Homework 1

Introduction

This problem set will replicate a well-known paper on racial bias in the labor market: "Are Emily and Greg More Employable Than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination" by Marianne Bertrand and Sendhil Mullainathan. The paper, which I'll refer to as BM for short, appears in Volume 94, Issue #4 of the *American Economic Review* (AER). Before beginning this lab, visit the website of the *American Economic Review* and search for the paper. Once you've found it, download a pdf of the paper. Alternatively, you can also download this paper by searching for it in the UPenn library.

Exercise #1

Read the introduction and conclusion of BM, and then write a one-paragraph summary addressing the following points:

1. What question does the paper try to answer?
2. What data and methodology are used to address the question?
3. What are the key findings?

Solution to Exercise #1

1. The paper tries to answer if there is still a differential racial treatment by race in the U.S. labor market.
2. Data is being collected from job applicants resume outside Boston and Chicago newspapers. There are few variables such as :
    1) college degree (dummy),
    2) years of experience,
    3) volunteering experience (dummy),
    4) military experience (dummy),
    5) email address (dummy),
    6) employment holes (dummy),
    7) work in school (dummy),
    8) honors (dummy),
    9) computer skills (dummy),
    10) special skills (dummy),
    11) fraction high school dropouts in applicant's zip code,
    12) fraction college or more in applicant's zip code,
    13) fraction Whites in applicant's zip code,

14) fraction African Americans in applicant's zip code
15) long (median per-capita income) in applicant's zip code

3.
- A white applicant have a higher chance for the callback for every 10 ads job vacancy they applied for, compared to african american if they want to get a callback they have to apply for 15 ads job vacancy.
- The paper found that a higher quality resumes of white-named applicant, they will receive more callback rates compared to a low quality resumes of white-named applicant. The same effects applied as well for the african-american named applicant, but with fewer callback rates compared to the white-named one. Three times larger improved callback rates for the white-named applicant compared to african-american named applicant. In-short, the white-named applicant receive more benefits in terms of improved resumes quality.
- The neighborhood and address location doesn't contribute for the callback rates. There is no significant callback rates for african-american people that lives in the whiter, more educated zip code compared to not live in that neighborhood. This one applies for the white-named applicant as well.
- There is a little systematic relationship between the requirements of the job and the racial gap in callback and the variables do not show any statistically significant.

Importing the Dataset

The dataset is in Canvas as `lakisha_aer.dta`. The extension `dta` implies that this is a STATA file. STATA is a commercial statistics software with a proprietary file format. Fortunately, some enterprising open-source programmers have written software that can decode `.dta` files and convert them into other formats. We'll use the function `read_dta()` to convert `lakisha_aer.dta` into a tibble that we can manipulate with `dplyr`. This function is in the package `haven`, which also contains functions for converting data from SPSS, STATA, and SAS formats. Make sure to install `haven` before proceeding.

Note that you'll have to specify the directory where you've saved the data. Below, I use `setwd()` to specify the directory. The path is for a Mac. If you're using Windows, it's a little trickier to specify the right file path: you may need to Google this.

Load the `.dta` file as a tibble into R using `read_dta()`. Call your tibble `ResumeNames`.

```
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.1 --

v ggplot2 3.3.5     v purrr   0.3.4
v tibble  3.1.4     v dplyr   1.0.7
v tidyr   1.1.3     v stringr 1.4.0
v readr   2.0.1     v forcats 0.5.1

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
library(haven)
setwd("/Users/damarega/Documents/Econ224/")
ResumeNames = read_dta('lakisha_aer.dta')
```

Each row in `ResumeNames` corresponds to a single fictitious job applicant. For instance, `sex` and `race` refer to the gender and ethnicity of the (fictitious) individual in the resume.

Exercise #2

1. Display the tibble `ResumeNames`. How many rows and columns does it have?
2. Display only the columns `sex`, `race` and `firstname` of `ResumeNames`.
3. Add two new columns to `ResumeNames`: `female` should take the value `TRUE` if `sex` is female, and `black` should take value `TRUE` if `race` is "b".

Solution to Exercise #2

```
#number 1
tibble(ResumeNames)
```

```
# A tibble: 4,870 x 65
   id    ad    education ofjobs yearsexp honors volunteer military empholes
   <chr> <chr>     <dbl>  <dbl>    <dbl>  <dbl>     <dbl>    <dbl>    <dbl>
 1 b     1             4      2        6      0         0        0        1
 2 b     1             3      3        6      0         1        1        0
 3 b     1             4      1        6      0         0        0        0
 4 b     1             3      4        6      0         1        0        1
 5 b     1             3      3       22      0         0        0        0
 6 b     1             4      2        6      1         0        0        0
 7 b     1             4      2        5      0         1        0        0
 8 b     1             3      4       21      0         1        0        1
 9 b     1             4      3        3      0         0        0        0
10 b     1             4      2        6      0         1        0        0
# ... with 4,860 more rows, and 56 more variables: occupspecific <dbl>,
#   occupbroad <dbl>, workinschool <dbl>, email <dbl>, computerskills <dbl>,
#   specialskills <dbl>, firstname <chr>, sex <chr>, race <chr>, h <dbl>,
#   l <dbl>, call <dbl>, city <chr>, kind <chr>, adid <dbl>, fracblack <dbl>,
#   fracwhite <dbl>, lmedhhinc <dbl>, fracdropout <dbl>, fraccolp <dbl>,
#   linc <dbl>, col <dbl>, expminreq <chr>, schoolreq <chr>, eoe <dbl>,
#   parent_sales <dbl>, parent_emp <dbl>, branch_sales <dbl>, ...
```

```
#It has 65 columns and 4870 rows
```

```
#number 2
ResumeNames %>%
  select(sex, race, firstname)
```

```
# A tibble: 4,870 x 3
   sex   race  firstname
   <chr> <chr> <chr>
 1 f     w     Allison
 2 f     w     Kristen
 3 f     b     Lakisha
 4 f     b     Latonya
 5 f     w     Carrie
 6 m     w     Jay
 7 f     w     Jill
 8 f     b     Kenya
```

```
 9 f      b      Latonya
10 m      b      Tyrone
# ... with 4,860 more rows
```

```
ResumeNames %>%
  mutate(female = case_when(
    sex == "f" ~ "TRUE"
  )) %>%
  mutate(black = case_when(
    race == "b" ~ "TRUE"
  )) %>%
  select(female, black)
```

```
# A tibble: 4,870 x 2
   female black
   <chr>  <chr>
 1 TRUE   <NA>
 2 TRUE   <NA>
 3 TRUE   TRUE
 4 TRUE   TRUE
 5 TRUE   <NA>
 6 <NA>   <NA>
 7 TRUE   <NA>
 8 TRUE   TRUE
 9 TRUE   TRUE
10 <NA>   TRUE
# ... with 4,860 more rows
```

Checking for Balance

The dataset is an example of a Randomized Control Trial (RCT). The authors randomly sent out fictitious "White sounding" and "Black sounding" resumes to employers in Boston and Chicago.

The first thing to check is whether the information mentioned in the resumes: `education`, `ofjobs`, `sex`, `computerskills` and `yearsexp` is similar across both sets of resumes. This is known as *Checking for Balance*, and I will refer to this in the context of the current experiment as checking for balance across race. Checking for balance is important to ensure everything else is held constant apart from the ethnicity of the applicant. That way any difference in call backs can be attributed entirely to discrimination.

The variable `computerskills` takes on the value `1` if a given resume says that the applicant has computer skills. The variables `education` and `yearsexp` indicate level of education and years experience, while `ofjobs` indicates the number of previous jobs listed on the resume.

You should answer all the relevant coding questions using `dplyr`.

Exercise #3

1. Read parts A-D of section II in BM and answer the following:

    - How did the experimenters create their bank of resumes for the experiment?
    - The experimenters classified the resumes into two groups. What were they and how did they make the classification?

- How did the experimenters generate identities for their fictitious job applicants?

2. Is sex balanced across race? Hint: what happens if you apply the function `sum` to a vector of `TRUE` and `FALSE` values?
3. Are computer skills balanced across race? Hint: the summary statistic you'll want to use is the *proportion* of individuals in each group with computer skills.
4. Compute the mean and standard deviation of `yearsexp` by race. Comment on your findings.
5. Are `education` and `ofjobs` balanced across race?
6. Is `education` balanced across `sex`? Does it matter whether it is balanced?

Solution to Exercise #3

Write your code and solutions here.

Number 1

A. The experimenter use the resumes of actual job searchers but they alter them to create distinct resumes. They done this by gather the resumes needed on two job search websites. To differ the datasets of resumes that they create with the real resumes job applicants, they eliminate the resumes sent out in the Boston and Chicago area. They also restrict four occupational categories : sales, administrative report, clerical services, and customer services. Not only that, they restrict resumes that has been posted for more than 6 months prior to the start of the experiment.

B. They classified the resumes into two groups : high and low quality resumes. To classify them, they use criteria such as labor market experience, career profile, existence of gaps in employment, and skills listed. And to make the gap further, they add to each high-qualityresumea subsetof the following features: summer or while-at-school employment experience, volunteeringexperience, extracomputerskills, certificationdegrees, foreign language skills, honors, or some military experience.

C. The researcher generated their fictitious job applicants by using some data from the real job applicants. The research itself aiming to unveil if there is a job discrimination across race for those who want to apply for a job. Thus, the impression of a name plays an important part. To give an impression of name, the researcher must differentiate between the white-sounding names and black-sounding names. In order to do that, the researcher use name from birth certificates of all babies that were born in Massachusetts between 1974 and 1979. Thus, they can create a new identities but based on a real name that doesn't distort the impression for the employer.

How they differentiate a white-sounding name from a black one? The researcher conduct a survey in various public areas in Chicago. Each of the respondent was given a name and asked to assess features of the person, one of which being race. In general, the names led respondents to readily attribute the expected race for the person but there were a few exceptions and these names were disregarded. Thus, by using these name, it will give the impression of the real white-sounding and black-sounding names for the employer.

```
#solution for number 2
ResumeNames %>%
  mutate(gender = case_when(
    sex == "f" ~ 1,
    sex == "m" ~ 1
  )) %>%
  group_by(race,sex) %>%
  summarize(total_sex_by_races = sum(gender))
```

```
`summarise()` has grouped output by 'race'. You can override using the `.groups` argument.
```

```
# A tibble: 4 x 3
# Groups:   race [2]
  race  sex   total_sex_by_races
  <chr> <chr>              <dbl>
1 b     f                   1886
2 b     m                    549
3 w     f                   1860
4 w     m                    575
```

```
#From this result, we can see that the white_male is only one third of the white_female population, so t
```

```
#solutions for number 3
ResumeNames %>%
  group_by(race) %>%
  summarize(total_comp_skills_by_race = sum(computerskills))
```

```
# A tibble: 2 x 2
  race  total_comp_skills_by_race
  <chr>                     <dbl>
1 b                          2027
2 w                          1969
```

```
#The computer skills between race are different. White race have fewer computer skills compared to black
```

```
#solutions for number 4
ResumeNames %>%
  group_by(race) %>%
  summarize(mean_yearsexp = mean(yearsexp), sd_yearsexp = sd(yearsexp))
```

```
# A tibble: 2 x 3
  race  mean_yearsexp sd_yearsexp
  <chr>         <dbl>       <dbl>
1 b              7.83        5.01
2 w              7.86        5.08
```

```
#From the table below, we can conclude that the mean years experience among the race seems pretty balan
```

```
#solutions for number 5
ResumeNames %>%
  group_by(race) %>%
  summarize(edu_by_race = sum(education), totaljobslisted_by_race = sum(ofjobs))
```

```
# A tibble: 2 x 3
  race  edu_by_race totaljobslisted_by_race
  <chr>       <dbl>                   <dbl>
1 b            8805                    8908
2 w            8817                    8923
```

```
#From the table below, we can conclude that education by race and total jobs listed by race is pretty b
```

Callbacks by Race and Sex

The outcome of interest in `ResumeNames` is `call` which takes on the value `1` if the corresponding resume elicits an email or telephone callback for an interview. Check your results in this section against Table 1 of the paper.

Exercise #4

1. Calculate the average callback rate for all resumes in `ResumeNames`.
2. Calculate the average callback rates separately for resumes with "white-sounding" and "black-sounding" names. What do your results suggest?
3. Repeat part 2, but calculate the average rates for each combination of race and sex. What do your results suggest?

Solution to Exercise #4

Write your code and solutions here.

```
#number 1 solutions
mean(ResumeNames$call)
```

```
[1] 0.08049281
```

```
#number 2 solutions
ResumeNames %>%
  mutate(names_by_races = case_when(
    race == "w" ~ "white-sounding names",
    race == "b" ~ "black-sounding names"
  )) %>%
  group_by(names_by_races) %>%
  summarize(callback_rates = mean(call))
```

```
# A tibble: 2 x 2
  names_by_races       callback_rates
  <chr>                         <dbl>
1 black-sounding names         0.0645
2 white-sounding names         0.0965
```

```
#From this calculation, we know that the average callback rate for white-sounding name was higher compa
```

```
#number 3 solutions (belom kelar)
ResumeNames %>%
  mutate(names_by_races = case_when(
    race == "w" ~ "white-sounding names",
    race == "b" ~ "black-sounding names"
  )) %>%
  group_by(names_by_races, sex) %>%
  summarize(callback_rates = mean(call))
```

```
`summarise()` has grouped output by 'names_by_races'. You can override using the `.groups` argument.
```

```
# A tibble: 4 x 3
# Groups:   names_by_races [2]
  names_by_races       sex   callback_rates
  <chr>                <chr>          <dbl>
1 black-sounding names f             0.0663
2 black-sounding names m             0.0583
3 white-sounding names f             0.0989
4 white-sounding names m             0.0887
```

```
#From the table below, we can see that female job applicants have higher callback rates compared to the
```

Comparing Returns to Quality

Bertrand and Mullainathan write: "for most of the employment ads we respond to, we send four different resumes: two higher-quality and two-lower quality ones." The column `h` takes on the value `1` if a resume as classified *a priori* as "high quality" and `0` if it was classified as "low quality." The columns `col`, `military`, `email`, `volunteer` are indicators for: college degree, has an email address, has done volunteer work, and served in the military.

Exercise #5

1. Compare the average value of `col`, `military`, `email`, and `volunteer` across "high quality" and "low quality" resumes. Discuss your findings.
2. Calculate average callback rates for black versus white-sounding names *separately* for "high-quality" and "low-quality" resumes. Discuss your findings

Solution to Exercise #5

Write your code and solutions here.

```
#Number 1
ResumeNames %>%
  mutate(quality = case_when(
    h == 1 ~ "high",
    l == 1 ~ "low",
  )) %>%
  group_by(quality) %>%
  summarize(mean_col=mean(col), mean_mil=mean(military), mean_email=mean(email), mean_volunteer=mean(vo
```

```
# A tibble: 2 x 5
  quality mean_col mean_mil mean_email mean_volunteer
  <chr>      <dbl>    <dbl>      <dbl>          <dbl>
1 high       0.724  0.190       0.924          0.792
2 low        0.715  0.00330     0.0305         0.0272
```

```
#From the table below, we can see that either high quality and low quality applicants have college degr
```

```
ResumeNames %>%
  mutate(names_by_race = case_when(
    race == "w" ~ "white-sounding names",
    race == "b" ~ "black-sounding names",
  )) %>%
  mutate(quality = case_when(
    h == 1 ~ "high",
    l == 1 ~ "low",
  )) %>%
group_by(names_by_race, quality) %>%
summarize(callbackrates = mean(call))
```

```
`summarise()` has grouped output by 'names_by_race'. You can override using the `.groups` argument.
```

```
# A tibble: 4 x 3
# Groups:   names_by_race [2]
  names_by_race        quality callbackrates
  <chr>                <chr>           <dbl>
1 black-sounding names high           0.0670
2 black-sounding names low            0.0619
3 white-sounding names high           0.108
4 white-sounding names low            0.0850
```

```
#From the table below, we can conclude that there is a lower effect for black-sounding names despite th
```