eAppendix for “Number of Healthcare Workers Who Have Died of Covid-19” by Bo Zhang and Dylan Small

Details on Name Matching:

We used the R package *rvest* to scrape the names of healthcare workers who have died from fighting Covid-19 off two webpages: <https://covid-heroes.com/> and (ii) <https://www.medscape.com/viewarticle/927976> . Capture-recapture method requires identifying the number of names appearing on both lists. To identify the same names, we first identified the first name and last name of each full name, calculated a similarity score (between 0 and 1) based on the Levenshtein edit distance separately for the first name and the last name, and assign a final similarity score by adding up two scores. Sometimes, the first name and last name on one website may be switched compared to the other (e.g., for Chinese names) and our algorithm took care of this issue by also calculating a version of the final score based on inverting the roles of first name and last name. Therefore, when comparing the similarity between two names, we had two versions of the final score, and our algorithm took the higher between two. In this way, we assign one final similarity score for each pair of two names, one from website (i) and the other from website (ii). For each name in website (i), our algorithm kept the name in website (ii) with the highest similarity score. We identified a total of 638 names in website (i) for whom we either identified names in website (ii) that matched perfectly to, i.e., with a similarity score of 2, or names in website (ii) that had very high similarity score and were confirmed to be true matched upon our case-by-case inspection.

R script for estimating number of healthcare workers killed by Covid-19 project

# Counts A = Covid Heroes website : B=Medscape website

lista=c(1,1,0) # Indictator for count in list a

listb=c(1,0,1) # Indicator for count in list b

# Counts as of May 20, 2021

count.ab=620 # Number on both lists

count.a.notb=928-count.ab # Number on list A, but not B

count.b.nota=1136-count.ab # Number on list B, but not A

counts=c(count.ab,count.a.notb,count.b.nota)

# Usual capture-recapture estimate assuming independence of lists

model1=glm(counts~lista+listb,family=poisson)

missing.both.lists.estimate=exp(coef(model1)[1])

lci.missing.both.lists=exp(coef(model1)[1]-1.96\*sqrt(vcov(model1)[1,1]))

uci.missing.both.lists=exp(coef(model1)[1]+1.96\*sqrt(vcov(model1)[1,1]))

total.estimate=count.ab+count.a.notb+count.b.nota+missing.both.lists.estimate

# 95% confidence interval

lci.total=count.ab+count.a.notb+count.b.nota+lci.missing.both.lists

uci.total=count.ab+count.a.notb+count.b.nota+uci.missing.both.lists

c(lci.total,total.estimate,uci.total)

# Sensitivity Analysis

# Sensitivity parameter is odds ratio of being captured on list a given being on list b

# compared to not being on list b

sens.parameter=5

listab=c(1,0,0)

model2=glm(counts~lista+listb+offset(listab\*log(sens.parameter)),family=poisson)

missing.both.lists.estimate=exp(coef(model2)[1])

lci.missing.both.lists=exp(coef(model2)[1]-1.96\*sqrt(vcov(model2)[1,1]))

uci.missing.both.lists=exp(coef(model2)[1]+1.96\*sqrt(vcov(model2)[1,1]))

total.estimate=count.ab+count.a.notb+count.b.nota+missing.both.lists.estimate

lci.total=count.ab+count.a.notb+count.b.nota+lci.missing.both.lists

uci.total=count.ab+count.a.notb+count.b.nota+uci.missing.both.lists

c(lci.total,total.estimate,uci.total)