vars(Field), scales = "free")
```

The birth day and birth month seem roughly uniformly distributed, while birth year is more concentrated aroun 1960. An erroneous birth year of 2062 is listed on one of the record. We can observe more duplication among last names than among first names; assuming comparable error levels, first name may therefore be more discriminative of distinct individuals.

Finally, we visualize the differences between duplicated records using the `visdat` package. Recall that `RLdata500` contains 50 duplicated records, each with a corresponding original. Figure 2 illustrates the differences between original and duplicated records.

```
# Duplicated records
dup_records = which(duplicated(identity.RLdata500))

# Original records
dup_IDs = identity.RLdata500[dup_records]
original_IDs = sapply(dup_IDs, function(i) {
```

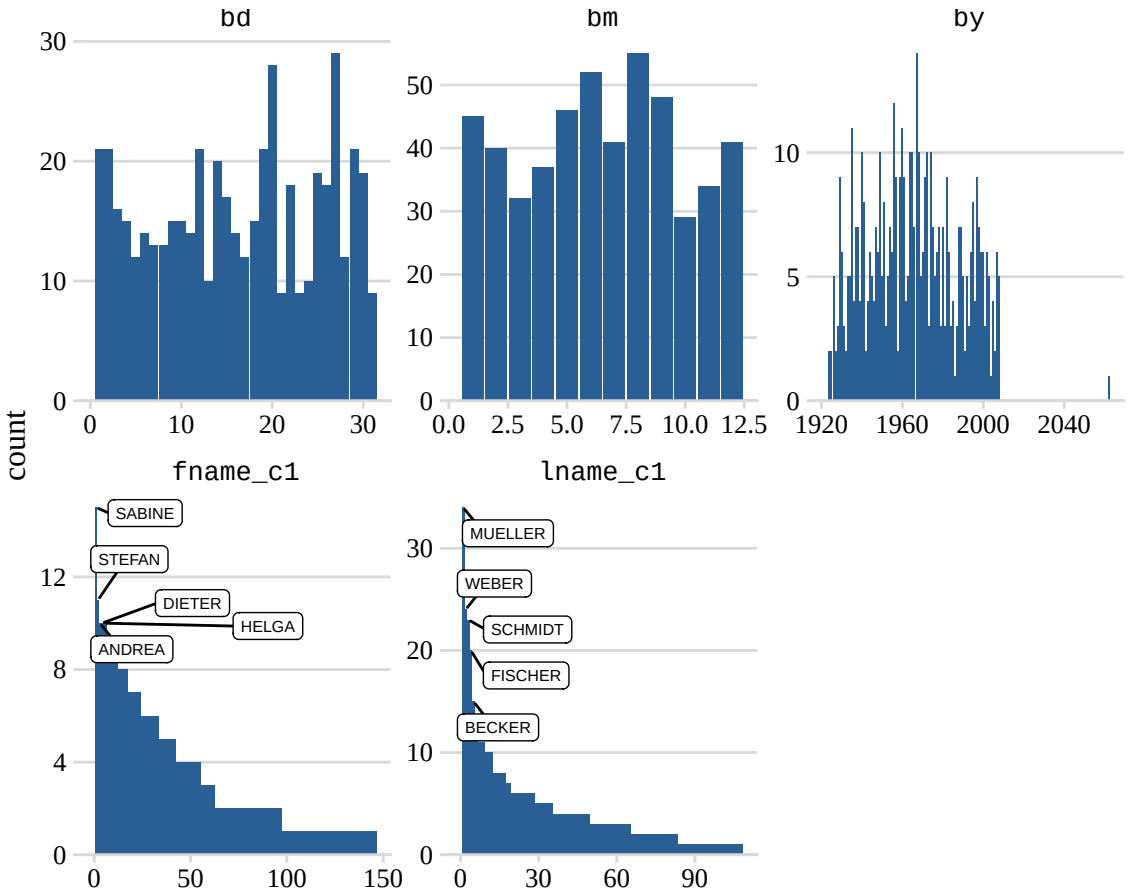

Figure 1: Frequency distribution of main record attributes. Note that first and last names have been reordered by frequency and the x-axis corresponds to unique name index.

```
  which(identity.RLdata500 == i)[[1]]
})

dfA = RLdata500[original_IDs, ]
dfB = RLdata500[dup_records, ]

vis_compare(dfA, dfB) +
  scale_fill_manual(limits = c("same", "different"),
                    breaks = c("same", "different"),
                    values = adjustcolor(cmap.knitr(c(1,2)), alpha.f = 0.9),
                    na.value = "grey") +
  labs(y="Duplicated Records", fill="Comparison") +
  theme(axis.text.x = element_text(family = "mono"))
```

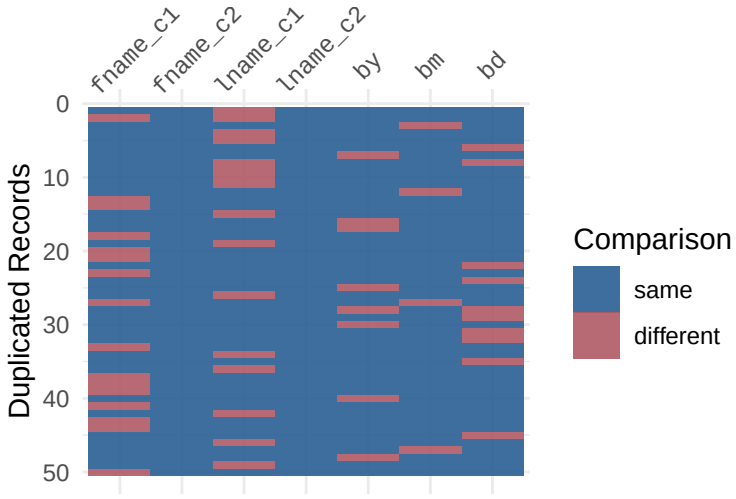

Figure 2: Visualization of the differences between the 50 original records that have been duplicated and perturbed in the `RLdata500` dataset. Each row represent one of the duplicated record. Each column indicates whether the duplicated record matches its original version in the given field.

Task 2

What happens if you randomly sample 10 records from the original dataset? Do this a few times and describe what happens? Is this representative of the original dataset? Explain and be specific.

Solution.

Let's first sample 10 records from the original dataset and take a look at the result in Table 2.

```
RLdata500 %>%
  add_column(ID = identity.RLdata500) %>%
  arrange(rnorm(1:nrow(.))) %>%
  head(10) %>%
  kbl(caption = "Ten random rows from the \\texttt{RLdata500} dataset with unique identifiers.",
        booktabs = TRUE, position = "h") %>%
  add_header_above(header = c("First name" = 2, "Last name" = 2, "Birth date" = 3, " " = 1),
                   bold=TRUE) %>%
  row_spec(0, monospace = TRUE)
```

First name		Last name		Birth date			
fname_c1	fname_c2	lname_c1	lname_c2	by	bm	bd	ID
KATHRIN	NA	KOCH	NA	2005	12	1	259
ROLF	NA	NEUMANN	NA	1967	3	29	68
BRIGITTE	EDITH	PETERS	NA	1956	8	21	389
INGRID	NA	KOERTIG	NA	1960	5	6	430
ANJA	URSULA	WEBER	NA	1995	6	26	415
MONIKA	NA	WEISS	NA	1994	8	19	369
GUENTHER	FRIEDRICH	WOLF	NA	1975	4	23	116
SUSANNE	NA	WEBER	NA	1997	11	25	265
DANIEL	NA	SCHMIDT	NA	1978	3	4	133
SABINE	NA	GRAF	NA	1980	9	5	295

Table 2: Ten random rows from the `RLdata500` dataset with unique identifiers.

In comparison to the full dataset, there is no duplicated record in this sample. Furthermore, there is no duplicate first name, no duplicate last name, no duplicate birth year, and no duplicate birth day. This particular sample therefore provides little to no useful information regarding the level of duplication in the data or regarding the distribution of the attributes.

Now supposed we wished to estimate the precentage of duplicate records in the whole dataset using such samples. Would the percentage of duplication in random samples be representative of duplicate in the whole? Figure 3 shows the distribution of the duplication level in 100,000 random samples of size 10 and compares it to the level of duplication in the whole dataset (10%).

```
k = 10
duplicate_levels = replicate(n=100000, expr={
  I = sample(1:nrow(RLdata500), k)
  sum(duplicated(identity.RLdata500[I]))/k
})

par(mar=c(3,3,1,1))
hist(duplicate_levels, xlab="Duplication level", alpha=1)
```

The mean level of duplication in the samples is only 0.001778, far from the target 10%.

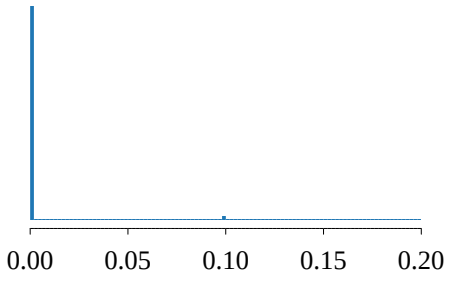

Figure 3: Histogram of duplication levels in 100,000 random samples of size 10 from the `RLdata500` dataset.

The naive duplication estimator, taking the observed mean duplication on a sample, is highly biased here. To see why this is the case, consider the coreference matrix C, defined as $C = [c_{i,j}]$ with $c_{i,j} = 1$ if records i and j match, and $c_{i,j} = 0$ otherwise. Consider the worst case scenario of sampling only two records from the whole dataset and checking if they match. This corresponds to sample an entry of C at random within its lower triangular section and dividing by two to obtain the duplication level. The expectation of this estimator is $2\ell/(n-1)$ where $n = 500$ is the size of the original dataset and $\ell = 10\%$ is the level of duplication.

More generally, if we sample k records, this corresponds to sampling $k(k-1)/2$ entries in the lower triangular section of C. The expected number of duplicates in this section is then around $\ell k(k-1)/(n-1)$. While we can adjust for the factor of $k(k-1)/(n-1)$ to obtain an unbiased estimator, the result would be highly inefficient.

We would face similar problems if trying to compute precision and recall of a proposed ER method on a subset of the data. An ER method which does not match anything would perform quite well on subsets of the data in terms of both precision and recall. However, its recall would be close to zero on the whole dataset.

There is therefore a need to both:

1. account for the unrepresentativeness of record samples in ER applications (such as by using the above adjustment factors to obtain unbiased estimators), and
2. propose ways to obtain more representative samples (as to improve the efficiency of estimators).

Task 3 deals with points (1) and (2).

Task 3

Propose something that works better than random sampling and explain why this works better.

Solution.