Problem Set 3

Due Monday September 25, 2023, 10am

To successfully complete this problem set, please follow these steps:

1. Download the contents of the assignment Dropbox link in a folder (directory) on your computer designated for this problem set only. Follow the directory structure, e.g., putting the `data` folder containing the datasets as a subdirectory of the project directory.

2. Insert your answers in the yellow boxes using Microsoft Word, and prepare a single `.R` script for what you produce. Save the word document as a `.PDF`.

3. Please submit the PDF to the designated `PS-XX: pdf` link and your R Script to the `PS-XX: R` link.

(1) Your name:

Nikhilla Bhuvana Sundar

(2) Group members, if any:

(3) Compliance with the Academic Code on problem set[1] (sign with an X below)

X

[1] You may use the same code from classmates, Ed Discussion Board, instructors, and generative AI. However, you must hand in your own unique written work and code in all cases. Any copy/paste of another's work is plagiarism. In other words, you can work with your classmate(s), sitting side-by-side and going through the problem set question-by-question, or use generative AI to provide potential code for you, but you must each type your own answers and your own code.

The first two problems will draw from this article on immigration, which you should skim first and read more closely for discussion on 9/25.

> Malhotra, Neil, Margalit, Yotam. and Mo, Cecilia H. (2013), "Economic Explanations for Opposition to Immigration: Distinguishing between Prevalence and Conditional Impact". *American Journal of Political Science*

Please complete the following pre-class exercise by 9/25 10am, the same deadline as this problem set:

https://yale.instructure.com/courses/89156/quizzes/62870

(it only asks for you one question and has a reading guide at the top).

Gradescope will have pre-installed the following packages:

```
tidyverse
gt
haven
urbnmapr
sf
```

Problem 1: Descriptive Statistics

The dataset is called `mmm_replication.dta` and should be read in by the `read_dta` which is a function in the `haven` dataset for Stata files. Assign it to the object `mmm`. The dataset includes the following variables:

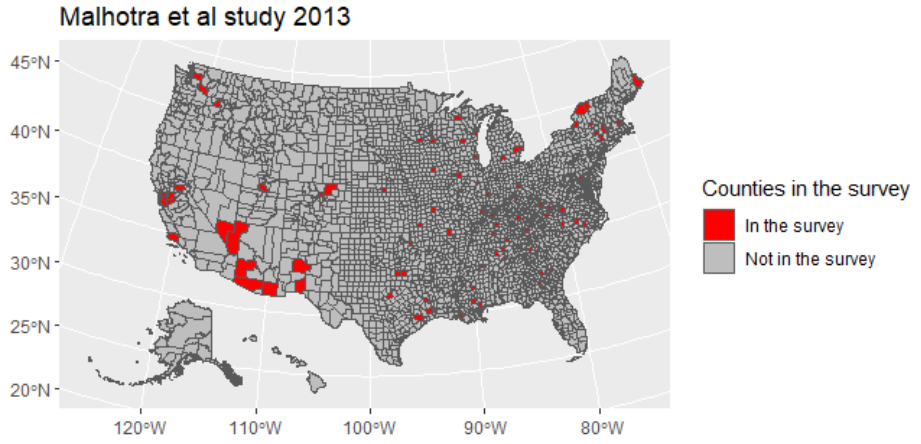
Variable name	Description
caseid	Unique Subject ID
zipcode	Zip Code of Respondent
techzip2	Binary variable for High-Technology Zip Codes in county (corresponds to "High-Technology County" in Table 1)
county_fips	County FIPS Code
weightvec	Sampling Weight
employed	Binary variable for employed (full-time, part-time, self-employed)
techwork	Binary variable for work in technology industry, including any job in Engineering, Computer related design, etc.
gender	Binary variable for female
age	Age, rescaled to 0 - 1 following footnote 26
maritalstatus	Marital Status (e.g., divorced)

Variable name	Description
income	Household income, rescaled to 0 - 1 following footnote 26
race	Race of Respondent (does not include Indian or Indian Americans, per p. 397 of the article)
education	Education Level of Respondent
financespecific	More fine grained information on industry type: Finance, Accounting, and Consulting
realestatespecific	More fine grained information on industry type: Real Estate, Rental, and Leasing
lawspecific	More fine grained information on industry type: Law and Legal Services
keepjob	Confidence About Keeping Job Over Next 3 Years
h1bvisas	Response to: Increase/Decrease H1-B Visas?
indianimmig	Response to: Increase/Decrease number of immigrants from India allowed in US?
threatened	Response to: How threatened American way of life is by foreign influence?
partyid	Party Affiliation and Strength
dscore	IAT D Score, rescaled according to footnote 25

1.1

Targeted selection of the population is a strength of this research. After skimming this part of the paper, create a map of US counties that colors the counties in the MMM dataset (e.g., in red).

For your shapefile, use the shapefile of counties provided by the `urbnmapr::get_urbn_map`. `urbnmapr` is a R package on Github (https://github.com/UrbanInstitute/urbnmapr). See tips.



1.2

Using `gt`, Create a table of proportions that summarizes the main finding in this paper. It should look identical to, or very close to the following:

	H1B attitude					
	Decrease a great deal	Decrease a little	Neither increase nor decrease	Increase a little	Increase a great deal	n
Tech worker	0.40	0.21	0.26	0.10	0.03	58
Other white collar	0.32	0.16	0.32	0.05	0.14	56
All other workers	0.30	0.22	0.32	0.11	0.05	472
Unemployed	0.31	0.24	0.28	0.10	0.07	527
Note: Cells show sample proportion, so values in each row sum to 1						

This table replicates the first group of barplots in Figure 2(a) in table form. It shows, for example, that 3% of tech workers (n = 58) support increasing H1B visas “a great deal”.

This table is formatted in the following way:

- **Columns**: the values of the H1B visa question. This variable is Stata's "labelled" variable that are numbers with labels attached to them. To display the labels, change them into their labels by the `haven::as_factor()` function. The last column indicates the sample size of each employment group.
- **Rows**: The four groups of employment that are shown in Figure 2(a). Using the `case_when` function within `mutate`, you will want to create a variable called `emplopymentgroup` that takes on four possible character values:
    i. Tech worker: employed, and working in the tech sector (`techwork == 1`)
    ii. Other white collar: employed, and working in a white collar high-skilled sector that is not tech: what the authors call the placebo occupational category. This should be created by combining information about the variables `lawspecific`, `financespecific`, `realestatespecific`, and `techwork` using the `&` and `|` operators. The first three variables are coded so that rows with missing values indicate not following into those categories. The `techwork` variable is coded so that 0 means not being in tech work.
    iii. All other workers: employed in sectors that are not white collar.
    iv. Unemployed: `employed == 0`.
- **Cells**: Each cell should show the proportion of responses in each of the four employment groups. In other words, the table should show the breakdown of the question response in each `employmentgroup`.
- **Spanner:** It is visually helpful to have a column name that spans multiple columns, as in the "H1B attitude" label in the screenshot. Add it by the `tab_spanner` function in gt (https://gt.rstudio.com/reference/tab_spanner.html**)**.

	H1B Attitude					
	Decrease a great deal	Decrease a little	Neither increase nor decrease	Increase a little	Increase a great deal	n
All other workers	0.30	0.22	0.32	0.11	0.05	472
Other White Collar	0.32	0.16	0.32	0.05	0.14	56
Tech Worker	0.40	0.21	0.26	0.10	0.03	58
Unemployed	0.31	0.24	0.28	0.10	0.07	527

Problem 2: Operationalization and Regression

We will now replicate the regression model in *column 3* of Table 1 of the paper. Replication means that your results should be the same as the published paper. This requires that we convert the survey categories into numeric values – a process we often will call "operationalize" in social science. Operationakization will happen in three fronts:

a. The authors rescale their *outcome* variable into a numeric score from 0 to 1 (p. 397, second column). They assume each option is equidistant: That is, if there are five ordered outcomes, they should be coded 0, 0.25, 0.5, 0.75, 1. For example for the `h1bvisas` variable, we recode:

Original value	Label	Recode to
1	[Decrease a great deal]	0
2	[Decrease a little]	0.25
3	[Neither increase nor decrease]	0.5
4	[Increase a little]	0.75
5	[Increase a great deal]	1

Create a variable called `h1bvisas_scaled` that corresponds to this recoded value. You can use `case_when()` here, but there is also a shorter one-line transformation that takes advantage of the numerical structure of the original value.

b. The right-hand side variables are operationalized with the following description in the paper:

"Control variables are coded as follows: (1) gender (*1 = female*); (2) age; (3) marital status (*1 = married*); (4) education level (*0 = not completed high school education, 1 = high school graduate, 2 = some college, 3 = two-year college degree, 4 = bachelor's degree, 5 = postgraduate degree*); (5) whether the respondent identifies himself or herself as white; (6) income (*0 = below $30,000, 1 = $30,000–40,000, 2 = $40,000–50,000, 3 = $50,000–60,000, 4 = $60,000–75,000, 5 = $75,000–90,000, 6 = $90,000–110,000, 7 = $110,000–130,000, 8 = $130,000–150,000, 9 = above $150,000*); and (7) party identification (*0 = strong Democrat, 1 = not-strong Democrat, 2 = lean Democrat, 3 = lean Republican, 4 = not-strong Republican, 5 = strong Republican*). All variables were recoded to lie between 0 and 1."

We have done the recoding for you for the variables `age`, `income`, and `dscore`, but you will need to do the recoding for `pid`, `educ`, marital status, and whether the respondent identifies as White.

c. Finally, specify the baseline level (or the "omitted category") of the categorical variable `employmentgroup` that you made in Problem 1.2 to the same baseline in the paper (all other workers).

You do not need to use `modelsummary` or make the regression table human-readable in this part of the problem. Simply screenshot the output of the `summary` function for the linear regression.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.51	0.06	9.14	0.00
employmentgroupOther White Collar	0.01	0.04	0.19	0.85
employmentgroupTech Worker	-0.10	0.04	-2.49	0.01
employmentgroupUnemployed	0.03	0.02	1.31	0.19
dscore	-0.12	0.06	-2.17	0.03
gender	-0.05	0.02	-2.45	0.01
age	-0.77	0.17	-4.66	0.00
I(age^2)	0.72	0.20	3.67	0.00
marital_scaled1	0.01	0.02	0.59	0.56
educ_scaled	0.20	0.03	5.85	0.00
race_scaled1	-0.04	0.02	-1.67	0.10
income	0.14	0.03	4.03	0.00
pid_scaled	-0.11	0.03	-4.00	0.00
techzip2	0.00	0.02	0.16	0.87

Problem 3: Z-scores

Let's return to the Banerjee et al. data and start recreating the standardized index. This problem corresponds to the part of the paper where they write:

> "We construct indices first by defining each outcome Y_{ijl}^{k} (outcome k, for observation i in family j, within country l) so that higher values correspond to better outcomes. Then we standardize each outcome into a z-score, by subtracting the country control group mean at the corresponding survey round and dividing by the country l's control group standard deviation (SD) at the corresponding survey round."

In this problem set, we will only start with the experiment in Pakistan and Peru and focus only on per capita total consumption in the household `ctotal_pcmonth_`. The variable descriptions are the same as Problem Set 2. The dataset is `graduation_pakistan-peru.dta` in the data folder, which you should read in by the `read_dta()` function.

3.1

Report the sample mean and sample standard deviation of the consumption variable in endline 2 (`ctotal_pcmonth_fup`), among the control group in each of the two countries in this dataset. You will need to use this dataset for subsequent problems, so assign the 2 by 3 tibble into its own object. `summarize()` should only appear once in your code here.

Endline 2 consumption variable summary:

The sample mean and standard deviation for **Pakistan's control group** is \$85.03 and \$50.87 respectively.

Sample mean and standard deviation for **Peru's control group** is \$155.55 and \$84.43 respectively

3.2

Following the paper, standardize the outcome into a Z-score with the mean and standard deviation of the values you just estimated. That is,

$$Y_{il}^{*} = \frac{Y_{il} - \bar{Y}_{0l}}{s_{0l}}$$

where Y_{il}^{*} is the standardized Z-score and Y_{il} is the unstandardized consumption value for household i in country l; $\bar{Y}_{0l}$ is the sample mean of the variable in the control group of country l, and s_{0l} is the sample standard deviation of the outcome also in the control group of country l. The indices k, j of the paper are not shown here since we are currently only analyzing one family of variables.

After standardizing, how can we interpret, for example, a value of $Y_{il}^{*} = 0.5$?

A standardized consumption value (Z score) of 0.5 for household i among the control group in country l means that they are 0.5 standard deviation away from mean consumption value of all households in the control group. In other words, this household's consumption value is higher than the group's average consumption

3.3

We can now start replicating the first coefficient in Figure 3 of the Banerjee et al. paper. Run regressions that estimating the causal effect of assignment to the Graduation program on (standardized) endline 2 per capita consumption.

You do not need to present a clean table for this question, but instead just report the rounded coefficient estimate for the average treatment effect for the following three specifications:

1. the regression outcome from a simple regression regressing the outcome on assignment.
2. same as 1 but adds the baseline consumption (`ctotal_pcmonth_bsl`) as a control (the Z variable in Banerjee et al. equation 1), and
3. same as 2 but adds a binary variable indicating whether the country is Peru (a "country fixed effect")

Finally, substantively interpret the magnitude of the treatment effects in proper units.

1. On average, households that were assigned to the graduation program had a $0.12 standard deviation higher mean per capita consumption when compared to the control group.
2. When controlled for consumption at baseline, those who received the treatment have a $0.1 standard deviation higher average per capita consumption when compared to the control group
3. When controlled for consumption at baseline, those who received the treatment in Peru have a $0.12 standard deviation higher per capita consumption than Pakistan. However, this result is not statistically significant

Tips

1.1

This problem is very similar to the world map assignment in problem set 1. The main differences are: (a) the shapefile comes from a package that is not on the official R package repository, CRAN, and (b) there are too many counties to list manually.

- `urbnmapr` is not uploaded to the the official R package repository. Therefore, it cannot be installed by the usual `pak::pkg_install("urbnmapr")` or `install.packages()` that we have been using so far. Instead, tell `pkg_install` to obtain the package from the Github, like so: `pkg_install("UrbanInstitute/urbnmapr")`.
- Like the past attempts, will need to load the `sf` package.
- Counties and other Census geographies in the US are given numeric identifiers called FIPS codes (`county_fips`). Use these as the bridge between the two datasets.

Ultimately, you want a county shapefile that has a column indicating whether or not to color the county. There are two possible ways to do this: joining and using `%in%`.

- When joining shapefiles, the order does matter: the first dataset needs to be the shapefile, and the non-GIS rectangular dataset comes as the second dataset. You will want to retain all counties, not just the ones that are in MMM, so you will need to use `left_join` instead of `inner_join`. See https://dplyr.tidyverse.org/reference/mutate-joins.html.
- When you join the two dataset, remember that the goal is to create a variable in the county shapefile that is an indicator for whether or not the county is in the survey dataset. That means you will want to make a variable beforehand in the MMM data.
- There is another way to get this indicator variable without joins, although joins are a more general tool that is always useful. The alternative way is to come up with a vector of county FIPS codes that are in the MMM data, and then create a new variable in the shapefile that is 1 or 0. To get an indicator of whether or a not values of a variable are in the data, we use the base-R operator `%in%` , as in `mutate(shapefil_fips %in% mmm_fips)` where `mmm_fips` is a vector of unique county FIPS codes.

1.2

- Getting the appropriate fraction is similar to PS-01 3.4, where you used mutate() combined with group_by().
- Labelled variables are used frequently in Stata and SPSS to encode both a number and a label to a data point. This is similar to R's `factor` but slightly different in that labelled variables are treated as numbers if operated on with numerical operators. The `haven` package in R handles between these formats. To see how this transformation occurs, see for example the output of this code:

```
mmm |>
  distinct(h1bvisas) |>
  arrange(h1bvisas) |>
  mutate(numeric = as.numeric(h1bvisas),
```

```
          factor = as_factor(h1bvisas))

# # A tibble: 5 × 3
#   h1bvisas                          numeric factor
#   <dbl+lbl>                           <dbl> <fct>
# 1 1 [Decrease a great deal]               1 Decrease a great deal
# 2 2 [Decrease a little]                   2 Decrease a little
# 3 3 [Neither increase nor decrease]       3 Neither increase nor decrease
# 4 4 [Increase a little]                   4 Increase a little
# 5 5 [Increase a great deal]               5 Increase a great deal
```

The first row shows, for example, that the first value of `h1bvisas` is a number, 1, with the label "Decrease a great deal". Numbers are called doubles in computer science and therefore the column is labelled "dbl" in the R output. Applying as.numeric() to it is called *coercision*: it changes the type of variable to a number, stripping off the label. Applying as_factor() (not as.factor) to it will coerce the labelled variable to a factor. Factors and labelled variables both have orderings, but you can perform numeric operations on labelled variables. For example:

```
mmm |>
  distinct(h1bvisas) |>
  arrange(h1bvisas) |>
  filter(h1bvisas <= 2)
```

- case_when (https://dplyr.tidyverse.org/reference/case_when.html) is a general dplyr function for recoding complex values. See the examples and note that within each line of case_when separated by commas, there is a formula representation where the condition is in the left hand side and the value to recode to is in the right hand side. Use the operators & (and), | (or), == (equals), or != (not equals). For example, a white collar worker it i
- To make the white collar variable, look at the one-way tabulation. of each dataset (e.g. mmm |> count(lawspecific) to see the possible values. You want to define a set of operations that together are TRUE when the white color job categories such as lawspecific is non-missing (!is.na(lawspecific)), but techwork is also not 1 (since techwork is usually considered white collar, but it is important for this paper to distinguish tech white collar vs. non-tech white collar).

Problem 2

- Your regression specification should include all the terms which have coefficients on Table 1 column 3.
- Just like we reordered factor levels to fix the ordering of value labels in ggplot, we will use the same function — fct_relevel — to set the baseline level.

- Two estimate a squared term on age, you can either create a separate variable beforehand that is age^2 and use that separately. To do this in the lm function internally, you will need to wrap the squares into the I() function, e.g. lm(y ~ x + I(x^2)) instead of lm(y ~ x + x^2).

3.1

- Remember that the authors only use the data from the control group subset when computing the mean and standard deviation, so that the all subsequent standardized units can be interpreted as units for the (untouched) control group.

- Like in the past, beware of missing values and compute the mean and standard deviations without them.

3.2

- An important feature of this standardization is that the values are centered and rescaled based on the summary statistics of the *same country*. For example, at endline 2, the average consumption in Peru is twice as large as that in Pakistan. Subtracting off the mean changes the resulting variable so that a 0 means the country average.
- This implies that when you do the standardization within `mutate`, you need to be subtracting and dividing by *country-specific* statistics. A good way to ensure that is to *left join* the appropriate statistic so that there is a column for mean and column for sd, and if household `i` belongs to country `l`, then the mean and sd is also for country `l`.

3.3

- `coef` and `summary` are the basic commands to summarize regression objects. `modelsummary` is used for formatting them in a presentable table.
- R formats very small and very large numbers in formats like `-1.337e-15`, `1.337e-01`, or `1.337e+10`. This is in "scientific notation", where `e` indicates an exponent of 10 and what follows `e` is the exponent. `e-15` indicates 10^{-15}, `e-01` indicates 10^{-1}, and `e+10` indicates 10^{15}. So `-1.337e-15` indicates -1.337×10^{-15}, a very small number, and `1.337e-01` indicates 1.337×10^{-01}, or about -0.11337.