````
---
title: "Tables S2-S21: Full dataset"
author: "Shelley Stephan"
date: "Last compiled on `r format(Sys.time(), '%d %B, %Y')`"
output:
  html_document:
    toc: yes
    toc_float: yes
    toc_collapsed: yes
---

```{r setup, include = FALSE}

# Set global options to show code of each chunk
knitr::opts_chunk$set(echo = TRUE)

# Clear Rstudio environment
rm(list=ls())

```


```{r load.packages, warning = F, message=F, echo=F, results='hide'}

# # Load pacman package
# # pacman is useful for installing/loading packages using a single
function, p_load()
if(!require("pacman")){
               install.packages("pacman")
 }

# # Installs/loads listed packages
 pacman::p_load(
   tidyverse, # required for tidy functions, also loads ggplot2 and other
packages
   pander, # required for some tables
   DT, # required for some tables
   ncar, # required for rounding using the "round half away from zero" rule
   readxl, # required to read excel files
   ggpubr) # required to arrange multiple plots
```


```{r files, include = FALSE}

## Required files, including "file name", description, and code to read
files into R

# 1. CANJEM 6 Digit SOC Matrix
# This is the JEM data, extracted from CANJEM, with the detailed SOC groups
matrix.soc.detailed.import <- read.csv("data/
CANJEM_soc6_j10s10_20220503_160052.csv")

# 2. CANJEM 6 Digit SOC Matrix with merged agents (welding fumes and
asbestos)
````
````

```
ma.matrix.soc.detailed.import <- read.csv("data/
new.matrix.soc6.p19852005.j10s10.csv")

# 3. CANJEM 5 Digit SOC Matrix
# This is the JEM data, extracted from CANJEM, with the broad SOC groups
matrix.soc.broad.import <- read.csv("data/
CANJEM_soc5_j10s10_20220503_160358.csv")

# 4. CANJEM 5 Digit SOC Matrix with merged agents (welding fumes and
asbestos)
ma.matrix.soc.broad.import <- read.csv("data/
new.matrix.soc5.p19852005.j10s10.csv")

# 5. CANJEM 3 Digit SOC Matrix
# This is the JEM data, extracted from CANJEM, with the minor SOC groups
matrix.soc.minor.import <- read.csv("data/
CANJEM_soc3_j10s10_20220503_161404.csv")

# 6. CANJEM 3 Digit SOC Matrix with merged agents (welding fumes and
asbestos) - "new.matrix.soc3.p19852005.csv"
ma.matrix.soc.minor.import <- read.csv("data/
new.matrix.soc3.p19852005.j10s10.csv")

# 7. 2019 CPS ELF data on occupation and race/ethnicity -
"ELF_CPS_2019_Race_Ethnicity.csv"
# Worker race/ethnicity data from CPS with 2010 Census Codes
CPS.race.eth.import <- read.csv("data/ELF_CPS_2019_Race_Ethnicity.csv", sep
= ",", strip.white = TRUE)

# 8. 2019 CPS ELF data on occupation and sex - "ELF_CPS_2019_Sex.csv"
# Worker sex data from CPS with 2010 Census Codes
CPS.sex.import <- read.csv("data/ELF_CPS_2019_Sex.csv", strip.white = TRUE)

# 9. 2019 CPS ELF data on occupation and nativity -
"ELF_CPS_2019_Nativity.csv"
# Worker nativity data from CPS with 2010 Census Codes
CPS.nativity.import <- read.csv("data/ELF_CPS_2019_Nativity.csv",
strip.white = TRUE)

# 10. 2019 CPS ELF data on occupation and education status -
"ELF_CPS_2019_Education.csv"
# Worker nativity data from CPS with 2010 Census Codes
CPS.education.import <- read.csv("data/ELF_CPS_2019_Education.csv",
strip.white = TRUE)

# 9. 2019 OEWS wage data - "OEWS_2021-10-14_national_M2019_dl.xlsx"
# Wage data based on 2018 SOC codes (includes detailed, broad, and minor
codes)
OEWS.import <- readxl::read_excel("data/
OEWS_2021-10-14_national_M2019_dl.xlsx")

# 10. Crosswalk - "Crosswalk_2010 Census_2010 SOC.csv"
```

```
# This is a file made by Shelley that is modified from the file
"Crosswalk_2010 Census_to_2010 SOC.xlsx" provided by BLS. This file is
modified to contain only the 2010 Census Code, 2010 SOC, and the
description.
crosswalk.Census.SOC.2010 <- read.csv("data/Crosswalk_2010 Census_2010
SOC.csv")

# 11. Key - "key.2010.Census.2019.OEWS.csv"
# This is a file made by Shelley to relate 2010 Census codes to equivalent,
or closest equivalent 2019 OEWS codes
key.2010.Census.2019.OEWS <- read.csv("data/key.2010.Census.2019.OEWS.csv",
colClasses = "character")

# 12. CANJEM agent categories
# This is a file made by Shelley that categorizes CANJEM agents by certain
parameters (e.g., OSHA regulated, agents representing specific agents or a
group of agents)
CANJEM.categories <- read.csv("data/CANJEM.OSHA.list.csv")

# 13. CANJEM list of spelling changes
# This is a file made by Shelley that translates some of the agents to
American English.
CANJEM.spelling <- read.csv("data/CANJEM_spelling.csv")

```

```{r functions, include = FALSE}

# Create commonly used functions specific to this report

# This function will round count estimates to the nearest 1000s
Round.counts <- function(x){
  x <- Round(x, -3)
  x
}

# This function will round percents to the nearest tenth
Round.percents <- function(x){
  x <- format(Round(x, 1), nsmall=1)
  x
}

```


```{r canjem.data, include = FALSE}

##################
## CANJEM DATA ##
##################

# Clean up and prepare all of the CANJEM files for use

# CANJEM matrix with detailed SOC groups (6 digit)
```

```
matrix.soc.detailed <- matrix.soc.detailed.import %>%
  # Join the merged agents to the main matrix
  bind_rows(ma.matrix.soc.detailed.import) %>%
  # Calculate probability of high exposure
  mutate(p_high = (n.FWI.greater / ntot) * 100) %>%
  # Rename columns to match syntax - underscores are used for variables/
columns and periods are be used for objects
  rename(agent_label = agent.label,
         soc_code = soc.code,
         soc_label = soc.label) %>%
  # Select desired columns
  select(idchem, agent_label, soc_code, soc_label, p, p_high, p.R3, Dmoy.5)

# CANJEM matrix with broad SOC groups (5 digit)
matrix.soc.broad <- matrix.soc.broad.import %>%
  # Add one 0 to the 5-digit SOC code to match the broad SOC groups
  mutate(soc5.code = paste0(soc5.code, 0)) %>%
  # Join the merged agents to the main matrix
  bind_rows(ma.matrix.soc.broad.import) %>%
  # Calculate probability of high exposure
  mutate(p_high = (n.FWI.greater / ntot) * 100) %>%
  # Rename columns to match syntax - underscores are used for variables/
columns and periods are be used for objects
  rename(agent_label = agent.label,
         soc_code = soc5.code,
         soc_label = soc5.label) %>%
  # Select desired columns
  select(idchem, agent_label, soc_code, soc_label, p, p_high, p.R3, Dmoy.5)


# CANJEM matrix with minor SOC groups (3 digit)
matrix.soc.minor <- matrix.soc.minor.import %>%
  # Add three 0's to the 3-digit SOC code to match the minor SOC groups
  mutate(soc3.code = paste0(soc3.code, 0),
         soc3.code = paste0(soc3.code, 0),
         soc3.code = paste0(soc3.code, 0)) %>%
  # Join the merged agents to the main matrix
  bind_rows(ma.matrix.soc.minor.import) %>%
  # Calculate probability of high exposure
  mutate(p_high = (n.FWI.greater / ntot) * 100) %>%
  # Rename columns to match syntax - underscores are used for variables/
columns and periods are be used for objects
  rename(agent_label = agent.label,
         soc_code = soc3.code,
         soc_label = soc3.label) %>%
  # Select desired columns
  select(idchem, agent_label, soc_code, soc_label, p, p_high, p.R3, Dmoy.5)


# Add the matrices together
matrix.soc <- matrix.soc.detailed %>%
  bind_rows(matrix.soc.broad) %>%
  bind_rows(matrix.soc.minor) %>%
  # Add the CANJEM categories to the matrix
```

```
  left_join(CANJEM.categories, by = "idchem") %>%
  # Filter out agents
  filter(!agent_label %in% c(
    # Irrelevant to time period
    "Coal gas", "Cutting fluids pre-1955", "Mineral spirits pre 1970",
"DDT",
    # Repetitive agents
    "PAHs from coal", "PAHs from wood", "PAHs from other sources", "PAHs
from petroleum",
    # Merged agents
    "Amphibole asbestos", "Chrysotile asbestos", "Gas welding fumes", "Arc
welding fumes")) %>%
  # Correct spelling of agents
  # Merge CANJEM.spelling to matrix.soc
  left_join(CANJEM.spelling, by = "idchem") %>%
  # Replace agent_label with label_correct where necessary
  mutate(agent_label = ifelse(!is.na(label_corrected), label_corrected,
agent_label)) %>%
  # Remove label_corrected column
  select(-label_corrected)

# Remove objects that are no longer needed
rm(matrix.soc.detailed.import, ma.matrix.soc.detailed.import,
matrix.soc.broad.import, ma.matrix.soc.broad.import,
matrix.soc.minor.import, ma.matrix.soc.minor.import, matrix.soc.detailed,
matrix.soc.broad, matrix.soc.minor, CANJEM.categories)
```

```{r OEWS.data, warning = F, message = F, echo = F}

###############
## OEWS DATA ##
###############

## Tidy OEWS Data

OEWS <- OEWS.import %>%
  # Select relevant columns
  select(occ_code, occ_title, tot_emp, a_median, h_median) %>%
  mutate(
    # Change "*" to NA
    h_median = ifelse(h_median == "*" | h_median == "#", NA, h_median),
    a_median = ifelse(a_median == "*" | a_median == "#", NA, a_median),
    # Change estimates from character string to numeric values
    h_median = as.numeric(h_median),
    a_median = as.numeric(a_median),
    # Calculate median hourly wage for occupations that only have an annual
median wage
    h_median = ifelse(is.na(h_median), a_median/2080, h_median)) %>%
  # Rename column
  rename(OEWS_tot_emp = tot_emp) %>%
  # Remove duplicates
  unique()
```

```
# Remove objects that are no longer needed
rm(OEWS.import)
```

```{r CPS.data, warning = F, message = F, echo = F}

###############
## CPS DATA ##
###############

## Create function to tidy CPS import data
fcn.CPS.import <- function(dat){
  dat <- dat %>%
    # Separate the first column into Census Code and Description
    separate(col = Occupation.Code.Primary.Job.2011..,
             into = c("Cen_2010", "Description"),
             sep = c(": ")) %>%
    # Get rid of the total columns
    filter(!is.na(Description)) %>%
    # Change columns containing estimates from character strings to numeric
values
    # Must remove "," before changing to numeric values
    mutate(across(!contains(c("Cen_2010", "Description", "Ethnicity")),
                  ~ as.numeric(gsub(",", "", .x))))

}

## Tidy CPS data

CPS.race.eth <- CPS.race.eth.import %>%
  fcn.CPS.import() %>%
  # Combine, reorganize, and rename estimate columns
  mutate("Multiracial" = White.Black + White.American.Indian + White.Asian
+ White.Hawaiian.Pacific.Islander + Black.American.Indian + Black..Asian +
Black.Hawaiian.Pacific.Islander + American.Indian.Asian +
Asian.Hawaiian.Pacific.Islander + White.Black.Amercian.Indian +
White.Black.Asian + White.American.Indian.Asian +
White.Asian.Hawaiian.Pacific.Islander + White.Black.American.Indian.Asian +
X2.or.3.races + X4.or.5.races) %>%
  select(Cen_2010, Description, Ethnicity, contains("only"), Multiracial)
%>%
  rename(White = White.only,
         Black = Black.only,
         AIAN = American.Indian..Alaskan.native.only,
         Asian = Asian.only,
         NHPI = Hawaiian.Pacific.Islander.only) %>%
  pivot_longer(cols = c(White, Black, AIAN, Asian, NHPI, Multiracial),
               names_to = "Race",
               values_to = "Count")

CPS.sex <- CPS.sex.import %>%
  fcn.CPS.import()
```

```
CPS.education <- CPS.education.import %>%
  fcn.CPS.import()


CPS.nativity <- CPS.nativity.import %>%
  fcn.CPS.import()

## Create final race/ethnicity categories
# Employees that identify as Black, AIAN, Asian, NHPI, or 2 or More Races
can be of any ethnicity. Employees that identify as Hispanic can be of any
race. Employees that identify as Non-Hispanic and White fall under the
White category. Note that this results in non-mutually exclusive race and
ethnicity categories.
  # "White": race = "White Alone" and ethnicity = "Not Hispanic or Latino"
  # "Black": race = "Black or African American Alone" and ethnicity =
"Hispanic or Latino"
  # "AIAN": race = "American Indian or Alaska Native Alone" and ethnicity =
"Hispanic or Latino"
  # "Asian": race = "Asian alone" and ethnicity = "Hispanic or Latino"
  # "NHPI": race = "Native Hawaiian or Other Pacific Islander Alone" and
ethnicity = "Hispanic or Latino"
  # "Hispanic": race = "All Races" and ethnicity = "Hispanic or Latino"
  # "Multiracial": race = "Multiracial" and ethnicity = "Hispanic or
Latino"

# Race counts
race <- CPS.race.eth %>%
  mutate(race_eth = case_when(Race == "White" & Ethnicity == "Non-Hispanic"
~ "NH_White",
                              Race == "White" & Ethnicity == "Hispanic" ~
"H_White",
                              Race == "Black" ~ "Black",
                              Race == "AIAN" ~ "AIAN",
                              Race == "Asian" ~ "Asian" ,
                              Race == "NHPI" ~ "NHPI" ,
                              Race == "Multiracial" ~ "Multiracial",
                              TRUE ~ "NA")) %>%
  group_by(Cen_2010, Description, race_eth) %>%
  summarise(Count = sum(Count)) %>%
  ungroup() %>%
  pivot_wider(names_from = race_eth,
              values_from = Count)

# Ethnicity counts
eth <- CPS.race.eth %>%
  group_by(Cen_2010, Description, Ethnicity) %>%
  summarise(Count = sum(Count)) %>%
  ungroup() %>%
  rename(race_eth = Ethnicity) %>%
  pivot_wider(names_from = race_eth,
              values_from = Count) %>%
  rename("NH" = "Non-Hispanic")
```

```
## Create final education categories
CPS.education <- CPS.education %>%
  mutate(less_HS = Less.than.1st.grade + X1st..2nd..3rd..or.4th.grade +
X5th.or.6th.grade + X7th.or.8th.grade + X9th.grade + X10th.grade +
X11th.grade + X12th.grade..no.diploma,
         HS_diploma = High.school.grad...H.S..diploma.or.equivalent,
         some_college = Some.college..no.degree +
Assoc..degree.in.college...Occ.Voc.program +
Assoc..degree.in.college...Academic.program,
         advanced_degree = Bachelor.s.degree + Master.s.degree +
Professional.school.degree + Doctorate.degree) %>%
  select(Cen_2010, Description, less_HS, HS_diploma, some_college,
advanced_degree)

## Create final nativity & citizenship status categories
CPS.nativity <- CPS.nativity %>%
  mutate(US_native = Native..Born.In.US +
Native..Born.in.PR.or.US.Outlying.Area +
Native..Born.Abroad.Of.US.Parent.s.,
         foreign_citizen = Foreign.Born..US.Cit.By.Naturalization,
         foreign_not_citizen = Foreign.Born..Not.a.US.Citizen) %>%
  select(Cen_2010, Description, US_native, foreign_citizen,
foreign_not_citizen)


## Merge all counts and finish calculating columns
CPS.data <- race %>%
  left_join(eth %>%
              select(Cen_2010, Hispanic, "NH"),
            by = "Cen_2010") %>%
  left_join(CPS.sex %>%
              select(Cen_2010, Male, Female),
            by = "Cen_2010") %>%
  left_join(CPS.education %>%
              select(Cen_2010, less_HS, HS_diploma, some_college,
advanced_degree),
            by = "Cen_2010") %>%
  left_join(CPS.nativity %>%
              select(Cen_2010, US_native, foreign_citizen,
foreign_not_citizen),
            by = "Cen_2010") %>%
  # Create BIPOC Count
  mutate(H_White = replace(H_White, is.na(H_White), 0),
         Hispanic = replace(Hispanic, is.na(Hispanic), 0),
         BIPOC = H_White + Black + AIAN + Asian + NHPI + Multiracial,
         All = NH_White + H_White + Black + AIAN + Asian + NHPI +
Multiracial) %>%
  # Calculate expected number of employees by demographic group, based on
their share of the total workforce
  mutate(across(!contains(c("Cen_2010", "Description", "All")),
            ~{sum(.) / sum(All) * All}, .names = "{.col}_expected"))

## Calculate employee totals for each demographic group
# All
```

```
All.total <- sum(CPS.data$All)
# Sex
Male.total <- sum(CPS.data$Male)
Female.total <- sum(CPS.data$Female)
# Race/eth
BIPOC.total <- sum(CPS.data$BIPOC)
AIAN.total <- sum(CPS.data$AIAN)
Asian.total <- sum(CPS.data$Asian)
Black.total <- sum(CPS.data$Black)
H.White.total <- sum(CPS.data$H_White)
Multiracial.total <- sum(CPS.data$Multiracial)
NH.White.total <- sum(CPS.data$NH_White)
NHPI.total <- sum(CPS.data$NHPI)
Hispanic.total <- sum(CPS.data$Hispanic)
NH.total <- sum(CPS.data$NH)
# Education
less.HS.total <- sum(CPS.data$less_HS)
HS.diploma.total <- sum(CPS.data$HS_diploma)
some.college.total <- sum(CPS.data$some_college)
advanced.degree.total <- sum(CPS.data$advanced_degree)
# Nativity
US.native.total <- sum(CPS.data$US_native)
foreign.citizen.total <- sum(CPS.data$foreign_citizen)
foreign.not.citizen.total <- sum(CPS.data$foreign_not_citizen)

```

```{r JEM.CPS.OEWS, warning = F, message = F, echo = F}
## Add OEWS wage information to CPS data

CPS.OEWS <- CPS.data %>%
  # Step 1: Merge CPS with 2019 OEWS key
  left_join(key.2010.Census.2019.OEWS %>%
              select(Cen_2010, OEWS_2019)) %>%
  # Step 2: Merge OEWS with CPS
  left_join(OEWS %>%
              select(occ_code, h_median, OEWS_tot_emp),
            by = c(OEWS_2019 = "occ_code")) %>%
  # Step 3: Average the median hourly wage of 2019 OEWS codes corresponding
to a single 2010 Census Code
  group_by(Cen_2010, Description, AIAN, Asian, Black, H_White, Multiracial,
NH_White, NHPI, Hispanic, NH, Male, Female, BIPOC, All, less_HS,
HS_diploma, some_college, advanced_degree, US_native, foreign_citizen,
foreign_not_citizen, AIAN_expected, Asian_expected, Black_expected,
H_White_expected, Multiracial_expected, NH_White_expected, NHPI_expected,
Hispanic_expected, NH_expected, Male_expected, Female_expected,
BIPOC_expected, less_HS_expected, HS_diploma_expected,
some_college_expected, advanced_degree_expected, US_native_expected,
foreign_citizen_expected, foreign_not_citizen_expected) %>%
  summarise(h_median = sum(OEWS_tot_emp*h_median, na.rm = T) /
sum(OEWS_tot_emp, na.rm = T)) %>%
  # Step 4: Round hourly median wage
  ungroup() %>%
  mutate(h_median = format(Round(h_median, 2), nsmall = 1))
```

```
## Create final JEM by adding CANJEM exposure information to CPS data

JEM.CPS.OEWS <- CPS.OEWS %>%
  # Step 1: Merge CPS with crosswalk
  left_join(crosswalk.Census.SOC.2010 %>%
              select(Cen_2010, SOC_2010),
            by = "Cen_2010") %>%
  # Step 2: Merge CANJEM matrix with CPS
  left_join(matrix.soc, by = c("SOC_2010" = "soc_code"))

## Remove objects that are no longer used
rm(CPS.race.eth.import, race, eth, CPS.race.eth, CPS.sex.import, CPS.sex,
CPS.nativity.import, CPS.nativity, CPS.education.import, CPS.education,
key.2010.Census.2019.OEWS, OEWS)

```

```{r coverage, warning = F, message = F, echo = F}

# Tally industries with exposure information
coverage <- JEM.CPS.OEWS %>%
  group_by(Cen_2010, Description, All) %>%
  summarise(exp_info = ifelse(sum(!is.na(p)) >= 1 , "yes", "no"))

# Occupation counts and percents
occ.counts <- coverage %>%
  group_by() %>%
  summarise(total_workforce = n_distinct(Cen_2010),
            yes_info = sum(exp_info == "yes"),
            no_info = sum(exp_info == "no")) %>%
  pivot_longer(cols = total_workforce:no_info,
               names_to = "Indicator",
               values_to = "n_occ_groups") %>%
  mutate("p_occ_groups" = n_occ_groups / n_distinct(coverage$Cen_2010) *
100)

# Employee counts and percents
emp.counts <- coverage %>%
  group_by() %>%
  summarise(total_workforce = sum(All, na.rm = T),
            yes_info = sum(All[exp_info == "yes"], na.rm = T),
            no_info = sum(All[exp_info== "no"], na.rm = T)) %>%
  pivot_longer(cols = total_workforce:no_info,
               names_to = "Indicator",
               values_to = "n_emp") %>%
  mutate("p_emp" = n_emp / sum(coverage$All, na.rm = T) * 100)

# Create a table that contains occupations and employee exposure
information
table.coverage <- occ.counts %>%
  # Merge the three tables
  left_join(emp.counts, by = "Indicator") %>%
  # Round percents to nearest tenth
```

```
  mutate(across(contains("p_"),
           Round.percents)) %>%
  # # Round count estimates (if <100: nearest 10s; if >1000 nearest 100s)
  # mutate_at(vars(contains("n_emp")),
  #           funs(ifelse(. > 1000,
  #                       Round(., -2),
  #                       Round(., -1)))) %>%
  # Paste % signs
    mutate(across(contains("p_"),
           ~ paste0(., "%"))) %>%
  # Replace values in "Indicator" column
  mutate(Indicator = replace(Indicator, Indicator == "total_workforce",
"Total workforce"),
         Indicator = replace(Indicator, Indicator == "yes_info", "Have
exposure information"),
         Indicator = replace(Indicator, Indicator == "no_info", "No
exposure information")) %>%
  # Rename columns
  rename("# of Occupation Groups" = n_occ_groups,
         "% Occupations Groups" = p_occ_groups,
         "# of Employees" = n_emp,
         "% Employees" = p_emp)

# Display table
# pander(table.coverage, justify = "left", split.table = Inf, caption =
"Table X. Summary of occupations in the United States and employees within
these occupations with exposure information.")

#Remove objects that are no longer needed
rm(occ.counts, emp.counts)

```


```{r agent.occ.matrix, include = FALSE}

## Agent + Occupation Matrix

# Create a list of all demographic groups
dem.list <- c("All", "AIAN", "Asian", "Black", "Multiracial", "NHPI",
"NH_White", "H_White", "NH","Hispanic", "BIPOC", "Female",  "Male",
"less_HS", "HS_diploma", "some_college", "advanced_degree", "US_native",
"foreign_citizen", "foreign_not_citizen")

# Agent + Occupation Matrix Estimated Exposures
agent.occ.matrix <- JEM.CPS.OEWS %>%
  # Calculate # of employees exposed and expected to be exposed
  # Calculate # of employees highly exposed and expected to be highly
exposed
  mutate(across(contains(c(dem.list)),
                list(exp_occ = ~{p / 100 * .},
                     h_exp_occ = ~{p_high / 100 * .}),
                .names = "{.col}_{.fn}"))
```

```

```{r agent.matrix, include = FALSE}

# Table with employees exposed by agent
agent.matrix <- agent.occ.matrix %>%
  group_by(idchem, agent_label, OSHA_reg, Type) %>%
  # Calculate the sum of employees exposed across all occupations for each
agent
  summarise(across(contains(c(dem.list)),
                   ~sum(., na.rm = T),
                   .names = "{.col}_all")) %>%
  # Remove unnecessary columns
  select(!contains("expected_all"), -All_all, -Male_all, -Female_all, -
BIPOC_all, -AIAN_all, -Asian_all, -Black_all, -H_White_all, -
Multiracial_all, -NH_White_all, -NHPI_all, -Hispanic_all, -NH_all, -
less_HS_all, -HS_diploma_all, -some_college_all, -advanced_degree_all, -
US_native_all, -foreign_citizen_all, -foreign_not_citizen_all) %>%
  ungroup() %>%
  #Filter out N/A value
  filter(!is.na(agent_label)) %>%
  # Calculate the percent of employees exposed
  mutate("All_exp_perc" = All_exp_occ_all / All.total * 100,
         "Male_exp_perc" = Male_exp_occ_all / Male.total * 100,
         "Female_exp_perc" = Female_exp_occ_all / Female.total * 100,
         "BIPOC_exp_perc" = BIPOC_exp_occ_all / BIPOC.total * 100,
         "AIAN_exp_perc" = AIAN_exp_occ_all / AIAN.total * 100,
         "Asian_exp_perc" = Asian_exp_occ_all / Asian.total * 100,
         "Black_exp_perc" = Black_exp_occ_all / Black.total * 100,
         "NHPI_exp_perc" = NHPI_exp_occ_all / NHPI.total * 100,
         "H_White_exp_perc" = H_White_exp_occ_all / H.White.total * 100,
         "NH_White_exp_perc" = NH_White_exp_occ_all / NH.White.total * 100,
         "Multiracial_exp_perc" = Multiracial_exp_occ_all /
Multiracial.total * 100,
         "Hispanic_exp_perc" = Hispanic_exp_occ_all / Hispanic.total * 100,
         "NH_exp_perc" = NH_exp_occ_all / NH.total * 100,
         "less_HS_exp_perc" = less_HS_exp_occ_all / less.HS.total * 100,
         "HS_diploma_exp_perc" = HS_diploma_exp_occ_all / HS.diploma.total
* 100,
         "some_college_exp_perc" = some_college_exp_occ_all /
some.college.total * 100,
         "advanced_degree_exp_perc" = advanced_degree_exp_occ_all /
advanced.degree.total * 100,
         "US_native_exp_perc" = US_native_exp_occ_all / US.native.total *
100,
         "foreign_citizen_exp_perc" = foreign_citizen_exp_occ_all /
foreign.citizen.total * 100,
         "foreign_not_citizen_exp_perc" = foreign_not_citizen_exp_occ_all /
foreign.not.citizen.total * 100
         ) %>%
  # Calculate the difference between the estimated number of employees
exposed and the expected number of employees exposed
  mutate("Male_disp_abs" = Male_exp_occ_all - Male_expected_exp_occ_all,

```
         "Female_disp_abs" = Female_exp_occ_all -
Female_expected_exp_occ_all,
         "BIPOC_disp_abs" = BIPOC_exp_occ_all- BIPOC_expected_exp_occ_all,
         "AIAN_disp_abs" = AIAN_exp_occ_all- AIAN_expected_exp_occ_all,
         "Asian_disp_abs" = Asian_exp_occ_all- Asian_expected_exp_occ_all,
         "Black_disp_abs" = Black_exp_occ_all - Black_expected_exp_occ_all,
         "NHPI_disp_abs" = NHPI_exp_occ_all- NHPI_expected_exp_occ_all,
         "H_White_disp_abs" = H_White_exp_occ_all-
H_White_expected_exp_occ_all,
         "NH_White_disp_abs" = NH_White_exp_occ_all-
NH_White_expected_exp_occ_all,
         "Multiracial_disp_abs" = Multiracial_exp_occ_all-
Multiracial_expected_exp_occ_all,
         "Hispanic_disp_abs" = Hispanic_exp_occ_all-
Hispanic_expected_exp_occ_all,
         "NH_disp_abs" = NH_exp_occ_all - NH_expected_exp_occ_all,
         "less_HS_disp_abs" = less_HS_exp_occ_all -
less_HS_expected_exp_occ_all,
         "HS_diploma_disp_abs" = HS_diploma_exp_occ_all -
HS_diploma_expected_exp_occ_all,
         "some_college_disp_abs" = some_college_exp_occ_all -
some_college_expected_exp_occ_all,
         "advanced_degree_disp_abs" = advanced_degree_exp_occ_all -
advanced_degree_expected_exp_occ_all,
         "US_native_disp_abs" = US_native_exp_occ_all -
US_native_expected_exp_occ_all,
         "foreign_citizen_disp_abs" = foreign_citizen_exp_occ_all -
foreign_citizen_expected_exp_occ_all,
         "foreign_not_citizen_disp_abs" = foreign_not_citizen_exp_occ_all -
foreign_not_citizen_expected_exp_occ_all) %>%
  # Calculate the percent difference between the estimated number of
employees exposed and the expected number of employees exposed
  mutate("Male_disp_rel" = Male_disp_abs / Male_expected_exp_occ_all * 100,
         "Female_disp_rel" = Female_disp_abs / Female_expected_exp_occ_all
* 100,
         "BIPOC_disp_rel" = BIPOC_disp_abs / BIPOC_expected_exp_occ_all *
100,
         "AIAN_disp_rel" = AIAN_disp_abs / AIAN_expected_exp_occ_all * 100,
         "Asian_disp_rel" = Asian_disp_abs / Asian_expected_exp_occ_all *
100,
         "Black_disp_rel" = Black_disp_abs / Black_expected_exp_occ_all *
100,
         "NHPI_disp_rel" = NHPI_disp_abs / NHPI_expected_exp_occ_all * 100,
         "H_White_disp_rel" = H_White_disp_abs /
H_White_expected_exp_occ_all * 100,
         "NH_White_disp_rel" = NH_White_disp_abs /
NH_White_expected_exp_occ_all * 100,
         "Multiracial_disp_rel" = Multiracial_disp_abs /
Multiracial_expected_exp_occ_all * 100,
         "Hispanic_disp_rel" = Hispanic_disp_abs /
Hispanic_expected_exp_occ_all * 100,
         "NH_disp_rel" = NH_disp_abs / NH_expected_exp_occ_all * 100,
         "less_HS_disp_rel" = less_HS_disp_abs /
less_HS_expected_exp_occ_all * 100,
```

```
         "HS_diploma_disp_rel" = HS_diploma_disp_abs /
HS_diploma_expected_exp_occ_all * 100,
         "some_college_disp_rel" = some_college_disp_abs /
some_college_expected_exp_occ_all * 100,
         "advanced_degree_disp_rel" = advanced_degree_disp_abs /
advanced_degree_expected_exp_occ_all * 100,
         "US_native_disp_rel" = US_native_disp_abs /
US_native_expected_exp_occ_all * 100,
         "foreign_citizen_disp_rel" = foreign_citizen_disp_abs /
foreign_citizen_expected_exp_occ_all * 100,
         "foreign_not_citizen_disp_rel" = foreign_not_citizen_disp_abs /
foreign_not_citizen_expected_exp_occ_all * 100) %>%
  # Calculate the percent of employees highly exposed
  mutate("All_h_exp_perc" = All_h_exp_occ_all / All.total * 100,
         "Male_h_exp_perc" = Male_h_exp_occ_all / Male.total * 100,
         "Female_h_exp_perc" = Female_h_exp_occ_all / Female.total * 100,
         "BIPOC_h_exp_perc" = BIPOC_h_exp_occ_all / BIPOC.total * 100,
         "AIAN_h_exp_perc" = AIAN_h_exp_occ_all / AIAN.total * 100,
         "Asian_h_exp_perc" = Asian_h_exp_occ_all / Asian.total * 100,
         "Black_h_exp_perc" = Black_h_exp_occ_all / Black.total * 100,
         "NHPI_h_exp_perc" = NHPI_h_exp_occ_all / NHPI.total * 100,
         "H_White_h_exp_perc" = H_White_h_exp_occ_all / H.White.total *
100,
         "NH_White_h_exp_perc" = NH_White_h_exp_occ_all / NH.White.total *
100,
         "Multiracial_h_exp_perc" = Multiracial_h_exp_occ_all /
Multiracial.total * 100,
         "Hispanic_h_exp_perc" = Hispanic_h_exp_occ_all / Hispanic.total *
100,
         "NH_h_exp_perc" = NH_h_exp_occ_all / NH.total * 100,
         "less_HS_h_exp_perc" = less_HS_h_exp_occ_all / less.HS.total *
100,
         "HS_diploma_h_exp_perc" = HS_diploma_h_exp_occ_all /
HS.diploma.total * 100,
         "some_college_h_exp_perc" = some_college_h_exp_occ_all /
some.college.total * 100,
         "advanced_degree_h_exp_perc" = advanced_degree_h_exp_occ_all /
advanced.degree.total * 100,
         "US_native_h_exp_perc" = US_native_h_exp_occ_all / US.native.total
* 100,
         "foreign_citizen_h_exp_perc" = foreign_citizen_h_exp_occ_all /
foreign.citizen.total * 100,
         "foreign_not_citizen_h_exp_perc" =
foreign_not_citizen_h_exp_occ_all / foreign.not.citizen.total * 100) %>%
    # Calculate the difference between the estimated number of employees
highly exposed and the expected number of employees highly exposed
  mutate("Male_h_disp_abs" = Male_h_exp_occ_all -
Male_expected_h_exp_occ_all,
         "Female_h_disp_abs" = Female_h_exp_occ_all -
Female_expected_h_exp_occ_all,
         "BIPOC_h_disp_abs" = BIPOC_h_exp_occ_all -
BIPOC_expected_h_exp_occ_all,
         "AIAN_h_disp_abs" = AIAN_h_exp_occ_all -
AIAN_expected_h_exp_occ_all,
```

```
         "Asian_h_disp_abs" = Asian_h_exp_occ_all -
Asian_expected_h_exp_occ_all,
         "Black_h_disp_abs" = Black_h_exp_occ_all -
Black_expected_h_exp_occ_all,
         "NHPI_h_disp_abs" = NHPI_h_exp_occ_all -
NHPI_expected_h_exp_occ_all,
         "H_White_h_disp_abs" = H_White_h_exp_occ_all -
H_White_expected_h_exp_occ_all,
         "NH_White_h_disp_abs" = NH_White_h_exp_occ_all -
NH_White_expected_h_exp_occ_all,
         "Multiracial_h_disp_abs" = Multiracial_h_exp_occ_all -
Multiracial_expected_h_exp_occ_all,
         "Hispanic_h_disp_abs" = Hispanic_h_exp_occ_all -
Hispanic_expected_h_exp_occ_all,
         "NH_h_disp_abs" = NH_h_exp_occ_all - NH_expected_h_exp_occ_all,
         "less_HS_h_disp_abs" = less_HS_h_exp_occ_all -
less_HS_expected_h_exp_occ_all,
         "HS_diploma_h_disp_abs" = HS_diploma_h_exp_occ_all -
HS_diploma_expected_h_exp_occ_all,
         "some_college_h_disp_abs" = some_college_h_exp_occ_all -
some_college_expected_h_exp_occ_all,
         "advanced_degree_h_disp_abs" = advanced_degree_h_exp_occ_all -
advanced_degree_expected_h_exp_occ_all,
         "US_native_h_disp_abs" = US_native_h_exp_occ_all -
US_native_expected_h_exp_occ_all,
         "foreign_citizen_h_disp_abs" = foreign_citizen_h_exp_occ_all -
foreign_citizen_expected_h_exp_occ_all,
         "foreign_not_citizen_h_disp_abs" =
foreign_not_citizen_h_exp_occ_all -
foreign_not_citizen_expected_h_exp_occ_all) %>%
    # Calculate the percent difference between the estimated number of
employees highly exposed and the expected number of employees highly
exposed
  mutate("Male_h_disp_rel" = Male_h_disp_abs / Male_expected_h_exp_occ_all
* 100,
         "Female_h_disp_rel" = Female_h_disp_abs /
Female_expected_h_exp_occ_all * 100,
         "BIPOC_h_disp_rel" = BIPOC_h_disp_abs /
BIPOC_expected_h_exp_occ_all * 100,
         "AIAN_h_disp_rel" = AIAN_h_disp_abs / AIAN_expected_h_exp_occ_all
* 100,
         "Asian_h_disp_rel" = Asian_h_disp_abs /
Asian_expected_h_exp_occ_all * 100,
         "Black_h_disp_rel" = Black_h_disp_abs /
Black_expected_h_exp_occ_all * 100,
         "NHPI_h_disp_rel" = NHPI_h_disp_abs / NHPI_expected_h_exp_occ_all
* 100,
         "H_White_h_disp_rel" = H_White_h_disp_abs /
H_White_expected_h_exp_occ_all * 100,
         "NH_White_h_disp_rel" = NH_White_h_disp_abs /
NH_White_expected_h_exp_occ_all * 100,
         "Multiracial_h_disp_rel" = Multiracial_h_disp_abs /
Multiracial_expected_h_exp_occ_all * 100,
```

```
         "Hispanic_h_disp_rel" = Hispanic_h_disp_abs /
Hispanic_expected_h_exp_occ_all * 100,
         "NH_h_disp_rel" = NH_h_disp_abs / NH_expected_h_exp_occ_all * 100,
         "less_HS_h_disp_rel" = less_HS_h_disp_abs /
less_HS_expected_h_exp_occ_all * 100,
         "HS_diploma_h_disp_rel" = HS_diploma_h_disp_abs /
HS_diploma_expected_h_exp_occ_all * 100,
         "some_college_h_disp_rel" = some_college_h_disp_abs /
some_college_expected_h_exp_occ_all * 100,
         "advanced_degree_h_disp_rel" = advanced_degree_h_disp_abs /
advanced_degree_expected_h_exp_occ_all * 100,
         "US_native_h_disp_rel" = US_native_h_disp_abs /
US_native_expected_h_exp_occ_all * 100,
         "foreign_citizen_h_disp_rel" = foreign_citizen_h_disp_abs /
foreign_citizen_expected_h_exp_occ_all * 100,
         "foreign_not_citizen_h_disp_rel" = foreign_not_citizen_h_disp_abs
/ foreign_not_citizen_expected_h_exp_occ_all * 100)

# agent.race.eth with rounded numbers
agent.matrix.rounded <- agent.matrix %>%
  # Round count estimates and percents
  mutate(across(contains(c("exp_occ_all", "abs")),
           Round.counts)) %>%
  mutate(across(contains(c("rel", "perc")),
           Round.percents))

```

```{r agents.summary, warning = F, message = F, echo = F}

# ## Determine which agents have 0 probability of exposure in CANJEM for
1985-2005
# agents.0.p <- agents.matrix %>%
#   group_by(agent_label) %>%
#   summarise(sum.p = sum(p, na.rm = T)) %>%
#   filter(sum.p == 0)
#
# ## Determine which agents have 0 probability of exposure when combined
with King County industries
# KC.agents.0.p <- canjem.bipoc %>%
#   group_by(agent_label) %>%
#   summarise(sum.p = sum(p, na.rm = T)) %>%
#   filter(sum.p == 0, !is.na(agent_label))
#

```

```{r table.agent, warning = F, message = F, echo = F}

# ## Generate a table with all and BIPOC estimates of employees exposed by
agent
```

```
# table.agent <- agent.matrix.rounded %>%
#   # Select desired columns
#   select(agent_label, OSHA_reg, Type, All_exp_occ_all, All_exp_perc,
BIPOC_exp_occ_all, BIPOC_exp_perc, BIPOC_disp_abs, BIPOC_disp_rel) %>%
#   # Arrange the data to show the agents with the highest number of BIPOC
employees over the expected
#   arrange(desc(All_exp_occ_all)) %>%
#   #Rename columns
#   rename("Agent" = agent_label,
#          "OSHA Regulated" = OSHA_reg,
#          "# Exposed" = All_exp_occ_all,
#          "% Exposed" = All_exp_perc,
#          "# BIPOC Exposed" = BIPOC_exp_occ_all,
#          "% BIPOC Exposed" = BIPOC_exp_perc,
#          "# BIPOC Over/ under-represented" = BIPOC_disp_abs,
#          "% BIPOC Over/ under-represented" = BIPOC_disp_rel)
#
# datatable(table.agent,
#           filter = "top",
#           extensions = "Buttons",
#           options = list(pageLength = 10, dom = 'Brtip',
#                          buttons = list(c('pageLength', 'copy', 'csv',
'excel', 'pdf'))),
#           caption = "Table X. Employee exposure estimates by agent.
Employee counts are rounded to the nearest 10s if <100, and to the nearest
100s if >1000.",
#           class = "display compact")

```
Estimates of exposure (number and percent of demographic group exposed)

By race and ethnicity

 Table S2. Number and percent of US workers
exposed to occupational agents in CANJEM by race and ethnicity, 2019.
Employee counts are rounded to the nearest 1000. Except for the persons who
identify as non-Hispanic White, persons of any race are of any ethnicity.
Persons of Hispanic or Latino ethnicity are of any race are also counted in
their preferred race category. BIPOC describes persons that identify as any
race/ethnicity other than non-Hispanic White. AIAN = American Indian and
Alaska Native, NHPI = Native Hawaiian and Other Pacific Islander, BIPOC =
Black, Indigenous, and People of Color

```{r table.agent.raceeth, warning = F, message = F, echo = F}

## Generate a table with estimates of employees exposed by agent for
detailed race/eth groups

table.agent.raceeth <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("exp_occ_all", "exp_perc")) & !
contains(c("_h", "expected"))) %>%
```

```
  select(agent_label, starts_with("All"), contains("AIAN"),
contains("Asian"), contains("Black"), contains("Multiracial"),
contains("NHPI"), contains("NH_White"), contains("Hispanic"),
contains("BIPOC")) %>%
# Arrange the data
  arrange(desc(All_exp_occ_all)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# All" = All_exp_occ_all,
         "# BIPOC" = BIPOC_exp_occ_all,
         "# AIAN" = AIAN_exp_occ_all,
         "# Asian" = Asian_exp_occ_all,
         "# Black" = Black_exp_occ_all,
         "# NHPI" = NHPI_exp_occ_all,
         "# White, non-Hispanic" = NH_White_exp_occ_all,
         "# Multiracial" = Multiracial_exp_occ_all,
         "# Hispanic" = Hispanic_exp_occ_all,
         "% All" = All_exp_perc,
         "% BIPOC" = BIPOC_exp_perc,
         "% AIAN" = AIAN_exp_perc,
         "% Asian" = Asian_exp_perc,
         "% Black" = Black_exp_perc,
         "% NHPI" = NHPI_exp_perc,
         "% White, non-Hispanic" = NH_White_exp_perc,
         "% Multiracial" = Multiracial_exp_perc,
         "% Hispanic" = Hispanic_exp_perc)

# Generate filterable data table
datatable(table.agent.raceeth,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")

```

By sex

 Table S3. Number and percent of US workers exposed to occupational agents in CANJEM by sex, 2019. Employee counts are rounded to the nearest 1000.

```{r table.agent.sex, warning = F, message = F, echo = F}

## Generate a table with estimates of employees exposed by agent and sex

table.agent.sex <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("exp_occ_all", "exp_perc")) & !
contains(c("_h", "expected"))) %>%
```

```
  select(agent_label, starts_with("All"), starts_with("Female"),
starts_with("Male")) %>%
# Arrange the data
  arrange(desc(All_exp_occ_all)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# Exposed" = All_exp_occ_all,
         "# Male Exposed" = Male_exp_occ_all,
         "# Female Exposed" = Female_exp_occ_all,
         "% Exposed" = All_exp_perc,
         "% Male Exposed" = Male_exp_perc,
         "% Female Exposed" = Female_exp_perc)

# Generate filterable data table
datatable(table.agent.sex,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")

```

By education status

 Table S4. Number and percent of US workers exposed to occupational agents in CANJEM by education status, 2019. Employee counts are rounded to the nearest 1000. <High school = less than high school diploma or equivalent, High school = high school diploma or equivalent, some college/associate = some college or associate degree, ≥ Bachelor’s = bachelor’s or other advanced degree

```{r table.agent.education, warning = F, message = F, echo = F}

## Generate a table with estimates of employees exposed by agent and
education status

table.agent.education <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("exp_occ_all", "exp_perc")) & !
contains(c("_h_", "expected"))) %>%
  select(agent_label, starts_with("All"), contains("less_HS"),
contains("HS_diploma"), contains("some_college"),
contains("advanced_degree")) %>%
# Arrange the data
  arrange(desc(All_exp_occ_all)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# All" = All_exp_occ_all,
         "# <High school" = less_HS_exp_occ_all,
         "# High school" = HS_diploma_exp_occ_all,
```

```
         "# Some college/associate" = some_college_exp_occ_all,
         "# ≥Bachelor's" = advanced_degree_exp_occ_all,
         "% All" = All_exp_perc,
         "% <High school" = less_HS_exp_perc,
         "% High school" = HS_diploma_exp_perc,
         "% Some college/associate" = some_college_exp_perc,
         "% ≥Bachelor's" = advanced_degree_exp_perc)

# Generate filterable data table
datatable(table.agent.education,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")

```

By nativity and citizenship status

 Table S5. Number and percent of US workers exposed to occupational agents in CANJEM by nativity and citizenship status, 2019. Employee counts are rounded to the nearest 1000.

```{r table.agent.nativity, warning = F, message = F, echo = F}

## Generate a table with estimates of employees exposed by agent and nativity and citizenship status

table.agent.nativity <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("exp_occ_all", "exp_perc")) & !
contains(c("_h_", "expected"))) %>%
  select(agent_label, starts_with("All"), contains("US_native"),
contains("foreign_citizen"), contains("foreign_not_citizen")) %>%
# Arrange the data
  arrange(desc(All_exp_occ_all)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# All" = All_exp_occ_all,
         "# Native-born" = US_native_exp_occ_all,
         "# Foreign-born, citizen" = foreign_citizen_exp_occ_all,
         "# Foreign-born, noncitizen" = foreign_not_citizen_exp_occ_all,
         "% All" = All_exp_perc,
         "% Native-born" = US_native_exp_perc,
         "% Foreign-born, citizen" = foreign_citizen_exp_perc,
         "% Foreign-born, noncitizen" = foreign_not_citizen_exp_perc)

# Generate filterable data table
datatable(table.agent.nativity,
          filter = "top",
```

```
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")

```
Estimates of exposure disproportionality (number and percent of workers over or underrepresented)

By race and ethnicity

 Table S6. Number and percent of US workers over or underrepresented in exposure to occupational agents in CANJEM by race and ethnicity, 2019. The number and percent of employees over or underrepresented represents the absolute and relative differences between the number of workers estimated to be exposed and the number of workers expected to be exposed in a counterfactual scenario in which workers of each sociodemographic group are evenly distributed across all occupations in the US based on their overall proportion of the total workforce. Employee counts are rounded to the nearest 1000. Except for the persons who identify as non-Hispanic White, persons of any race are of any ethnicity. Persons of Hispanic or Latino ethnicity are of any race are also counted in their preferred race category. BIPOC describes persons that identify as any race/ethnicity other than non-Hispanic White. AIAN = American Indian and Alaska Native, NHPI = Native Hawaiian and Other Pacific Islander, BIPOC = Black, Indigenous, and People of Color
```{r table.agent.disp.raceeth, warning = F, message = F, echo = F}

## Generate a table with estimates of workers over or under expected for each agent by detailed race/eth groups

table.agent.disp.raceeth <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("disp_abs", "disp_rel")) & !
contains("_h")) %>%
  select(agent_label,  contains("AIAN"), contains("Asian"),
contains("Black"), contains("Multiracial"), contains("NHPI"),
contains("NH_White"), contains("Hispanic"), contains("BIPOC")) %>%
# Arrange the data
  arrange(desc(BIPOC_disp_rel)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# BIPOC" = BIPOC_disp_abs,
         "# AIAN" = AIAN_disp_abs,
         "# Asian" = Asian_disp_abs,
         "# Black" = Black_disp_abs,
         "# NHPI" = NHPI_disp_abs,
         "# White, non-Hispanic" = NH_White_disp_abs,
         "# Multiracial" = Multiracial_disp_abs,
         "# Hispanic" = Hispanic_disp_abs,
         "% BIPOC" = BIPOC_disp_rel,
```

```
          "% AIAN" = AIAN_disp_rel,
          "% Asian" = Asian_disp_rel,
          "% Black" = Black_disp_rel,
          "% NHPI" = NHPI_disp_rel,
          "% White, non-Hispanic" = NH_White_disp_rel,
          "% Multiracial" = Multiracial_disp_rel,
          "% Hispanic" = Hispanic_disp_rel)

# Generate filterable data table
datatable(table.agent.disp.raceeth,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")


```

By sex

 Table S7. Number and percent of US workers over or underrepresented in exposure to occupational agents in CANJEM by sex, 2019. The number and percent of employees over or underrepresented represents the absolute and relative differences between the number of workers estimated to be exposed and the number of workers expected to be exposed in a counterfactual scenario in which workers of each sociodemographic group are evenly distributed across all occupations in the US based on their overall proportion of the total workforce. Employee counts are rounded to the nearest 1000.
```{r table.agent.disp.sex, warning = F, message = F, echo = F}

## Generate a table with estimates of workers over or under expected for each agent by detailed race/eth groups

table.agent.disp.sex <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("disp_abs", "disp_rel")) & !
contains("_h")) %>%
  select(agent_label, starts_with("Female"), starts_with("Male")) %>%
# Arrange the data
  arrange(desc(Female_disp_rel)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# Male" = Male_disp_abs,
         "# Female" = Female_disp_abs,
         "% Male" = Male_disp_rel,
         "% Female" = Female_disp_rel)

# Generate filterable data table
datatable(table.agent.disp.sex,
```

```
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")


```

By education status

 Table S8. Number and percent of US workers over or underrepresented in exposure to occupational agents in CANJEM by education status, 2019. The number and percent of employees over or underrepresented represents the absolute and relative differences between the number of workers estimated to be exposed and the number of workers expected to be exposed in a counterfactual scenario in which workers of each sociodemographic group are evenly distributed across all occupations in the US based on their overall proportion of the total workforce. Employee counts are rounded to the nearest 1000. <High school = less than high school diploma or equivalent, High school = high school diploma or equivalent, some college/associate = some college or associate degree, ≥ Bachelor's = bachelor's or other advanced degree
```{r table.agent.disp.education, warning = F, message = F, echo = F}

## Generate a table with estimates of workers over or under expected for each agent by education status

table.agent.disp.education <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("disp_abs", "disp_rel")) & !
contains("_h_")) %>%
  select(agent_label, contains("less_HS"), contains("HS_diploma"),
contains("some_college"), contains("advanced_degree")) %>%
  # Arrange the data
  arrange(desc(less_HS_disp_rel)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# <High school" = less_HS_disp_abs,
         "# High school" = HS_diploma_disp_abs,
         "# Some college/associate" = some_college_disp_abs,
         "# ≥Bachelor's" = advanced_degree_disp_abs,
         "% <High school" = less_HS_disp_rel,
         "% High school" = HS_diploma_disp_rel,
         "% Some college/associate" = some_college_disp_rel,
         "% ≥Bachelor's" = advanced_degree_disp_rel)

# Generate filterable data table
datatable(table.agent.disp.education,
          filter = "top",
          extensions = c("Buttons"),
```

```
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")


```

By nativity and citizenship status

 Table S9. Number and percent of US workers over or underrepresented in exposure to occupational agents in CANJEM by nativity and citizenship status, 2019. The number and percent of employees over or underrepresented represents the absolute and relative differences between the number of workers estimated to be exposed and the number of workers expected to be exposed in a counterfactual scenario in which workers of each sociodemographic group are evenly distributed across all occupations in the US based on their overall proportion of the total workforce. Employee counts are rounded to the nearest 1000.
```{r table.agent.disp.nativity, warning = F, message = F, echo = F}

## Generate a table with estimates of workers over or under expected for each agent by detailed race/eth groups

table.agent.disp.nativity <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("disp_abs", "disp_rel")) & !
contains("_h")) %>%
  select(agent_label, contains("US_native"), contains("foreign_citizen"),
contains("foreign_not_citizen")) %>%
# Arrange the data
  arrange(desc(foreign_not_citizen_disp_rel)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# Native-born" = US_native_disp_abs,
         "# Foreign-born, citizen" = foreign_citizen_disp_abs,
         "# Foreign-born, noncitizen" = foreign_not_citizen_disp_abs,
         "% Native-born" = US_native_disp_rel,
         "% Foreign-born, citizen" = foreign_citizen_disp_rel,
         "% Foreign-born, noncitizen" = foreign_not_citizen_disp_rel)

# Generate filterable data table
datatable(table.agent.disp.nativity,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")
```

```

```{r disp.graphs, , warning = F, message = F, echo = F}

# # Prepare tables to be used for the graph
#
# graph.disp.raceeth <- table.agent.disp.raceeth %>%
#   # Count # of agents overrepresented
#   summarise(AIAN = sum(`# AIAN` > 0),
#             Asian = sum(`# Asian` > 0),
#             Black = sum(`# Black` > 0),
#             Multiracial = sum(`# Multiracial` > 0),
#             NHPI = sum(`# NHPI` > 0),
#             "White, non-Hispanic" = sum(`# White` > 0),
#             Hispanic = sum(`# Hispanic` > 0),
#             BIPOC = sum(`# BIPOC` > 0)) %>%
#   # Pivot long
#   pivot_longer(cols = AIAN:BIPOC,
#                names_to = "Race/ethnicity",
#                values_to = "# Overrepresented") %>%
#   # Relevel factors so they display in desired order on graph
#   mutate(`Race/ethnicity` = fct_relevel(`Race/ethnicity`, "AIAN",
"Asian", "Black", "Multiracial", "NHPI", "White, non-Hispanic", "Hispanic",
"BIPOC"))
#
# graph.disp.sex <- table.agent.disp.sex %>%
#   # Count # of agents overrepresented
#   summarise(Female = sum(`# Female` > 0),
#             Male = sum(`# Male` > 0)) %>%
#   # Pivot long
#   pivot_longer(cols = Female:Male,
#                names_to = "Sex",
#                values_to = "# Overrepresented") %>%
#   # Relevel factors so they display in desired order on graph
#   mutate(Sex = fct_relevel(Sex, "Female", "Male"))
#
# graph.disp.education <- table.agent.disp.education %>%
#   # Count # of agents overrepresented
#   summarise("<High school" = sum(`# Less than high school diploma or
equivalent` > 0),
#             "High school" = sum(`# High school diploma or equivalent` >
0),
#             "Some college/associate" = sum(`# Some college or associate
degree` > 0),
#             "≥Bachelor's" = sum(`# Bachelor's or other advanced degree` >
0)) %>%
#   # Pivot long
#   pivot_longer(cols = `<High school`:`≥Bachelor's`,
#                names_to = "Education Status",
#                values_to = "# Overrepresented") %>%
#   # Relevel factors so they display in desired order on graph
#   mutate(`Education Status` = fct_relevel(`Education Status`, "<High
school", "High school", "Some college/associate", "≥Bachelor's"))
```
```

```
#
# graph.disp.nativity <- table.agent.disp.nativity %>%
#   # Count # of agents overrepresented
#   summarise("US native" = sum(`# US native` > 0),
#             "Foreign Born, US Citizen" = sum(`# Foreign Born, US Citizen`
> 0),
#             "Foreign Born, Non-US Citizen" = sum(`# Foreign Born, Non-US
Citizen` > 0)) %>%
#   # Pivot long
#   pivot_longer(cols = `US native`:`Foreign Born, Non-US Citizen`,
#                names_to = "Nativity and Citizenship Status",
#                values_to = "# Overrepresented") %>%
#   # Relevel factors so they display in desired order on graph
#   mutate(`Nativity and Citizenship Status` = fct_relevel(`Nativity and
Citizenship Status`, "US native", "Foreign Born, US Citizen", "Foreign
Born, Non-US Citizen"))
#
# # Generate plot for each sociodemographic category
#
# ggplot.disp.raceeth <- graph.disp.raceeth %>%
#   ggplot(aes(x = `Race/ethnicity`, y = `# Overrepresented`)) +
#   geom_col() +
#   geom_text(
#     aes(label = `# Overrepresented`),
#     color = "black",
#     size = 4,
#     vjust = -0.5) +
#   theme(panel.grid.major = element_blank(),
#         panel.grid.minor = element_blank(),
#         panel.background = element_blank(),
#         axis.line = element_line(colour = "black"),
#         axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size
= 12),
#         axis.text.y = element_text(size = 12),
#         axis.title = element_text(size = 12)) +
#   scale_y_continuous(limits = c(NA, 250))
#
# ggplot.disp.raceeth
#
# ggplot.disp.sex <- graph.disp.sex %>%
#   ggplot(aes(x = `Sex`, y = `# Overrepresented`)) +
#   geom_col() +
#   geom_text(
#     aes(label = `# Overrepresented`),
#     color = "black",
#     size = 4,
#     vjust = -0.5) + theme_classic()+
#   labs(x = "Sex", y = "# Agents Overrepresented") +
#   theme(panel.grid.major = element_blank(),
#         panel.grid.minor = element_blank(),
#         panel.background = element_blank(),
#         axis.line = element_line(colour = "black"),
#         axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size
= 12),
```

```
#         axis.text.y = element_text(size = 12),
#         axis.title = element_text(size = 12)) +
#   scale_y_continuous(limits = c(NA, 250))
# ggplot.disp.sex
#
# ggplot.disp.education <- graph.disp.education %>%
#   ggplot(aes(x = `Education Status`, y = `# Overrepresented`)) +
#   geom_col() +
#   geom_text(
#     aes(label = `# Overrepresented`),
#     color = "black",
#     size = 4,
#     vjust = -0.5) + theme_classic()+
#   theme(panel.grid.major = element_blank(),
#         panel.grid.minor = element_blank(),
#         panel.background = element_blank(),
#         axis.line = element_line(colour = "black"),
#         axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size
= 12),
#         axis.text.y = element_text(size = 12),
#         axis.title = element_text(size = 12)) +
#   scale_y_continuous(limits = c(NA, 250))
# ggplot.disp.education
#
# ggplot.disp.nativity <- graph.disp.nativity %>%
#   ggplot(aes(x = `Nativity and Citizenship Status`, y = `#
Overrepresented`)) +
#   geom_col() +
#   geom_text(
#     aes(label = `# Overrepresented`),
#     color = "black",
#     size = 4,
#     vjust = -0.5) + theme_classic()+
#   theme(panel.grid.major = element_blank(),
#         panel.grid.minor = element_blank(),
#         panel.background = element_blank(),
#         axis.line = element_line(colour = "black"),
#         axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size
= 12),
#         axis.text.y = element_text(size = 12),
#         axis.title = element_text(size = 12)) +
#   scale_y_continuous(limits = c(NA, 250))
# ggplot.disp.nativity

```


<!-- ### ALTERNATE TABLE - Estimated number and percent of workers exposed
to occupational agents in CANJEM by detailed race/eth groups -->

```{r table.agent.detailed2, warning = F, message = F, echo = F}

## Generate a table with estimates of employees exposed by agent for
detailed demographic categories
```

```
# table.agent.detailed2 <- agent.matrix.rounded %>%
#   # Select desired columns
#   select(agent_label, OSHA_reg, Type, All_exp_perc,
contains("exp_occ_all") & !contains(c("_h", "expected"))) %>%
#   mutate_at(vars(contains("exp_occ_all") & !All_exp_occ_all),
#             funs(perc = Round((. / All_exp_occ_all * 100), 1))) %>%
#   select(agent_label, OSHA_reg, Type, All_exp_occ_all, All_exp_perc,
contains("perc") & !H_White_exp_occ_all_perc & !NH_exp_occ_all_perc) %>%
# # Arrange the data
#   arrange(desc(All_exp_occ_all)) %>%
#   #Rename columns
#   rename("Agent" = agent_label,
#          "OSHA Regulated" = OSHA_reg,
#          "# Total Exposed" = All_exp_occ_all,
#          "% of All Employees Exposed" = All_exp_perc,
#          "% Male" = Male_exp_occ_all_perc,
#          "% Female" = Female_exp_occ_all_perc,
#          "% BIPOC" = BIPOC_exp_occ_all_perc,
#          "% AIAN" = AIAN_exp_occ_all_perc,
#          "% Asian" = Asian_exp_occ_all_perc,
#          "% Black" = Black_exp_occ_all_perc,
#          "% NHPI" = NHPI_exp_occ_all_perc,
#          "% White" = NH_White_exp_occ_all_perc,
#          "% Multiracial" = Multiracial_exp_occ_all_perc,
#          "% Hispanic" = Hispanic_exp_occ_all_perc)
#
# # Generate filterable data table
# datatable(table.agent.detailed2,
#           filter = "top",
#           extensions = c("Buttons"),
#           options = list(pageLength = 10,
#                          scrollX = TRUE,
#                          dom = 'Brtip',
#                          buttons =
list(I('colvis'),c('pageLength','copy', 'csv', 'excel', 'pdf'))),
#           class = "display compact")
```

Exposure Burden Index (EBI)
Overview
The EBI is reflective of the overall exposure burden of an occupation and is based on the average ranking for probability of exposure, frequency-weighted intensity (FWI) of exposure, and the estimated number of exposed workers.

Instructions
Filter the datatable by agent of interest to obtain occupations ranked by their overall exposure burden, as indicated by their exposure burden index (EBI). Results can be further filtered by wage and other measures of equity.

 Table S10. Occupations by EBI rank for all occupational agents in CANJEM. The EBI rank is based on the average ranking for probability of exposure, FWI, and the estimated number of exposed workers for each occupation. Employee counts are rounded to the nearest 1000. Low-wage is defined a median hourly wage of less than $15/hour, which is the minimum salary of a full-time worker needed to achieve a modest but adequate standard of living for a single adult without children in all areas across the US (Economic Policy Institute, 2021). EBI = exposure burden index, FWI = frequency-weighted intensity of exposure

```{r table.occupation.agent, warning = F, message = F, echo = F}

# Create filterable table including occupation, agent, exposure demographic information
table.agent.occ <- agent.occ.matrix %>%
  # Filter out occupations that have 0 probability of exposure
  filter(p > 0) %>%
  # Select desired columns
  select(Cen_2010, Description, agent_label, All, p, p_high, OSHA_reg, Type, contains("exp_occ"), h_median, Dmoy.5) %>%
  group_by(agent_label) %>%
  # Rank agents by p and average FWI
  mutate(p_rank = min_rank(desc(p)),
         FWI_rank = min_rank(desc(Dmoy.5)),
         All_exp_occ_rank = min_rank(desc(All_exp_occ)),
         EBI_rank = min_rank((p_rank + FWI_rank + All_exp_occ_rank)/3)) %>%
  # Calculate the % of workers in each occupation that account for the total number of exposed workers for each agent
  mutate(p_of_total = All_exp_occ / sum(All_exp_occ) * 100) %>%
  ungroup() %>%
  mutate(
    # Calculate # and % of employees in each demographic group disproportionately exposed
    Male_exp_occ_disp_abs = Male_exp_occ - Male_expected_exp_occ,
    Female_exp_occ_disp_abs = Female_exp_occ - Female_expected_exp_occ,
    BIPOC_exp_occ_disp_abs = BIPOC_exp_occ - BIPOC_expected_exp_occ,
    Hispanic_exp_occ_disp_abs = Hispanic_exp_occ - Hispanic_expected_exp_occ,
    AIAN_exp_occ_disp_abs = AIAN_exp_occ - AIAN_expected_exp_occ,
    Asian_exp_occ_disp_abs = Asian_exp_occ - Asian_expected_exp_occ,
    Black_exp_occ_disp_abs = Black_exp_occ - Black_expected_exp_occ,
    Multiracial_exp_occ_disp_abs = Multiracial_exp_occ - Multiracial_expected_exp_occ,
    NHPI_exp_occ_disp_abs = NHPI_exp_occ - NHPI_expected_exp_occ,
    NH_White_exp_occ_disp_abs = NH_White_exp_occ - NH_White_expected_exp_occ,
    less_HS_exp_occ_disp_abs = less_HS_exp_occ - less_HS_expected_exp_occ,
    HS_diploma_exp_occ_disp_abs = HS_diploma_exp_occ - HS_diploma_expected_exp_occ,
    some_college_exp_occ_disp_abs = some_college_exp_occ - some_college_expected_exp_occ,
    advanced_degree_exp_occ_disp_abs = advanced_degree_exp_occ - advanced_degree_expected_exp_occ,
```

```
    US_native_exp_occ_disp_abs = US_native_exp_occ -
US_native_expected_exp_occ,
    foreign_citizen_exp_occ_disp_abs = foreign_citizen_exp_occ -
foreign_citizen_expected_exp_occ,
    foreign_not_citizen_exp_occ_disp_abs = foreign_not_citizen_exp_occ -
foreign_not_citizen_expected_exp_occ,

    Male_exp_occ_disp_rel = Male_exp_occ_disp_abs / Male_expected_exp_occ *
100,
    Female_exp_occ_disp_rel = Female_exp_occ_disp_abs /
Female_expected_exp_occ * 100,
    BIPOC_exp_occ_disp_rel = BIPOC_exp_occ_disp_abs /
BIPOC_expected_exp_occ * 100,
    Hispanic_exp_occ_disp_rel = Hispanic_exp_occ_disp_abs /
Hispanic_expected_exp_occ * 100,
    AIAN_exp_occ_disp_rel = AIAN_exp_occ_disp_abs / AIAN_expected_exp_occ *
100,
    Asian_exp_occ_disp_rel = Asian_exp_occ_disp_abs /
Asian_expected_exp_occ * 100,
    Black_exp_occ_disp_rel = Black_exp_occ_disp_abs /
Black_expected_exp_occ * 100,
    Multiracial_exp_occ_disp_rel = Multiracial_exp_occ_disp_abs /
Multiracial_expected_exp_occ * 100,
    NHPI_exp_occ_disp_rel = NHPI_exp_occ_disp_abs / NHPI_expected_exp_occ *
100,
    NH_White_exp_occ_disp_rel = NH_White_exp_occ_disp_abs /
NH_White_expected_exp_occ * 100,
    less_HS_exp_occ_disp_rel = less_HS_exp_occ_disp_abs /
less_HS_expected_exp_occ * 100,
    HS_diploma_exp_occ_disp_rel = HS_diploma_exp_occ_disp_abs /
HS_diploma_expected_exp_occ * 100,
    some_college_exp_occ_disp_rel = some_college_exp_occ_disp_abs /
some_college_expected_exp_occ * 100,
    advanced_degree_exp_occ_disp_rel = advanced_degree_exp_occ_disp_abs /
advanced_degree_expected_exp_occ * 100,
    US_native_exp_occ_disp_rel = US_native_exp_occ_disp_abs /
US_native_expected_exp_occ * 100,
    foreign_citizen_exp_occ_disp_rel = foreign_citizen_exp_occ_disp_abs /
foreign_citizen_expected_exp_occ * 100,
    foreign_not_citizen_exp_occ_disp_rel =
foreign_not_citizen_exp_occ_disp_abs / foreign_not_citizen_expected_exp_occ
* 100,

    Male_h_exp_occ_disp_abs = Male_h_exp_occ - Male_expected_h_exp_occ,
    Female_h_exp_occ_disp_abs = Female_h_exp_occ -
Female_expected_h_exp_occ,
    BIPOC_h_exp_occ_disp_abs = BIPOC_h_exp_occ - BIPOC_expected_h_exp_occ,
    Hispanic_h_exp_occ_disp_abs = Hispanic_h_exp_occ -
Hispanic_expected_h_exp_occ,
    AIAN_h_exp_occ_disp_abs = AIAN_h_exp_occ - AIAN_expected_h_exp_occ,
    Asian_h_exp_occ_disp_abs = Asian_h_exp_occ - Asian_expected_h_exp_occ,
    Black_h_exp_occ_disp_abs = Black_h_exp_occ - Black_expected_h_exp_occ,
    Multiracial_h_exp_occ_disp_abs = Multiracial_h_exp_occ -
Multiracial_expected_h_exp_occ,
```

```
    NHPI_h_exp_occ_disp_abs = NHPI_h_exp_occ - NHPI_expected_h_exp_occ,
    NH_White_h_exp_occ_disp_abs = NH_White_h_exp_occ -
NH_White_expected_h_exp_occ,
    less_HS_h_exp_occ_disp_abs = less_HS_h_exp_occ -
less_HS_expected_h_exp_occ,
    HS_diploma_h_exp_occ_disp_abs = HS_diploma_h_exp_occ -
HS_diploma_expected_h_exp_occ,
    some_college_h_exp_occ_disp_abs = some_college_h_exp_occ -
some_college_expected_h_exp_occ,
    advanced_degree_h_exp_occ_disp_abs = advanced_degree_h_exp_occ -
advanced_degree_expected_h_exp_occ,
    US_native_h_exp_occ_disp_abs = US_native_h_exp_occ -
US_native_expected_h_exp_occ,
    foreign_citizen_h_exp_occ_disp_abs = foreign_citizen_h_exp_occ -
foreign_citizen_expected_h_exp_occ,
    foreign_not_citizen_h_exp_occ_disp_abs = foreign_not_citizen_h_exp_occ
- foreign_not_citizen_expected_h_exp_occ,

    Male_h_exp_occ_disp_rel = Male_h_exp_occ_disp_abs /
Male_expected_h_exp_occ * 100,
    Female_h_exp_occ_disp_rel = Female_h_exp_occ_disp_abs /
Female_expected_h_exp_occ * 100,
    BIPOC_h_exp_occ_disp_rel = BIPOC_h_exp_occ_disp_abs /
BIPOC_expected_h_exp_occ * 100,
    Hispanic_h_exp_occ_disp_rel = Hispanic_h_exp_occ_disp_abs /
Hispanic_expected_h_exp_occ * 100,
    AIAN_h_exp_occ_disp_rel = AIAN_h_exp_occ_disp_abs /
AIAN_expected_h_exp_occ * 100,
    Asian_h_exp_occ_disp_rel = Asian_h_exp_occ_disp_abs /
Asian_expected_h_exp_occ * 100,
    Black_h_exp_occ_disp_rel = Black_h_exp_occ_disp_abs /
Black_expected_h_exp_occ * 100,
    Multiracial_h_exp_occ_disp_rel = Multiracial_h_exp_occ_disp_abs /
Multiracial_expected_h_exp_occ * 100,
    NHPI_h_exp_occ_disp_rel = NHPI_h_exp_occ_disp_abs /
NHPI_expected_h_exp_occ * 100,
    NH_White_h_exp_occ_disp_rel = NH_White_h_exp_occ_disp_abs /
NH_White_expected_h_exp_occ * 100,
    less_HS_h_exp_occ_disp_rel = less_HS_h_exp_occ_disp_abs /
less_HS_expected_h_exp_occ * 100,
    HS_diploma_h_exp_occ_disp_rel = HS_diploma_h_exp_occ_disp_abs /
HS_diploma_expected_h_exp_occ * 100,
    some_college_h_exp_occ_disp_rel = some_college_h_exp_occ_disp_abs /
some_college_expected_h_exp_occ * 100,
    advanced_degree_h_exp_occ_disp_rel = advanced_degree_h_exp_occ_disp_abs
/ advanced_degree_expected_h_exp_occ * 100,
    US_native_h_exp_occ_disp_rel = US_native_h_exp_occ_disp_abs /
US_native_expected_h_exp_occ * 100,
    foreign_citizen_h_exp_occ_disp_rel = foreign_citizen_h_exp_occ_disp_abs
/ foreign_citizen_expected_h_exp_occ * 100,
    foreign_not_citizen_h_exp_occ_disp_rel =
foreign_not_citizen_h_exp_occ_disp_abs /
foreign_not_citizen_expected_h_exp_occ * 100,
```

```
    # Combine occ code and label
    Cen_2010 = paste0(Cen_2010, " - ", Description),

    # Flag low-wage occ
    Low_wage = ifelse(h_median < 15, "Yes", "No"),

    # Flag race/eth groups disproportionately exposed
    BIPOC_disp_flag = ifelse(BIPOC_exp_occ_disp_abs > 0, "Yes", "No"),
    Hispanic_disp_flag = ifelse(Hispanic_exp_occ_disp_abs > 0, "Yes",
"No"),
    AIAN_disp_flag = ifelse(AIAN_exp_occ_disp_abs > 0, "Yes", "No"),
    Asian_disp_flag = ifelse(Asian_exp_occ_disp_abs > 0, "Yes", "No"),
    Black_disp_flag = ifelse(Black_exp_occ_disp_abs > 0, "Yes", "No"),
    Multiracial_disp_flag = ifelse(Multiracial_exp_occ_disp_abs > 0, "Yes",
"No"),
    NHPI_disp_flag = ifelse(NHPI_exp_occ_disp_abs > 0, "Yes", "No"),

    # Flag female disproportionately exposed
    Female_disp_flag = ifelse(Female_exp_occ_disp_abs > 0, "Yes", "No"),

    # Flag less than high school diploma disproportionately exposed
    less_HS_disp_flag = ifelse(less_HS_exp_occ_disp_abs > 0, "Yes", "No"),

    # Flag foreign born, non-US citizen disproportionately exposed
    foreign_not_citizen_disp_flag =
ifelse(foreign_not_citizen_exp_occ_disp_abs > 0, "Yes", "No")) %>%

    # Round count estimates and percents
  mutate(across(c(All, contains("exp_occ") & !contains("_rel") & !
contains("rank")),
            Round.counts)) %>%
  mutate(across(c(p, p_of_total),
            Round.percents)) %>%

  # Sort by greatest # of employees exposed
  arrange(EBI_rank) %>%

  # Select and rename desired columns
  select(agent_label, Cen_2010, All, p, Dmoy.5, All_exp_occ, p_of_total,
p_rank, FWI_rank, All_exp_occ_rank, EBI_rank, h_median, Low_wage,
BIPOC_disp_flag, Hispanic_disp_flag, AIAN_disp_flag, Asian_disp_flag,
Black_disp_flag, Multiracial_disp_flag, NHPI_disp_flag, Female_disp_flag,
less_HS_disp_flag, foreign_not_citizen_disp_flag) %>%

  rename("Agent" = agent_label,
         "Occupation" = Cen_2010,
         "# of workers in occupation" = All,
         "Probability of exposure" = p,
         "# exposed" = All_exp_occ,
         "FWI" = Dmoy.5,
         "% of total exposed" = p_of_total,
         "Probability rank" = p_rank,
         "FWI rank" = FWI_rank,
         "Worker estimate rank" = All_exp_occ_rank,
```

```
         "Overall EBI rank" = EBI_rank,
         "Median hourly wage ($)" = h_median,
         "Low-wage" = Low_wage,
         "BIPOC overrepresented" = BIPOC_disp_flag,
         "Hispanic overrepresented" = Hispanic_disp_flag,
         "AIAN overrepresented" = AIAN_disp_flag,
         "Asian overrepresented" = Asian_disp_flag,
         "Black overrepresented" = Black_disp_flag,
         "Multiracial overrepresented" = Multiracial_disp_flag,
         "NHPI overrepresented" = NHPI_disp_flag,
         "Female overrepresented" = Female_disp_flag,
         "<High school overrepresented" = less_HS_disp_flag,
         "Foreign-born, noncitizen overrepresented" =
foreign_not_citizen_disp_flag)

# Show using data table
datatable(table.agent.occ,
           filter = "top",
           extensions = c("Buttons", "Scroller"),
           options = list(pageLength = 10,
                          scrollX = TRUE,
                          dom = 'Brtip',
                          columnDefs = list(list(visible = FALSE, targets
= c(4, 5, 6))),
                        buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
           class = "display compact")
```

Sensitivity Analyses

 Table S11. Sensitivity analysis results comparing the number of exposed workers in the primary analysis versus more stringent analysis. Employee counts are rounded to the nearest 1000.

```{r sensitivity.analysis, echo = F, message = F, warning = F}

## Agent + Occupation Matrix Estimated Exposures
sens.agent.occ.matrix <- JEM.CPS.OEWS %>%
  # Calculate # of employees exposed
  mutate(across(contains(c(dem.list)),
                list(exp_occ = ~{p / 100 * .},
                     R3_exp_occ = ~{p.R3 / 100 * .}),
                .names = "{.col}_{.fn}"))

## Table with employees exposed by agent
sens.agent.matrix <- sens.agent.occ.matrix %>%
  group_by(idchem, agent_label) %>%
  # Calculate the sum of employees exposed across all occupations for each
agent
  summarise(across(contains(c(dem.list)),
              ~sum(., na.rm = T),
              .names = "{.col}_all")) %>%
```

```
  # Remove unnecessary columns
  select(!contains("expected_all"), -All_all, -Male_all, -Female_all, -
BIPOC_all, -AIAN_all, -Asian_all, -Black_all, -H_White_all, -
Multiracial_all, -NH_White_all, -NHPI_all, -Hispanic_all, -NH_all, -
less_HS_all, -HS_diploma_all, -some_college_all, -advanced_degree_all, -
US_native_all, -foreign_citizen_all, -foreign_not_citizen_all) %>%
  ungroup() %>%
  # Filter out N/A values and agents in which no workers are expected to be
exposed in the primary study
  filter(!is.na(agent_label), All_exp_occ_all > 0) %>%
  # Calculate the difference between the estimated number of employees
exposed and the expected number of employees exposed - primary analysis
  mutate("Male_disp_abs" = Male_exp_occ_all - Male_expected_exp_occ_all,
         "Female_disp_abs" = Female_exp_occ_all -
Female_expected_exp_occ_all,
         "BIPOC_disp_abs" = BIPOC_exp_occ_all- BIPOC_expected_exp_occ_all,
         "AIAN_disp_abs" = AIAN_exp_occ_all- AIAN_expected_exp_occ_all,
         "Asian_disp_abs" = Asian_exp_occ_all- Asian_expected_exp_occ_all,
         "Black_disp_abs" = Black_exp_occ_all - Black_expected_exp_occ_all,
         "NHPI_disp_abs" = NHPI_exp_occ_all- NHPI_expected_exp_occ_all,
         "H_White_disp_abs" = H_White_exp_occ_all-
H_White_expected_exp_occ_all,
         "NH_White_disp_abs" = NH_White_exp_occ_all-
NH_White_expected_exp_occ_all,
         "Multiracial_disp_abs" = Multiracial_exp_occ_all-
Multiracial_expected_exp_occ_all,
         "Hispanic_disp_abs" = Hispanic_exp_occ_all-
Hispanic_expected_exp_occ_all,
         "NH_disp_abs" = NH_exp_occ_all - NH_expected_exp_occ_all,
         "less_HS_disp_abs" = less_HS_exp_occ_all -
less_HS_expected_exp_occ_all,
         "HS_diploma_disp_abs" = HS_diploma_exp_occ_all -
HS_diploma_expected_exp_occ_all,
         "some_college_disp_abs" = some_college_exp_occ_all -
some_college_expected_exp_occ_all,
         "advanced_degree_disp_abs" = advanced_degree_exp_occ_all -
advanced_degree_expected_exp_occ_all,
         "US_native_disp_abs" = US_native_exp_occ_all -
US_native_expected_exp_occ_all,
         "foreign_citizen_disp_abs" = foreign_citizen_exp_occ_all -
foreign_citizen_expected_exp_occ_all,
         "foreign_not_citizen_disp_abs" = foreign_not_citizen_exp_occ_all -
foreign_not_citizen_expected_exp_occ_all) %>%
  # Calculate the percent difference between the estimated number of
employees exposed and the expected number of employees exposed - primary
analysis
  mutate("Male_disp_rel" = Male_disp_abs / Male_expected_exp_occ_all * 100,
         "Female_disp_rel" = Female_disp_abs / Female_expected_exp_occ_all
* 100,
         "BIPOC_disp_rel" = BIPOC_disp_abs / BIPOC_expected_exp_occ_all *
100,
         "AIAN_disp_rel" = AIAN_disp_abs / AIAN_expected_exp_occ_all * 100,
         "Asian_disp_rel" = Asian_disp_abs / Asian_expected_exp_occ_all *
100,
```

```
         "Black_disp_rel" = Black_disp_abs / Black_expected_exp_occ_all *
100,
         "NHPI_disp_rel" = NHPI_disp_abs / NHPI_expected_exp_occ_all * 100,
         "H_White_disp_rel" = H_White_disp_abs /
H_White_expected_exp_occ_all * 100,
         "NH_White_disp_rel" = NH_White_disp_abs /
NH_White_expected_exp_occ_all * 100,
         "Multiracial_disp_rel" = Multiracial_disp_abs /
Multiracial_expected_exp_occ_all * 100,
         "Hispanic_disp_rel" = Hispanic_disp_abs /
Hispanic_expected_exp_occ_all * 100,
         "NH_disp_rel" = NH_disp_abs / NH_expected_exp_occ_all * 100,
         "less_HS_disp_rel" = less_HS_disp_abs /
less_HS_expected_exp_occ_all * 100,
         "HS_diploma_disp_rel" = HS_diploma_disp_abs /
HS_diploma_expected_exp_occ_all * 100,
         "some_college_disp_rel" = some_college_disp_abs /
some_college_expected_exp_occ_all * 100,
         "advanced_degree_disp_rel" = advanced_degree_disp_abs /
advanced_degree_expected_exp_occ_all * 100,
         "US_native_disp_rel" = US_native_disp_abs /
US_native_expected_exp_occ_all * 100,
         "foreign_citizen_disp_rel" = foreign_citizen_disp_abs /
foreign_citizen_expected_exp_occ_all * 100,
         "foreign_not_citizen_disp_rel" = foreign_not_citizen_disp_abs /
foreign_not_citizen_expected_exp_occ_all * 100) %>%
  # Calculate the difference between the estimated number of employees
exposed and the expected number of employees exposed - stringent analysis
  mutate("Male_R3_disp_abs" = Male_R3_exp_occ_all -
Male_expected_R3_exp_occ_all,
         "Female_R3_disp_abs" = Female_R3_exp_occ_all -
Female_expected_R3_exp_occ_all,
         "BIPOC_R3_disp_abs" = BIPOC_R3_exp_occ_all-
BIPOC_expected_R3_exp_occ_all,
         "AIAN_R3_disp_abs" = AIAN_R3_exp_occ_all-
AIAN_expected_R3_exp_occ_all,
         "Asian_R3_disp_abs" = Asian_R3_exp_occ_all-
Asian_expected_R3_exp_occ_all,
         "Black_R3_disp_abs" = Black_R3_exp_occ_all -
Black_expected_R3_exp_occ_all,
         "NHPI_R3_disp_abs" = NHPI_R3_exp_occ_all-
NHPI_expected_R3_exp_occ_all,
         "H_White_R3_disp_abs" = H_White_R3_exp_occ_all-
H_White_expected_R3_exp_occ_all,
         "NH_White_R3_disp_abs" = NH_White_R3_exp_occ_all-
NH_White_expected_R3_exp_occ_all,
         "Multiracial_R3_disp_abs" = Multiracial_R3_exp_occ_all-
Multiracial_expected_R3_exp_occ_all,
         "Hispanic_R3_disp_abs" = Hispanic_R3_exp_occ_all-
Hispanic_expected_R3_exp_occ_all,
         "NH_R3_disp_abs" = NH_R3_exp_occ_all - NH_expected_R3_exp_occ_all,
         "less_HS_R3_disp_abs" = less_HS_R3_exp_occ_all -
less_HS_expected_R3_exp_occ_all,
```

```
         "HS_diploma_R3_disp_abs" = HS_diploma_R3_exp_occ_all -
HS_diploma_expected_R3_exp_occ_all,
         "some_college_R3_disp_abs" = some_college_R3_exp_occ_all -
some_college_expected_R3_exp_occ_all,
         "advanced_degree_R3_disp_abs" = advanced_degree_R3_exp_occ_all -
advanced_degree_expected_R3_exp_occ_all,
         "US_native_R3_disp_abs" = US_native_R3_exp_occ_all -
US_native_expected_R3_exp_occ_all,
         "foreign_citizen_R3_disp_abs" = foreign_citizen_R3_exp_occ_all -
foreign_citizen_expected_R3_exp_occ_all,
         "foreign_not_citizen_R3_disp_abs" =
foreign_not_citizen_R3_exp_occ_all -
foreign_not_citizen_expected_R3_exp_occ_all) %>%
  # Calculate the percent difference between the estimated number of
employees exposed and the expected number of employees exposed - stringent
analysis
  mutate("Male_R3_disp_rel" = Male_R3_disp_abs /
Male_expected_R3_exp_occ_all * 100,
         "Female_R3_disp_rel" = Female_R3_disp_abs /
Female_expected_R3_exp_occ_all * 100,
         "BIPOC_R3_disp_rel" = BIPOC_R3_disp_abs /
BIPOC_expected_R3_exp_occ_all * 100,
         "AIAN_R3_disp_rel" = AIAN_R3_disp_abs /
AIAN_expected_R3_exp_occ_all * 100,
         "Asian_R3_disp_rel" = Asian_R3_disp_abs /
Asian_expected_R3_exp_occ_all * 100,
         "Black_R3_disp_rel" = Black_R3_disp_abs /
Black_expected_R3_exp_occ_all * 100,
         "NHPI_R3_disp_rel" = NHPI_R3_disp_abs /
NHPI_expected_R3_exp_occ_all * 100,
         "H_White_R3_disp_rel" = H_White_R3_disp_abs /
H_White_expected_R3_exp_occ_all * 100,
         "NH_White_R3_disp_rel" = NH_White_R3_disp_abs /
NH_White_expected_R3_exp_occ_all * 100,
         "Multiracial_R3_disp_rel" = Multiracial_R3_disp_abs /
Multiracial_expected_R3_exp_occ_all * 100,
         "Hispanic_R3_disp_rel" = Hispanic_R3_disp_abs /
Hispanic_expected_R3_exp_occ_all * 100,
         "NH_R3_disp_rel" = NH_R3_disp_abs / NH_expected_R3_exp_occ_all *
100,
         "less_HS_R3_disp_rel" = less_HS_R3_disp_abs /
less_HS_expected_R3_exp_occ_all * 100,
         "HS_diploma_R3_disp_rel" = HS_diploma_R3_disp_abs /
HS_diploma_expected_R3_exp_occ_all * 100,
         "some_college_R3_disp_rel" = some_college_R3_disp_abs /
some_college_expected_R3_exp_occ_all * 100,
         "advanced_degree_R3_disp_rel" = advanced_degree_R3_disp_abs /
advanced_degree_expected_R3_exp_occ_all * 100,
         "US_native_R3_disp_rel" = US_native_R3_disp_abs /
US_native_expected_R3_exp_occ_all * 100,
         "foreign_citizen_R3_disp_rel" = foreign_citizen_R3_disp_abs /
foreign_citizen_expected_R3_exp_occ_all * 100,
```

```
         "foreign_not_citizen_R3_disp_rel" =
foreign_not_citizen_R3_disp_abs /
foreign_not_citizen_expected_R3_exp_occ_all * 100)

## Prepare data table
table.sens <- sens.agent.matrix %>%
  select(agent_label, All_exp_occ_all, All_R3_exp_occ_all) %>%
  arrange(desc(All_exp_occ_all)) %>%
  # Calculate rank of agents based on # of exposed employees
  mutate_at(vars(contains("All")),
            funs(rank = min_rank(desc(.)))) %>%
  # Calculate rank and estimate differences
  mutate(rank_diff = (All_R3_exp_occ_all_rank - All_exp_occ_all_rank),
         All_diff = (All_R3_exp_occ_all - All_exp_occ_all),
         All_diff_p = (All_R3_exp_occ_all - All_exp_occ_all)/
All_exp_occ_all*100) %>%
  # Round counts and percents
  mutate_at(vars(contains("exp_occ_all") & !contains("rank"), All_diff),
            funs(Round.counts(.))) %>%
  mutate_at(vars(ends_with("_p")),
            funs(Round.percents(.))) %>%
  # Select desired columns
  select(agent_label, All_exp_occ_all, All_R3_exp_occ_all, All_diff,
All_diff_p, All_exp_occ_all_rank, All_R3_exp_occ_all_rank, rank_diff) %>%
  rename(Agent = agent_label,
         "# of exposed workers in primary analysis" = All_exp_occ_all,
         "# of exposed workers in stringent analysis" = All_R3_exp_occ_all,
         "Absolute change (#)" = All_diff,
         "Percent change (%)" = All_diff_p,
         "Primary analysis rank" = All_exp_occ_all_rank,
         "Stringent analysis rank" = All_R3_exp_occ_all_rank,
         "Rank change" = rank_diff
)

# Create filterable datatable to display and explore data
datatable(table.sens,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 25,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")

# sensitivity %>%
#   ggplot(aes(x = 'Rank Change'))+
#   geom_histogram()
#
# sensitivity$'Rank Change' %>%
#   abs() %>%
#   quantile(.95)
#
# sensitivity$All_diff_p %>%
```

```
#   mean(na.rm = T)

```

 Table S12. Sensitivity analysis results comparing the number of workers over or underrepresnted in exposure for all agents in the primary analysis versus more stringent analysis. Except for the persons who identify as non-Hispanic White, persons of any race are of any ethnicity. Persons of Hispanic or Latino ethnicity are of any race are also counted in their preferred race category. BIPOC describes persons that identify as any race/ethnicity other than non-Hispanic White. PA = primary analysis, SA = stringent analysis, AIAN = American Indian and Alaska Native, NHPI = Native Hawaiian and Other Pacific Islander, BIPOC = Black, Indigenous, and People of Color, <High school = less than high school diploma or equivalent, High school = high school diploma or equivalent, some college/associate = some college or associate degree, ≥ Bachelor's = bachelor's or other advanced degree

```{r sensitivity.analysis.disp, warning = F, message = F, echo = F}

## Generate a table to compare estimates of disproportionality for the primary vs stringent analysis

table.sens.disp <- sens.agent.matrix %>%
  # Select desired columns and arrange
  select(agent_label, contains("disp_abs"), contains("disp_rel")) %>%
  select(agent_label, contains(c(dem.list))) %>%
  # Calculate absolute and relative change
  mutate(AIAN_diff = (AIAN_R3_disp_abs - AIAN_disp_abs),
         AIAN_diff_p = (AIAN_diff/abs(AIAN_disp_abs)*100),
         AIAN_change_under = ifelse(AIAN_disp_abs >= 0 & AIAN_R3_disp_abs < 0, "Yes", "No"),
         AIAN_change_over = ifelse(AIAN_disp_abs <= 0 & AIAN_R3_disp_abs > 0, "Yes", "No"),
         AIAN_change = ifelse(AIAN_change_under == "Yes" | AIAN_change_over == "Yes", "Yes", "No"),

         Asian_diff = (Asian_R3_disp_abs - Asian_disp_abs),
         Asian_diff_p = (Asian_diff/abs(Asian_disp_abs)*100),
         Asian_disp_change = ifelse(Asian_diff_p < -100, "yes", "no"),

         Black_diff = (Black_R3_disp_abs - Black_disp_abs),
         Black_diff_p = (Black_diff/abs(Black_disp_abs)*100),
         Black_disp_change = ifelse(Black_diff_p < -100, "yes", "no"),

         Multiracial_diff = (Multiracial_R3_disp_abs - Multiracial_disp_abs),
         Multiracial_diff_p = (Multiracial_diff/ abs(Multiracial_disp_abs)*100),
         Multiracial_disp_change = ifelse(Multiracial_diff_p < -100, "yes", "no"),

         NHPI_diff = (NHPI_R3_disp_abs - NHPI_disp_abs),
         NHPI_diff_p = (NHPI_diff/abs(NHPI_disp_abs)*100),
         NHPI_disp_change = ifelse(NHPI_diff_p < -100, "yes", "no"),
```

```
         NH_White_diff = (NH_White_R3_disp_abs - NH_White_disp_abs),
         NH_White_diff_p = (NH_White_diff/abs(NH_White_disp_abs)*100),
         NH_White_disp_change = ifelse(NH_White_diff_p < -100, "yes",
"no"),

         H_White_diff = (H_White_R3_disp_abs - H_White_disp_abs),
         H_White_diff_p = (H_White_diff/abs(H_White_disp_abs)*100),
         H_White_disp_change = ifelse(H_White_diff_p < -100, "yes", "no"),

         NH_diff = (NH_R3_disp_abs - NH_disp_abs),
         NH_diff_p = (NH_diff/abs(NH_disp_abs)*100),
         NH_disp_change = ifelse(NH_diff_p < -100, "yes", "no"),

         Hispanic_diff = (Hispanic_R3_disp_abs - Hispanic_disp_abs),
         Hispanic_diff_p = (Hispanic_diff/abs(Hispanic_disp_abs)*100),
         Hispanic_disp_change = ifelse(Hispanic_diff_p < -100, "yes",
"no"),

         BIPOC_diff = (BIPOC_R3_disp_abs - BIPOC_disp_abs),
         BIPOC_diff_p = (BIPOC_diff/abs(BIPOC_disp_abs)*100),
         BIPOC_disp_change = ifelse(BIPOC_diff_p < -100, "yes", "no"),

         Female_diff = (Female_R3_disp_abs - Female_disp_abs),
         Female_diff_p = (Female_diff/abs(Female_disp_abs)*100),
         Female_disp_change = ifelse(Female_diff_p < -100, "yes", "no"),

         Male_diff = (Male_R3_disp_abs - Male_disp_abs),
         Male_diff_p = (Male_diff/abs(Male_disp_abs)*100),
         Male_disp_change = ifelse(Male_diff_p < -100, "yes", "no"),

         less_HS_diff = (less_HS_R3_disp_abs - less_HS_disp_abs),
         less_HS_diff_p = (less_HS_diff/abs(less_HS_disp_abs)*100),
         less_HS_disp_change = ifelse(less_HS_diff_p < -100, "yes", "no"),

         HS_diploma_diff = (HS_diploma_R3_disp_abs - HS_diploma_disp_abs),
         HS_diploma_diff_p = (HS_diploma_diff/
abs(HS_diploma_disp_abs)*100),
         HS_diploma_disp_change = ifelse(HS_diploma_diff_p < -100, "yes",
"no"),

         some_college_diff = (some_college_R3_disp_abs -
some_college_disp_abs),
         some_college_diff_p = (some_college_diff/
abs(some_college_disp_abs)*100),
         some_college_disp_change = ifelse(some_college_diff_p < -100,
"yes", "no"),

         advanced_degree_diff = (advanced_degree_R3_disp_abs -
advanced_degree_disp_abs),
         advanced_degree_diff_p = (advanced_degree_diff/
abs(advanced_degree_disp_abs)*100),
         advanced_degree_disp_change = ifelse(advanced_degree_diff_p <
-100, "yes", "no"),
```

```
         US_native_diff = (US_native_R3_disp_abs - US_native_disp_abs),
         US_native_diff_p = (US_native_diff/abs(US_native_disp_abs)*100),
         US_native_disp_change = ifelse(US_native_diff_p < -100, "yes",
"no"),

         foreign_citizen_diff = (foreign_citizen_R3_disp_abs -
foreign_citizen_disp_abs),
         foreign_citizen_diff_p = (foreign_citizen_diff/
abs(foreign_citizen_disp_abs)*100),
         foreign_citizen_disp_change = ifelse(foreign_citizen_diff_p <
-100, "yes", "no"),

         foreign_not_citizen_diff = (foreign_not_citizen_R3_disp_abs -
foreign_not_citizen_disp_abs),
         foreign_not_citizen_diff_p = (foreign_not_citizen_diff/
abs(foreign_not_citizen_disp_abs)*100),
         foreign_not_citizen_disp_change =
ifelse(foreign_not_citizen_diff_p < -100, "yes", "no")) %>%

  # Round counts and percents
  mutate_at(vars(contains("disp_abs"), contains("diff") & !contains("_p")),
            funs(Round.counts(.))) %>%
  mutate_at(vars(ends_with("_p"), contains("disp_rel")),
            funs(Round.percents(.))) %>%

  # Select and arrange the data
  select(agent_label, contains("disp_abs")) %>%
  select(agent_label, contains(c(dem.list)), -H_White_disp_abs, -
H_White_R3_disp_abs, -NH_disp_abs, -NH_R3_disp_abs) %>%
  arrange(agent_label) %>%

  # Rename columns
  rename(Agent = agent_label,
         "# AIAN, PA" = AIAN_disp_abs,
         "# AIAN, SA" = AIAN_R3_disp_abs,
         "# Asian, PA" = Asian_disp_abs,
         "# Asian, SA" = Asian_R3_disp_abs,
         "# Black, PA" = Black_disp_abs,
         "# Black, SA" = Black_R3_disp_abs,
         "# Multiracial, PA" = Multiracial_disp_abs,
         "# Multiracial, SA" = Multiracial_R3_disp_abs,
         "# NHPI, PA" = NHPI_disp_abs,
         "# NHPI, SA" = NHPI_R3_disp_abs,
         "# White, non-Hispanic, PA" = NH_White_disp_abs,
         "# White, non-Hispanic, SA" = NH_White_R3_disp_abs,
         "# Hispanic, PA" = Hispanic_disp_abs,
         "# Hispanic, SA" = Hispanic_R3_disp_abs,
         "# BIPOC, PA" = BIPOC_disp_abs,
         "# BIPOC, SA" = BIPOC_R3_disp_abs,
         "# Female, PA" = Female_disp_abs,
         "# Female, SA" = Female_R3_disp_abs,
         "# Male, PA" = Male_disp_abs,
         "# Male, SA" = Male_R3_disp_abs,
```

```
          "# <High school, PA" = less_HS_disp_abs,
          "# <High school, SA" = less_HS_R3_disp_abs,
          "# High school, PA" = HS_diploma_disp_abs,
          "# High school, SA" = HS_diploma_R3_disp_abs,
          "# Some college, PA" = some_college_disp_abs,
          "# Some college, SA" = some_college_R3_disp_abs,
          "# ≥Bachelor's, PA" = advanced_degree_disp_abs,
          "# ≥Bachelor's, SA" = advanced_degree_R3_disp_abs,
          "# Native-born, PA" = US_native_disp_abs,
          "# Native-born, SA" = US_native_R3_disp_abs,
          "# Foreign-born, citizen, PA" = foreign_citizen_disp_abs,
          "# Foreign-born, citizen, SA" = foreign_citizen_R3_disp_abs,
          "# Foreign-born, noncitizen, PA" = foreign_not_citizen_disp_abs,
          "# Foreign-born, noncitizen, SA" =
foreign_not_citizen_R3_disp_abs)

# Generate filterable data table
datatable(table.sens.disp,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")
```

```{r sens.disp.graphs, warning = F, message = F, echo = F}

# # Prepare tables to be used for the graph
#
# graph.sens.disp.raceeth <- table.sens.disp %>%
#   # Count # of agents overrepresented for primary analysis
#   summarise(AIAN = sum(AIAN_R3_disp_abs > 0),
#             Asian = sum(Asian_R3_disp_abs > 0),
#             Black = sum(Black_R3_disp_abs > 0),
#             Multiracial = sum(Multiracial_R3_disp_abs > 0),
#             NHPI = sum(NHPI_R3_disp_abs > 0),
#             "White, non-Hispanic" = sum(NH_White_R3_disp_abs > 0),
#             Hispanic = sum(Hispanic_R3_disp_abs > 0),
#             BIPOC = sum(BIPOC_R3_disp_abs > 0)) %>%
#   # Pivot long
#   pivot_longer(cols = AIAN:BIPOC,
#                names_to = "Race/ethnicity",
#                values_to = "# Overrepresented") %>%
#   # Add column to designate which analysis the estimates are for
#   mutate(Analysis = "Stringent analysis") %>%
#   # Relevel factors so they display in desired order on graph
#   mutate(`Race/ethnicity` = fct_relevel(`Race/ethnicity`, "AIAN",
"Asian", "Black", "Multiracial", "NHPI", "White, non-Hispanic", "Hispanic",
"BIPOC")) %>%
```

```
#   # Merge with primary analysis results
#   rbind(graph.disp.raceeth %>%
#               # Add column to designate which analysis the estimates are
for
#               mutate(Analysis = "Primary analysis"))
#
#
# graph.disp.sex <- table.agent.disp.sex %>%
#   # Count # of agents overrepresented
#   summarise(Female = sum(Female_disp_abs > 0),
#             Male = sum(Male_disp_abs > 0)) %>%
#   # Pivot long
#   pivot_longer(cols = Female:Male,
#                names_to = "Sex",
#                values_to = "# Overrepresented") %>%
#   # Relevel factors so they display in desired order on graph
#   mutate(Sex = fct_relevel(Sex, "Female", "Male"))
#
# graph.disp.education <- table.agent.disp.education %>%
#   # Count # of agents overrepresented
#   summarise("<High school" = sum(`# Less than high school diploma or
equivalent` > 0),
#             "High school" = sum(`# High school diploma or equivalent` >
0),
#             "Some college/associate" = sum(`# Some college or associate
degree` > 0),
#             "≥Bachelor's" = sum(`# Bachelor's or other advanced degree` >
0)) %>%
#   # Pivot long
#   pivot_longer(cols = `<High school`:`≥Bachelor's`,
#                names_to = "Education Status",
#                values_to = "# Overrepresented") %>%
#   # Relevel factors so they display in desired order on graph
#   mutate(`Education Status` = fct_relevel(`Education Status`, "<High
school", "High school", "Some college/associate", "≥Bachelor's"))
#
# graph.disp.nativity <- table.agent.disp.nativity %>%
#   # Count # of agents overrepresented
#   summarise("US native" = sum(`# US native` > 0),
#             "Foreign Born, US Citizen" = sum(`# Foreign Born, US Citizen`
> 0),
#             "Foreign Born, Non-US Citizen" = sum(`# Foreign Born, Non-US
Citizen` > 0)) %>%
#   # Pivot long
#   pivot_longer(cols = `US native`:`Foreign Born, Non-US Citizen`,
#                names_to = "Nativity and Citizenship Status",
#                values_to = "# Overrepresented") %>%
#   # Relevel factors so they display in desired order on graph
#   mutate(`Nativity and Citizenship Status` = fct_relevel(`Nativity and
Citizenship Status`, "US native", "Foreign Born, US Citizen", "Foreign
Born, Non-US Citizen"))
#
# # Generate plot for each sociodemographic category
#
```

```
# ggplot.sens.disp.raceeth <- graph.sens.disp.raceeth %>%
#   ggplot(aes(x = `Race/ethnicity`, y = `# Overrepresented`, fill=
Analysis)) +
#   geom_col(position = "dodge") +
#   geom_text(
#     aes(label = `# Overrepresented`),
#     color = "black",
#     size = 4,
#     vjust = -0.5) +
#   theme(panel.grid.major = element_blank(),
#         panel.grid.minor = element_blank(),
#         panel.background = element_blank(),
#         axis.line = element_line(colour = "black"),
#         axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size
= 12),
#         axis.text.y = element_text(size = 12),
#         axis.title = element_text(size = 12)) +
#   scale_y_continuous(limits = c(NA, 250))
#
# ggplot.sens.disp.raceeth
#
# ggplot.disp.sex <- graph.disp.sex %>%
#   ggplot(aes(x = `Sex`, y = `# Overrepresented`)) +
#   geom_col(position="dodge") +
#   geom_text(
#     aes(label = `# Overrepresented`),
#     color = "black",
#     size = 4,
#     vjust = -0.5) + theme_classic()+
#   labs(x = "Sex", y = "# Agents Overrepresented") +
#   theme(panel.grid.major = element_blank(),
#         panel.grid.minor = element_blank(),
#         panel.background = element_blank(),
#         axis.line = element_line(colour = "black"),
#         axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size
= 12),
#         axis.text.y = element_text(size = 12),
#         axis.title = element_text(size = 12)) +
#   scale_y_continuous(limits = c(NA, 250))
# ggplot.disp.sex
#
# ggplot.disp.education <- graph.disp.education %>%
#   ggplot(aes(x = `Education Status`, y = `# Overrepresented`)) +
#   geom_col() +
#   geom_text(
#     aes(label = `# Overrepresented`),
#     color = "black",
#     size = 4,
#     vjust = -0.5) + theme_classic()+
#   theme(panel.grid.major = element_blank(),
#         panel.grid.minor = element_blank(),
#         panel.background = element_blank(),
#         axis.line = element_line(colour = "black"),
```

```
#         axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size
= 12),
#         axis.text.y = element_text(size = 12),
#         axis.title = element_text(size = 12)) +
#   scale_y_continuous(limits = c(NA, 250))
# ggplot.disp.education
#
# ggplot.disp.nativity <- graph.disp.nativity %>%
#   ggplot(aes(x = `Nativity and Citizenship Status`, y = `#
Overrepresented`)) +
#   geom_col() +
#   geom_text(
#     aes(label = `# Overrepresented`),
#     color = "black",
#     size = 4,
#     vjust = -0.5) + theme_classic()+
#   theme(panel.grid.major = element_blank(),
#         panel.grid.minor = element_blank(),
#         panel.background = element_blank(),
#         axis.line = element_line(colour = "black"),
#         axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size
= 12),
#         axis.text.y = element_text(size = 12),
#         axis.title = element_text(size = 12)) +
#   scale_y_continuous(limits = c(NA, 250))
# ggplot.disp.nativity

# ggplot.disp.combined <- ggarrange(ggplot.disp.raceeth, ggplot.disp.sex,
ggplot.disp.education, ggplot.disp.nativity,
#                                   labels = c("A", "B", "C", "D"),
#                                   ncol = 2, nrow = 2)
# ggplot.disp.combined
```

Additional Tables

US Worker Sociodemographics, 2019

```{r sociodemographics, warning = F, message = F, echo = F}
# Create a table with summary information about employees in the US
emp.totals.table <- CPS.data %>%
  summarise(across(!contains(c("Cen_2010", "Description", "expected")),
sum)) %>%
  pivot_longer(cols = everything()) %>%
  mutate(percent = paste0(format(Round((value / sum(CPS.data$All) * 100),
1), nsmall=1), "%"),
         value = Round(value, -3),
         value = format(value, big.mark=",", scientific=FALSE)) %>%
  mutate(Category = ifelse(name == "AIAN", "Race/Ethnicity",
                           ifelse(name == "All", "Total",
                                  ifelse(name =="Female", "Sex",
                                         ifelse(name == "less_HS",
"Education Status",
                                                ifelse(name == "US_native",
"Nativity and Citizenship Status", NA)))))) %>%
```

```
  mutate(name = replace(name, name %in% c("All"), NA),
         name = replace(name, name == "H_White", "White, Hispanic"),
         name = replace(name, name == "NH_White", "White, Non-Hispanic"),
         name = replace(name, name == "NH", "Not Hispanic or Latino"),
         name = replace(name, name == "Hispanic", "Hispanic or Latino"),
         name = replace(name, name == "AIAN", "American Indian and Alaska
Native"),
         name = replace(name, name == "NHPI", "Native Hawaiian or Other
Pacific Islander"),
         name = replace(name, name == "Black", "Black or African
American"),
         name = replace(name, name == "less_HS", "Less than high school
diploma"),
         name = replace(name, name == "HS_diploma", "High school diploma or
equivalent"),
         name = replace(name, name == "some_college", "Some college or
associate degree"),
         name = replace(name, name == "advanced_degree", "Bachelor's or
other advanced degree"),
         name = replace(name, name == "US_native", "US Native"),
         name = replace(name, name == "foreign_citizen", "Foreign born, US
citizen"),
         name = replace(name, name == "foreign_not_citizen", "Foreign born,
non-US citizen")) %>%
  mutate(name = fct_relevel(name, "American Indian and Alaska Native",
"Asian", "Black or African American", "Multiracial", "Native Hawaiian or
Other Pacific Islander", "White, Non-Hispanic", "White, Hispanic",
"Hispanic or Latino", "Not Hispanic or Latino", "BIPOC", "Female", "Male",
"Less than high school diploma", "High school diploma or equivalent", "Some
college or associate degree", "Bachelor's or other advanced degree", "US
Native", "Foreign born, US citizen", "Foreign born, non-US citizen")) %>%
  select(Category, everything()) %>%
  arrange(name, Category) %>%
  rename("Sociodemographic group" = name,
         "Employee Count" = value,
         "Percent of Workforce" = percent)

pander(emp.totals.table, justify = "left", split.table = Inf, missing = "",
caption = "Table S13. Average employment estimates in the United States in
2019. Employees that identify as any race, except those that identify as
White, can be of any ethnicity. Employees that identify as any ethnicity
can be of any race.")
```

Occupations with No Exposure Information

```{r no.coverage, warning = F, message = F, echo = F}

# Create a table that contains occupations with no exposure information
no.info <- coverage %>%
  filter(exp_info == "no") %>%
  select(!exp_info) %>%
  arrange(desc(All)) %>%
  rename("Census Occupation Code" = Cen_2010,
         "Total # of Employees" = All)
```

```
# Display table
datatable(no.info,
          filter = "top",
          extensions = "Buttons",
          options = list(pageLength = 10, dom = 'Brtip',
                         buttons = list(c('pageLength', 'copy', 'csv',
'excel', 'pdf'))),
          caption = "Table S14. List of occupations in the United States
that do no have exposure information. Occupations are listed by Census
occupation code and arranged by those with the most number of employees.")


```
Estimates of high exposure (number and percent of demographic group
exposed)

By race and ethnicity

 Table S15. Number and percent of US workers
highly exposed to occupational agents in CANJEM by race and ethnicity,
2019. Employee counts are rounded to the nearest 1000. Except for the
persons who identify as non-Hispanic White, persons of any race are of any
ethnicity. Persons of Hispanic or Latino ethnicity are of any race are also
counted in their preferred race category. BIPOC describes persons that
identify as any race/ethnicity other than non-Hispanic White. AIAN =
American Indian and Alaska Native, NHPI = Native Hawaiian and Other Pacific
Islander, BIPOC = Black, Indigenous, and People of Color

```{r table.agent.raceeth.h, warning = F, message = F, echo = F}

## Generate a table with estimates of employees exposed by agent for
detailed race/eth groups

table.agent.raceeth.h <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("h_exp_occ_all", "h_exp_perc")) & !
contains("expected")) %>%
  select(agent_label, starts_with("All"), contains("AIAN"),
contains("Asian"), contains("Black"), contains("Multiracial"),
contains("NHPI"), contains("NH_White"), contains("Hispanic"),
contains("BIPOC")) %>%
# Arrange the data
  arrange(desc(All_h_exp_occ_all)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# Exposed" = All_h_exp_occ_all,
         "# BIPOC Exposed" = BIPOC_h_exp_occ_all,
         "# AIAN Exposed" = AIAN_h_exp_occ_all,
         "# Asian Exposed" = Asian_h_exp_occ_all,
         "# Black Exposed" = Black_h_exp_occ_all,
         "# NHPI Exposed" = NHPI_h_exp_occ_all,
         "# White, non-Hispanic Exposed" = NH_White_h_exp_occ_all,
         "# Multiracial Exposed" = Multiracial_h_exp_occ_all,
```

```
         "# Hispanic Exposed" = Hispanic_h_exp_occ_all,
         "% Exposed" = All_h_exp_perc,
         "% BIPOC Exposed" = BIPOC_h_exp_perc,
         "% AIAN Exposed" = AIAN_h_exp_perc,
         "% Asian Exposed" = Asian_h_exp_perc,
         "% Black Exposed" = Black_h_exp_perc,
         "% NHPI Exposed" = NHPI_h_exp_perc,
         "% White, non-Hispanic Exposed" = NH_White_h_exp_perc,
         "% Multiracial Exposed" = Multiracial_h_exp_perc,
         "% Hispanic Exposed" = Hispanic_h_exp_perc)

# Generate filterable data table
datatable(table.agent.raceeth.h,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")

```

By sex

 Table S16. Number and percent of US workers highly exposed to occupational agents in CANJEM by sex, 2019. Employee counts are rounded to the nearest 1000.

```{r table.agent.sex.h, warning = F, message = F, echo = F}

## Generate a table with estimates of employees exposed by agent and sex

table.agent.sex.h <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("h_exp_occ_all", "h_exp_perc")) & !
contains("expected")) %>%
  select(agent_label, starts_with("All"), starts_with("Female"),
starts_with("Male")) %>%
# Arrange the data
  arrange(desc(All_h_exp_occ_all)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# Exposed" = All_h_exp_occ_all,
         "# Male Exposed" = Male_h_exp_occ_all,
         "# Female Exposed" = Female_h_exp_occ_all,
         "% Exposed" = All_h_exp_perc,
         "% Male Exposed" = Male_h_exp_perc,
         "% Female Exposed" = Female_h_exp_perc)

# Generate filterable data table
datatable(table.agent.sex.h,
          filter = "top",
```

```
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")

```

By education status

 Table S17. Number and percent of US workers highly exposed to occupational agents in CANJEM by education status, 2019. Employee counts are rounded to the nearest 1000.

```{r table.agent.education.h, warning = F, message = F, echo = F}

## Generate a table with estimates of employees exposed by agent and
education status

table.agent.education.h <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("h_exp_occ_all", "h_exp_perc")) & !
contains("expected")) %>%
  select(agent_label, starts_with("All"), contains("less_HS"),
contains("HS_diploma"), contains("some_college"),
contains("advanced_degree")) %>%
# Arrange the data
  arrange(desc(All_h_exp_occ_all)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# All" = All_h_exp_occ_all,
         "# Less than high school diploma or equivalent" =
less_HS_h_exp_occ_all,
         "# High school diploma or equivalent" = HS_diploma_h_exp_occ_all,
         "# Some college or associate degree" = some_college_h_exp_occ_all,
         "# Bachelor's or other advanced degree" =
advanced_degree_h_exp_occ_all,
         "% All" = All_h_exp_perc,
         "% Less than high school diploma or equivalent" =
less_HS_h_exp_perc,
         "% High school diploma or equivalent" = HS_diploma_h_exp_perc,
         "% Some college or associate degree" = some_college_h_exp_perc,
         "% Bachelor's or other advanced degree" =
advanced_degree_h_exp_perc)

# Generate filterable data table
datatable(table.agent.education.h,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
```

```
                        buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")

```

By nativity and citizenship status

 Table S18. Number and percent of US workers highly exposed to occupational agents in CANJEM by nativity and citizenship status, 2019. Employee counts are rounded to the nearest 1000.

```{r table.agent.nativity.h, warning = F, message = F, echo = F}

## Generate a table with estimates of employees exposed by agent and
nativity and citizenship status

table.agent.nativity.h <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("h_exp_occ_all", "h_exp_perc")) & !
contains("expected")) %>%
  select(agent_label, starts_with("All"), contains("US_native"),
contains("foreign_citizen"), contains("foreign_not_citizen")) %>%
# Arrange the data
  arrange(desc(All_h_exp_occ_all)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "All" = All_h_exp_occ_all,
         "US Native" = US_native_h_exp_occ_all,
         "Foreign Born, US Citizen" = foreign_citizen_h_exp_occ_all,
         "Foreign Born, Non-US Citizen" =
foreign_not_citizen_h_exp_occ_all,
         "% All" = All_h_exp_perc,
         "% US Native" = US_native_h_exp_perc,
         "% Foreign Born, US Citizen" = foreign_citizen_h_exp_perc,
         "% Foreign Born, Non-US Citizen" = foreign_not_citizen_h_exp_perc)

# Generate filterable data table
datatable(table.agent.nativity.h,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")

```

Estimates of disproportionality at a high level of exposure (number and percent of workers over or underrepresented)

By race and ethnicity

 Table S19. Number and percent of US workers over or underrepresented in exposure at a high level to occupational agents in CANJEM by race and ethnicity, 2019. The number and percent of employees over or underrepresented represents the absolute and relative differences between the number of workers estimated to be exposed and the number of workers expected to be exposed in a counterfactual scenario in which workers of each sociodemographic group are evenly distributed across all occupations in the US based on their overall proportion of the total workforce. Employee counts are rounded to the nearest 1000. Except for the persons who identify as non-Hispanic White, persons of any race are of any ethnicity. Persons of Hispanic or Latino ethnicity are of any race are also counted in their preferred race category. BIPOC describes persons that identify as any race/ethnicity other than non-Hispanic White. AIAN = American Indian and Alaska Native, NHPI = Native Hawaiian and Other Pacific Islander, BIPOC = Black, Indigenous, and People of Color

```{r table.agent.disp.raceeth.h, warning = F, message = F, echo = F}

## Generate a table with estimates of workers over or under expected for each agent by detailed race/eth groups

table.agent.disp.raceeth.h <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("h_disp_abs", "h_disp_rel"))) %>%
  select(agent_label, contains("BIPOC"), contains("Hispanic"),
contains("AIAN"), contains("Asian"), contains("Black"),
contains("Multiracial"), contains("NHPI"), contains("NH_White")) %>%
# Arrange the data
  arrange(desc(BIPOC_h_disp_rel)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# BIPOC" = BIPOC_h_disp_abs,
         "# AIAN" = AIAN_h_disp_abs,
         "# Asian" = Asian_h_disp_abs,
         "# Black" = Black_h_disp_abs,
         "# NHPI" = NHPI_h_disp_abs,
         "# White" = NH_White_h_disp_abs,
         "# Multiracial" = Multiracial_h_disp_abs,
         "# Hispanic" = Hispanic_h_disp_abs,
         "% BIPOC" = BIPOC_h_disp_rel,
         "% AIAN" = AIAN_h_disp_rel,
         "% Asian" = Asian_h_disp_rel,
         "% Black" = Black_h_disp_rel,
         "% NHPI" = NHPI_h_disp_rel,
         "% White" = NH_White_h_disp_rel,
         "% Multiracial" = Multiracial_h_disp_rel,
         "% Hispanic" = Hispanic_h_disp_rel)

# Generate filterable data table
datatable(table.agent.disp.raceeth.h,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
```

```
                        buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")


```

By sex

 Table S20. Number and percent of US workers over or underrepresented in exposure at a high level to occupational agents in CANJEM by sex, 2019. The number and percent of employees over or underrepresented represents the absolute and relative differences between the number of workers estimated to be exposed and the number of workers expected to be exposed in a counterfactual scenario in which workers of each sociodemographic group are evenly distributed across all occupations in the US based on their overall proportion of the total workforce. Employee counts are rounded to the nearest 1000.
```{r table.agent.disp.sex.h, warning = F, message = F, echo = F}

## Generate a table with estimates of workers over or under expected for each agent by detailed race/eth groups

table.agent.disp.sex.h <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("h_disp_abs", "h_disp_rel"))) %>%
  select(agent_label, starts_with("Female"), starts_with("Male")) %>%
# Arrange the data
  arrange(desc(Female_h_disp_rel)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# Male" = Male_h_disp_abs,
         "# Female" = Female_h_disp_abs,
         "% Male" = Male_h_disp_rel,
         "% Female" = Female_h_disp_rel)

# Generate filterable data table
datatable(table.agent.disp.sex.h,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy',
'csv', 'excel', 'pdf'))),
          class = "display compact")


```


By education status

 Table S21. Number and percent of US workers over or underrepresented in exposure at a high level to occupational agents in CANJEM by education status, 2019. The number and percent of employees over or underrepresented represents the absolute and relative differences between the number of workers estimated to be exposed and the number of workers expected to be exposed in a counterfactual scenario in which workers of each sociodemographic group are evenly distributed across all occupations in the US based on their overall proportion of the total workforce. Employee counts are rounded to the nearest 1000. <High school = less than high school diploma or equivalent, High school = high school diploma or equivalent, some college/associate = some college or associate degree, ≥ Bachelor's = bachelor's or other advanced degree

```{r table.agent.disp.education.h, warning = F, message = F, echo = F}

## Generate a table with estimates of workers over or under expected for each agent by education status

table.agent.disp.education.h <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, OSHA_reg, Type, contains(c("h_disp_abs", "h_disp_rel"))) %>%
  select(agent_label, OSHA_reg, Type, contains("less_HS"), contains("HS_diploma"), contains("some_college"), contains("advanced_degree")) %>%
  # Arrange the data
  arrange(desc(less_HS_h_disp_rel)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "OSHA Regulated" = OSHA_reg,
         "# Less than high school diploma or equivalent" = less_HS_h_disp_abs,
         "# High school diploma or equivalent" = HS_diploma_h_disp_abs,
         "# Some college or associate degree" = some_college_h_disp_abs,
         "# Bachelor's or other advanced degree" = advanced_degree_h_disp_abs,
         "% Less than high school diploma or equivalent" = less_HS_h_disp_rel,
         "% High school diploma or equivalent" = HS_diploma_h_disp_rel,
         "% Some college or associate degree" = some_college_h_disp_rel,
         "% Bachelor's or other advanced degree" = advanced_degree_h_disp_rel)

# Generate filterable data table
datatable(table.agent.disp.education.h,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy', 'csv', 'excel', 'pdf'))),
          class = "display compact")
```

```

#### By nativity and citizenship status

<span style = "color:gray"> Table S22. Number and percent of US workers over or underrepresented in exposure at a high level to occupational agents in CANJEM by nativity and citizenship status, 2019. The number and percent of employees over or underrepresented represents the absolute and relative differences between the number of workers estimated to be exposed and the number of workers expected to be exposed in a counterfactual scenario in which workers of each sociodemographic group are evenly distributed across all occupations in the US based on their overall proportion of the total workforce. Employee counts are rounded to the nearest 1000.</span>
```{r table.agent.disp.nativity.h, warning = F, message = F, echo = F}

## Generate a table with estimates of workers over or under expected for each agent by detailed race/eth groups

table.agent.disp.nativity.h <- agent.matrix.rounded %>%
  # Select desired columns
  select(agent_label, contains(c("h_disp_abs", "h_disp_rel"))) %>%
  select(agent_label, contains("US_native"), contains("foreign_citizen"), contains("foreign_not_citizen")) %>%
# Arrange the data
  arrange(desc(foreign_not_citizen_h_disp_rel)) %>%
  #Rename columns
  rename("Agent" = agent_label,
         "# US native" = US_native_h_disp_abs,
         "# Foreign Born, US Citizen" = foreign_citizen_h_disp_abs,
         "# Foreign Born, Non-US Citizen" = foreign_not_citizen_h_disp_abs,
         "% US native" = US_native_h_disp_rel,
         "% Foreign Born, US Citizen" = foreign_citizen_h_disp_rel,
         "% Foreign Born, Non-US Citizen" = foreign_not_citizen_h_disp_rel)

# Generate filterable data table
datatable(table.agent.disp.nativity.h,
          filter = "top",
          extensions = c("Buttons"),
          options = list(pageLength = 10,
                         scrollX = TRUE,
                         dom = 'Brtip',
                         buttons = list(I('colvis'),c('pageLength','copy', 'csv', 'excel', 'pdf'))),
          class = "display compact")


# FOR THESIS REPORT ONLY - Merging all tables together
# table.agent.raceeth %>%
#   left_join(table.agent.sex %>%
#               select(-"OSHA Regulated", -"Type"),
#             by = "Agent") %>%
#   left_join(table.agent.education %>%
```
```

```
#               select(-"OSHA Regulated", -"Type", -"# All", -"% All"),
#             by = "Agent") %>%
#   left_join(table.agent.nativity %>%
#               select(-"OSHA Regulated", -"Type", -"# All", -"% All"),
#             by = "Agent") %>%
#   write.csv("table.all.csv")
#
# table.agent.disp.raceeth %>%
#   left_join(table.agent.disp.sex %>%
#               select(-"OSHA Regulated", -"Type"),
#             by = "Agent") %>%
#   left_join(table.agent.disp.education %>%
#               select(-"OSHA Regulated", -"Type"),
#             by = "Agent") %>%
#   left_join(table.agent.disp.nativity %>%
#               select(-"OSHA Regulated", -"Type"),
#             by = "Agent") %>%
#   write.csv("table.disp.all.csv")
#
# graph.disp <- agent.matrix.rounded %>%
#   select(agent_label, contains(c("disp_rel")) & !contains("_h"))
#
# colnames(graph.disp) <- gsub("_disp_rel", "", colnames(graph.disp))
#
# graph.disp <- graph.disp %>%
#   pivot_longer(cols = contains(c(dem.list))) %>%
#   mutate(name = replace(name, name == "NH_White", "White"),
#          name = replace(name, name == "less_HS", "Less than HS diploma or
equivalent"),
#          name = replace(name, name == "HS_diploma", "HS diploma or
equivalent"),
#          name = replace(name, name == "some_college", "Some college or
associate degree"),
#          name = replace(name, name == "advanced_degree", "Bachelor's or
other advanced degree"),
#          name = replace(name, name == "US_native", "US Native"),
#          name = replace(name, name == "foreign_citizen", "Foreign born,
US citizen"),
#          name = replace(name, name == "foreign_not_citizen", "Foreign
born, non-US citizen")) %>%
#     filter(name != "NH" & name != "H_White") %>%
#   mutate(name = fct_relevel(name, "Male", "Female", "BIPOC", "Hispanic",
"AIAN", "Asian", "Black", "Multiracial", "NHPI", "White", "Less than HS
diploma or equivalent", "HS diploma or equivalent", "Some college or
associate degree", "Bachelor's or other advanced degree", "US Native",
"Foreign born, US citizen", "Foreign born, non-US citizen"))
#
#
# ggplot(data = graph.disp %>%
#          filter(agent_label == "Cleaning agents"), aes(x=name, y =
value))+
#   geom_bar(stat = "identity") +
#   labs(title = "Cleaning agents", x = "Sociodemographic Group", y = "%
Over or underrepresented") +
```

```
#   theme(axis.text.x = element_text(angle = 90, hjust = 0.95))
#

```