Problem Set 2. Experiments

Relevant material will be covered by **Sep 7**. Problem set is due **Sep 14**.

To complete the problem set, **Download the .Rmd** and complete the homework. Omit your name so we can have anonymous peer feedback. Compile to a PDF and submit the PDF on Canvas.

This problem set is based on:

Bertrand, M & Mullainathan, S. 2004. "Are Emily and Greg More Employable Than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination." American Economic Review 94(4):991–1013.

Here's a heads-up about what will be hard in this problem set

- for some, reading a social science paper will be hard
- for some, mathematical statistics will be hard
- for some, R coding will be hard

For almost no one will all three be easy.

> We want to support you to succeed! Text in this format is here to help you.

1. Conceptual questions about the study design

Read the first 10 pages of the paper (through the end of section 2). In this paper,

- the unit of analysis is a resume submitted to a job opening
- the treatment is the name at the top of the resume
- the outcome is whether the employer called or emailed back for an interview

1.1. (5 points) Fundamental Problem

One submitted resume had the name "Emily Baker." It yielded a callback. The same resume could have had the name "Lakisha Washington." Explain how the Fundamental Problem of Causal Inference applies to this case (1–2 sentences).

The Fundamental Problem of Casual Inference in this case is that we know that if we place Emily Baker on the resume yielded a callback, but on the other hand we don't know what would have happened if the name Lakisha Washington was placed on that same resume

1.2. (5 points) Exchangeability

In a sentence, state what exchangeability means in this study. For concreteness, for this question you may suppose that the only names in the study were "Emily Baker" and "Lakisha Washington." Be sure to explicitly state the treatment and the potential outcomes.

Exchangeability in this study means that the potential outcomes of each of the treatments are independent of each other. Using the names Emily Baker and Lakisha Washington as the examples the outcome of using either of these names on the same resume will be independent from each other and wouldn't cause any inherit bias in the probabilities as to which condition occurs which in this case would be getting a callback or not.

1.3. (10 points) Something you liked

State something concrete that you appreciate about the study design, other than randomization.

Ilike how many restrictions where used along side the randomization process to ensure that data collected wouldn't be as skewd if the left muliple variables to inter play with the experiment.

2. Analyzing the experimental data

Load packages that our code will use.

```
library(tidyverse)
```

```
FALSE Warning: package 'tidyverse' was built under R version 4.2.3

FALSE Warning: package 'ggplot2' was built under R version 4.2.3

FALSE Warning: package 'tibble' was built under R version 4.2.3

FALSE Warning: package 'purrr' was built under R version 4.2.3

FALSE Warning: package 'dplyr' was built under R version 4.2.3

FALSE Warning: package 'stringr' was built under R version 4.2.3
```

```
library(haven)
```

```
FALSE Warning: package 'haven' was built under R version 4.2.3
```

Download the study's data from OpenICPSR: https://www.openicpsr.org/openicpsr/project/116023/version/V1/view. This will require creating an account and agreeing to terms for using the data ethically. Put the data in the folder on your computer where this .Rmd is located. Read the data into R using `read_dta`.

```
d <- read_dta("lakisha_aer.dta")
```

> If you have an error, you might need to set your working directory first. This tells R where to look for data files. At the top of RStudio, click Session -> Set Working Directory -> To Source File Location.

You will now see `d` in your Global Environment at the top right of RStudio.

We will use four variables:

Name	Role	Values
`call`	outcome	1 if resume submission yielded a callback 0 if not
`firstname`	treatment	first name randomly assigned to resume
`race`	category of treatments	`b` if first name signals Black

Name	Role	Values
`sex`	category of treatments	`w` if first name signals white `f` if first name signals female `m` if first name signals male

For 2.1–2.4, we will think of `race` as the treatment. For 2.5–2.6, we will think of `firstname` as the treatment.

Restrict to these variables using `select()`.

```
d_selected <- d %>%
  select(call, firstname, race, sex)
```

If you are new to R, here is what just happened:

- created a new object `d_selected`
- used the assignment operator `<-` to put something in that object
- we started with our data object `d`
- we used the pipe operator `%>%` to hand `d` down into a new action
- the action `select()` selected only the variables of interest

We will often analyze data by starting with a data object and handing that through a series of actions connected by the pipe `%>%`

2.1. (5 points) Point estimates of expected potential outcomes

The top of Table 1 reports callback rates: 9.65% for white names and 6.45% for Black names. Reproduce those numbers. To do so, take the code below but add a `group_by()` action between `d_selected` and `summarize`.

Here's a reference that introduces `group_by` and `summarize`.

Answer. (modify this code)

```
d_summarized <- d_selected|>
  group_by(race)|>
  summarize(callback_rate = mean(call),
            number_cases = n()) %>%
  print()
```

```
## # A tibble: 2 x 3
##   race  callback_rate number_cases
##   <chr>         <dbl>        <int>
## 1 b            0.0645         2435
## 2 w            0.0965         2435
```

2.2. (5 points) Inference for expected potential outcomes

Use `mutate()` (see reference page) to create a new columns containing the standard error of each estimate as well as lower and upper limits of 95% confidence intervals.

To make this easier, here is a quick math review and R functions you can use.

Standard error in math. Let Y^a be a Bernoulli random variable, taking the value 1 if a random resume with name a yields a callback and 0 otherwise. Let $\pi^a = P(Y^a = 1)$ be the probability of a callback. From statistics, you know this has variance $V(Y^a) = \pi^a(1-\pi^a)$. We have estimated by an average: $\hat{\pi}^a = \frac{1}{n_a}\sum_{i:A_i=a} Y_i^a$. If we did this many times in many hypothetical samples, we would not always get the same estimate. In fact, our estimate would have sampling variance $V(\hat{\pi}^a) = \frac{\pi^a(1-\pi^a)}{n_a}$. We know this because $\hat{\pi}^a$ is a mean of n_a independent and identically distributed random variables Y^a. The standard error is the square root of the sampling variance: $SE(\hat{\pi}^a) = \sqrt{\frac{\pi^a(1-\pi^a)}{n_a}}$. We can estimate this standard error by plugging in our estimate $\hat{\pi}^a$ for the true unknown π^a wherever it appears.

Standard error in code. We translated the standard error formula into code for you below. This function accepts an estimated probability `p` and sample size `n` and returns the estimated standard error. You can use this `se_binary()` function in your code within `mutate()` just like how `mean()` was used within `summarize()` at the start of the problem set.

```
se_binary <- function(p, n) {
  se <- sqrt( p * (1 - p) / n )
  return(se)}

d_summarized<-d_summarized|>
  mutate(
    se=se_binary(callback_rate,number_cases)
  )
d_summarized
```

```
## # A tibble: 2 x 4
##   race  callback_rate number_cases      se
##   <chr>         <dbl>        <int>   <dbl>
## 1 b            0.0645         2435 0.00498
## 2 w            0.0965         2435 0.00598
```

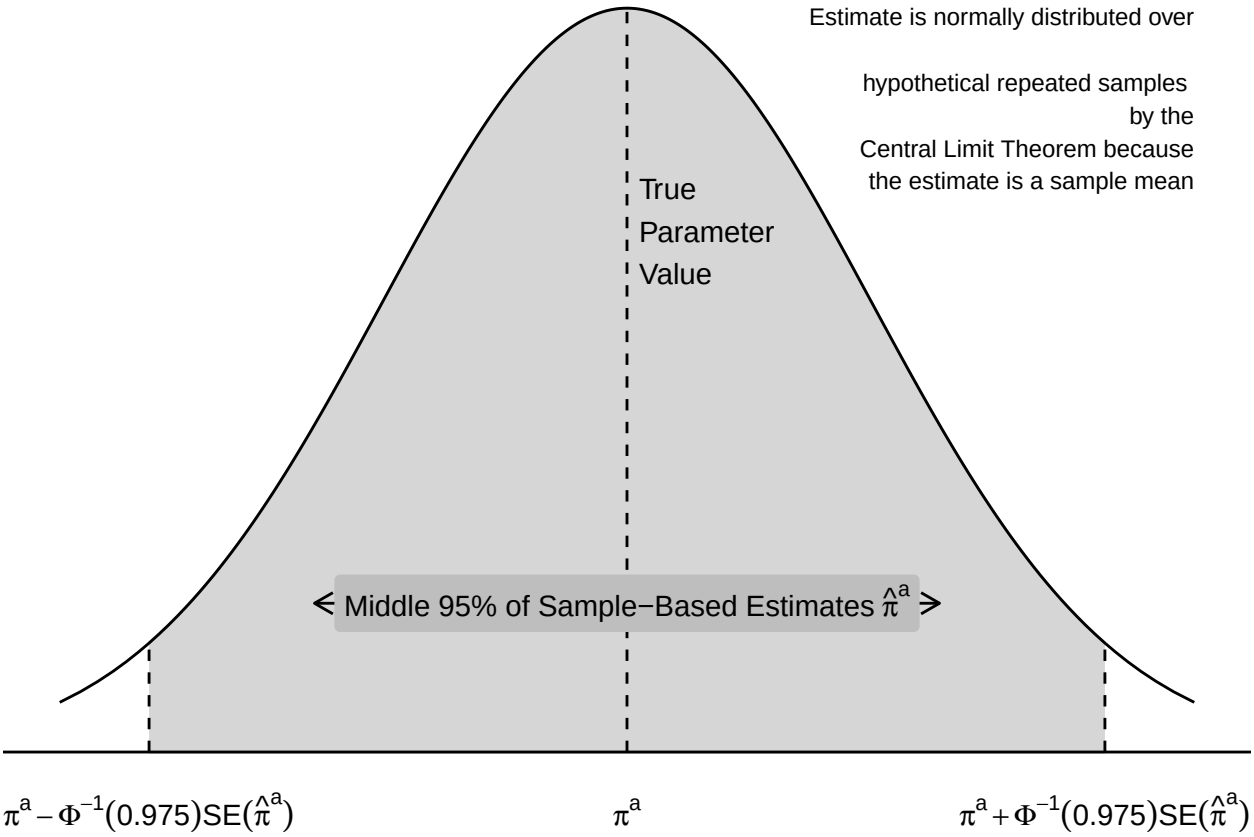
Sampling distribution in math. Because $\hat{\pi}^a$ is a sample mean, we know something about its sampling distribution: in the limit as the sample size grows to infinity, across hypothetical repeated samples the distribution of $\hat{\pi}^a$ estimates becomes Normal. This is by the Central Limit Theorem! Across repeated samples, the middle 95% of estimates will fall within a known range: $\pi^a \pm \Phi^{-1}(.975) \times SE(\hat{\pi}^a)$, where $\Phi^{-1}()$ is the inverse cumulative distribution function of the standard Normal distribution. You might have previously learned that $\Phi^{-1}(.975) \approx 1.96$, so what might be familiar to you is the number 1.96.

Sampling distribution in a graph.

```
#|echo: F
#|fig.height: 2
normal_curve <- data.frame(x = qnorm(seq(.01,.99,.0025))) %>%
  mutate(f = dnorm(x))

normal_curve %>%
  ggplot(aes(x = x, y = f)) +
  geom_ribbon(data = normal_curve %>%
```

```
              filter(x >= qnorm(.025) & x <= qnorm(.975)),
            aes(ymin = 0, ymax = f),
            color = "gray", alpha = .2) +
  geom_line() +
  geom_hline(yintercept = 0) +
  annotate(geom = "segment",
           x = qnorm(c(.025,.5,.975)),
           xend = qnorm(c(.025,.5,.975)),
           y = 0, yend = dnorm(qnorm(c(.025,.5,.975))),
           linetype = "dashed") +
  annotate(geom = "text", hjust = 0,
           x = 0.05, y = .7 * dnorm(0),
           label = "True\nParameter\nValue") +
  annotate(geom = "segment",
           x = qnorm(.1), xend = qnorm(.9),
           y = .2 * dnorm(0), yend = .2 * dnorm(0),
           arrow = arrow(ends = "both", length = unit(8,"pt"))) +
  annotate(geom = "label",
           x = 0, y = .2*dnorm(0),
           label = "'Middle 95% of Sample-Based Estimates'~hat(pi)^a",
           parse = T,
           fill = "gray", label.size = NA) +
  annotate(geom = "text", label = "Estimate is normally distributed over
  \nhypothetical repeated samples
by the\nCentral Limit Theorem because\nthe estimate is a sample mean",
           size = 3,
           x = qnorm(.99), y = dnorm(qnorm(.5)),
           hjust = 1, vjust = 1) +
  scale_x_continuous(breaks = c(-qnorm(.975),0,qnorm(.975)),
                     labels = c(expression(pi^a - {Phi^{-1}} (.975)*SE(hat(pi)^a),
                                           pi^a,
                                           pi^a + {Phi^{-1}} (.975)*SE(hat(pi)^a)))) +
  theme_void() +
  theme(axis.text.x = element_text(color = "black"))
```



> **Confidence interval in math.** We get a 95% confidence interval by plugging in the estimates $\hat{\pi}^a$ and $\widehat{SE}(\hat{\pi}^a)$ to the limits above. This interval is centered on the estimate $\hat{\pi}^a$ and has a nice property: if we repeatedly made a confidence interval by this procedure using hypothetical samples from the population, our interval would contain the unknown true parameter π^a 95% of the time.

> **Confidence interval in code.** We translated the confidence interval formula into code for you below. These functions accept an estimate and standard error and return the lower and upper bounds (respectively) of a 95% confidence interval that assumes a Normal sampling distribution. You can use these functions in your code within `mutate()` just like how `mean()` was used within `summarize()` at the start of the problem set.

```
ci_lower <- function(estimate, standard_error) {
  estimate - qnorm(.975) * standard_error
}
ci_upper <- function(estimate, standard_error) {
  estimate + qnorm(.975) * standard_error
}
```

Answer.

```
d_summarized<-d_summarized|>
  mutate(
    uper_ci=ci_upper(callback_rate,se),
    lower_ci=ci_lower(callback_rate,se)
  )

d_summarized
```

```
## # A tibble: 2 x 6
##   race  callback_rate number_cases      se uper_ci lower_ci
##   <chr>         <dbl>        <int>   <dbl>   <dbl>    <dbl>
## 1 b            0.0645         2435 0.00498  0.0742   0.0547
## 2 w            0.0965         2435 0.00598  0.108    0.0848
```

2.3. (5 points) Interpret your confidence interval

In words, interpret the confidence intervals. Be sure to discuss what their property is over hypothetical repeated samples, and be sure to frame your answer using the numbers and variables from the actual experiment we are analyzing.

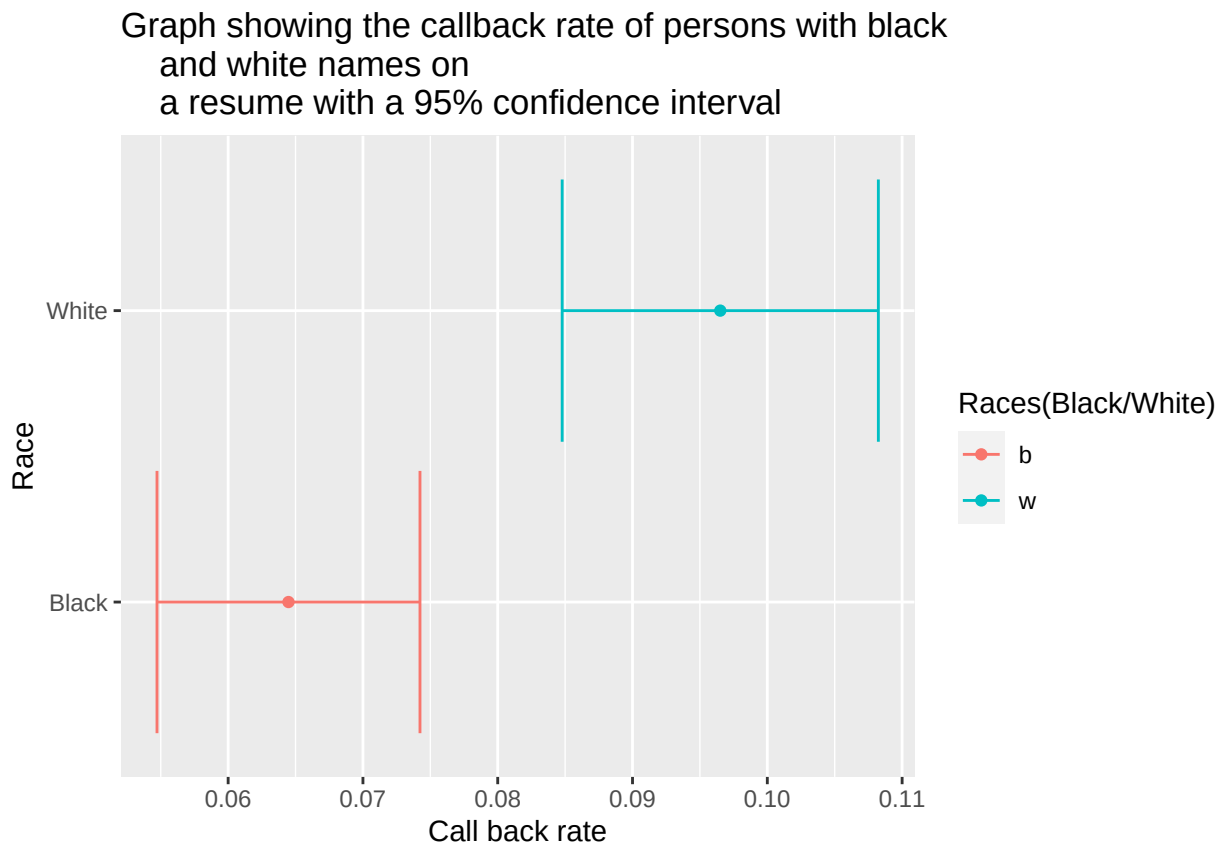
A confidence interval tells us at what percentage the range of the values we expect to see appear. The call back rate for persons with black names is 0.064476386036961 while the callback rate for persons with white names is 0.0965092402464066, both values that appeared in from the actual experiement. When looking at them we clearing see that both of these values fall into their respective 95% confidence interval.

2.4. (5 points) Visualize expected potential outcomes

Using `ggplot()`, visualize the estimated callback rate by race. Use `geom_point()` for point estimates and `geom_errorbar()` for confidence intervals, with race on the `x` axis and estimates on the `y` axis. Label the axes using full words.

If you have never used `ggplot`, see Ch 3 of R for Data Science by Hadley Wickham.

```
d_summarized|>
  ggplot(aes(x=callback_rate,y=race,color=race))+geom_point()+
  geom_errorbar(aes(xmin=lower_ci,xmax=uper_ci))+
  labs(
    title = 'Graph showing the callback rate of persons with black
    and white names on
    a resume with a 95% confidence interval',
    x='Call back rate',
    y='Race',
    color="Races(Black/White)"
  )+
  scale_y_discrete(labels=c("Black",'White'))
```



2.5. (5 points) Estimate and visualize by firstname

Do distinct first names yield distinct effects? Repeat 2.2–2.4, but now create estimates grouped by `race`, `sex`, and `firstname`. Visualize point estimates and confidence intervals.

> One way to visualize is by placing first names on the x-axis and using a `facet_wrap()` layer to facet over race and sex. Any strategy to visualize is fine, as long as it shows estimates for each `firstname` and indicates the `race` and `sex` signaled by that `firstname`

```
#|echo: FALSE
d_newsum<-d_selected|>
  group_by(race,sex,firstname)|>
  summarize(callback_rate = mean(call),
            number_cases = n())
```

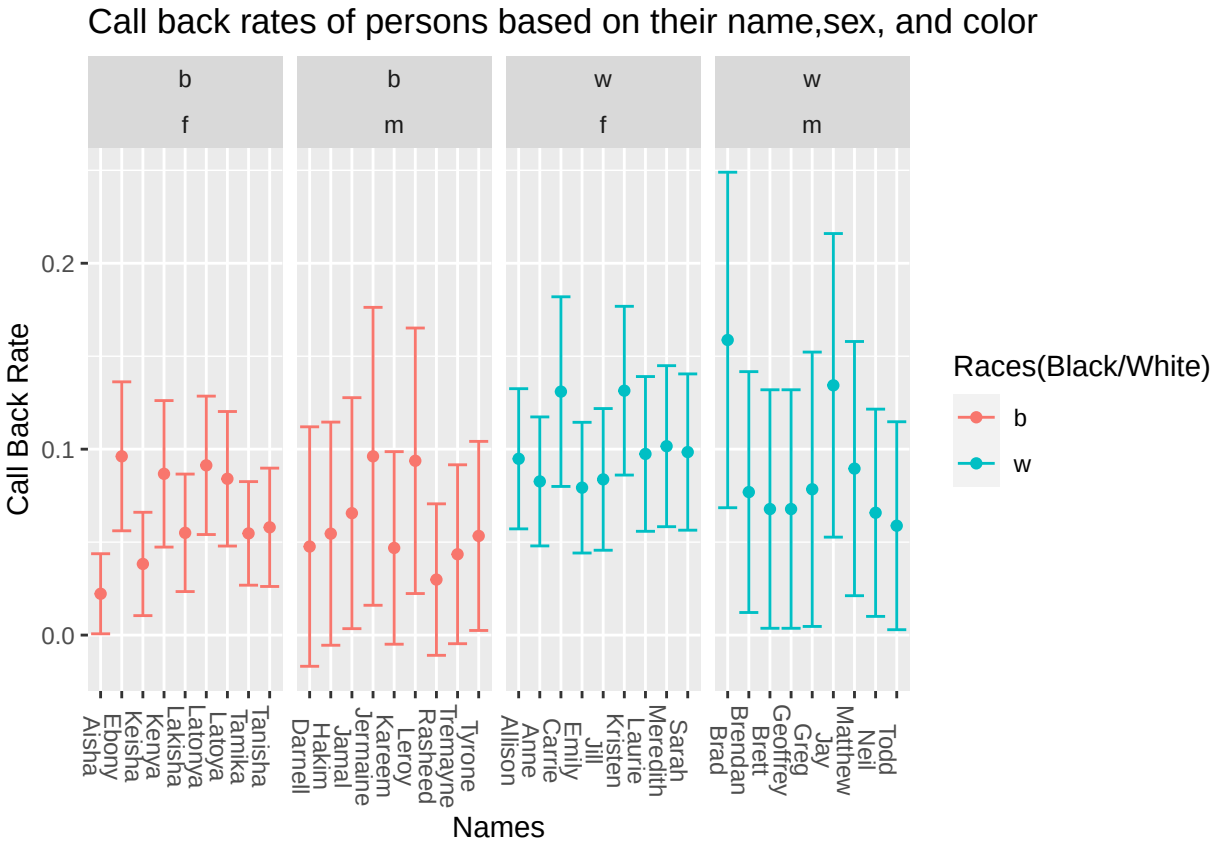
```
## `summarise()` has grouped output by 'race', 'sex'. You can override using the
## `.groups` argument.
```

```
d_newsum<-d_newsum|>
  mutate(
    se=se_binary(callback_rate,number_cases)
  )
```

```r
d_newsum<-d_newsum|>
  mutate(
    uper_ci=ci_upper(callback_rate,se),
    lower_ci=ci_lower(callback_rate,se)
  )
d_newsum
```

```
## # A tibble: 36 x 8
## # Groups:   race, sex [4]
##    race  sex   firstname callback_rate number_cases     se uper_ci  lower_ci
##    <chr> <chr> <chr>             <dbl>        <int>  <dbl>   <dbl>     <dbl>
##  1 b     f     Aisha            0.0222          180 0.0110  0.0438  0.000688
##  2 b     f     Ebony            0.0962          208 0.0204  0.136   0.0561
##  3 b     f     Keisha           0.0383          183 0.0142  0.0660  0.0105
##  4 b     f     Kenya            0.0867          196 0.0201  0.126   0.0473
##  5 b     f     Lakisha          0.055           200 0.0161  0.0866  0.0234
##  6 b     f     Latonya          0.0913          230 0.0190  0.129   0.0541
##  7 b     f     Latoya           0.0841          226 0.0185  0.120   0.0479
##  8 b     f     Tamika           0.0547          256 0.0142  0.0825  0.0268
##  9 b     f     Tanisha          0.0580          207 0.0162  0.0898  0.0261
## 10 b     m     Darnell          0.0476           42 0.0329  0.112  -0.0168
## # i 26 more rows
```

```r
d_newsum|>
  ggplot(aes(x=firstname,y=callback_rate,color=race))+
  geom_point()+
  facet_wrap(vars(race,sex),nrow=1,scales = "free_x")+
  geom_errorbar(aes(ymin=lower_ci,ymax=uper_ci))+
  theme(
    axis.text.x = element_text(angle = 270)
  )+
  labs(
    color="Races(Black/White)",
    y='Call Back Rate',
    x='Names',
    title = 'Call back rates of persons based on their name,sex, and color'
  )
```



2.6. (5 points) Interpret

Within race and sex, not all first names have the same effect. Suppose these are true differences (not due to sampling variability). What does this tell you about the importance of researcher decisions about which names to use as treatments?

What this tells us is that if a black sounding is used for a resume for a person whose race is black will have drastically different results than if it was a white sounding name attached to a persons resume if they themselves where white. By mixing and matching names such as putting white sounding name to black persons or putting a black sounding name to a white person should produce interesting results