Shiny Workflow

This document explains the workflow of the shiny app in this repository. The basic workflow is:

1. Take inputs about user's nonprofit
2. Let the user define the job market of their potential CEO by inputting comparison criteria for other nonprofits.
3. The report - take in the inputs from the first two steps and output a suggested pay as well as characteristics about the comparison set generated in step 2.

Step 0: global.R

The following needs to be in the global.R file.

```r
packs = c("shiny", "shinydashboard", "shinythemes", "shinyWidgets","shinyvalidate",
          "shinyhelper", "shinyglide", "plotly", "DT",
          "knitr", "stats", "knitr", "kableExtra",
          "dplyr", "readr", "tidyr","bslib", "datasets" )

#invisible(lapply(packs, library, character.only = TRUE))


### Run the following command to verify that the required packages are installed. If some package
# is missing, it will be installed automatically
package.check <- lapply(packs, FUN = function(x) {
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = TRUE)
  }
})
#load libraries
invisible(lapply(packs, library, character.only = TRUE))


### Load Data set
#dat <- read_csv("data-wrangle/data-by-sector.csv")
setwd("~/Dropbox/Olivia/Conflict/school/UI/Project6/compensation-appraisal")
dat <- readr::read_csv("data-rodeo/dat-shinyapp.csv")

### Load internal functions
source("funcs/applying-filters-func.R")
source("funcs/dat-filtering-hard.R")
source("funcs/distance-metric.R")
source("funcs/dollarize.R")
source("funcs/state-distance.R")
```

```r
### Needed in distance metric
region <- data.table::data.table(state = sort(c(state.abb, "PR", "DC")),
                                 region = c(9, 6, 7, 8, 9, 8, 1, 5, 1, 5,
                                            5, 9, 5, 8, 3, 3, 4, 6, 7, 1,
                                            1, 1, 6, 4, 4, 6, 8, 5, 4, 4,
                                            1, 1, 8, 8, 1, 3, 7, 9, 1, 5,
                                            1, 5, 4, 6, 7, 8, 5, 1, 8, 3,
                                            5, 8))
```

The remaining steps in the document are included in the ui.R and server.R.

Step 1: User Inputs

The user will input their organization's characteristics.

The possible organization characteristics are:

- State (AL, AK, . . . , WY)
- What type of location are you in? (Metropolitan, Rural)
- Broad Category (1, 2, . . . , 10). (https://nccs.urban.org/project/national-taxonomy-exempt-entities-ntee-codes)
- Major Group (A, . . . , Z) note major groups exist inside broad categories (https://nccs.urban.org/project/national-taxonomy-exempt-entities-ntee-codes)
- Does your organization fall into any of following NTEE-CC categories? (https://nccs.urban.org/publication/irs-activity-codes)
  - 01: Alliance/Advocacy Organizations
  - 02: Management and Technical Assistance
  - 03: Professional Societies/Associations
  - 05: Research Institutes and/or Public Policy Analysis
  - 11: Monetary Support - Single Organization
  - 12: Monetary Support - Multiple Organizations
  - 19: Non-monetary Support Not Elsewhere Classified (N.E.C.)
  - 00: I am a regular non-profit. I do not fit into any of these specialty organizations.
- Are you a university? (Yes or No)
- Are you a hospital? (Yes or No)
- Total Expenses (0 - Inf)
- Total Employees (0 - Inf)

The organization inputs are taken in from `intputs` in the shiny app. These `inputs` then need to be formatted in EXACTLY the following `list` format with EXACTLY these names. It is required for the inputs of the later functions. Here is a table of the list names and the available inputs for organization:

Input Name	**Potential Input Options**
State	State abbreviations as a character vector (including D.C. and Puerto Rico): "AL" , "AK", ... "WY"
Loc	Either "Rural" or "Metropolitan"
MajorGroup	One of the 10 Broad Category options: 1, 2, ..., 10
NTEE	One of the 26 Major Group options: "A", "B", ..., "Z"
NTEE.CC	One of the 7 common codes: "01", "02", "03", "05", "11", "12", "19", "00"
UNIV	Are you a university? TRUE or FALSE
HOSP	Are you a hospital? TRUE or FALSE
TotalExpense	Total expenses from last IRS filing year: any number greater than 0
TotalEmployee	Total employees from last IRS filing year: any whole number greater than 0

Note: There can only be one item under each name for the organizations list (this is not the case for the search list later). E.g. “State” can only have one state abbreviation in it, not 2. This functionality is build into the structure of the shiny app. (i.e. using pickerInput(multiple=FALSE)).

Here is a pretend organization:

```r
#org characteristics
org <- list(State = "DC",
            Loc = "Metropolitan",
            MajorGroup = 2,
            NTEE = "B",
            NTEE.CC = NA,
            UNIV = FALSE,
            HOSP = FALSE,
            TotalExpense = 3000000,
            TotalEmployee = 9
)
```

Step 2: Search Criteria

User inputs The user then inputs their search criteria. They specify which types of organizations they want to compare themselves to and if each criteria is a hard or soft match. (Hard match is we will only consider what they have provided in that criteria and then if there are multiple options listed we will prioritize based on exact matches, and a soft match is we will prioritize organizations that meet their search criteria and exact matches. This is explained more in the distance-metric vignette.). The search criteria is:

- List of States (AL, AK, . . . , WY)
- What type of locations do you want to consider? (Urban, Suburban, Rural)
- Major Group (1, 2, . . . , 10). (https://nccs.urban.org/project/national-taxonomy-exempt-entities-ntee-codes)
- NTEE Code (A, . . . , Z) (https://nccs.urban.org/project/national-taxonomy-exempt-entities-ntee-codes)

- Do you want to include any organizations that have the following Common Codes codes in your search? (https://nccs.urban.org/publication/irs-activity-codes)
  - 01 Alliance/Advocacy Organizations
  - 02 Management and Technical Assistance
  - 03 Professional Societies/Associations
  - 05 Research Institutes and/or Public Policy Analysis
  - 11 Monetary Support - Single Organization
  - 12 Monetary Support - Multiple Organizations
  - 19 Non-monetary Support Not Elsewhere Classified (N.E.C.)
  - 00 I am a regular nonprofit. I do not fit any of these specality categories.
- Do you want to include universities? (1- no, 2-yes, I want to include both orgs that are and are not universities, 3 - yes, I only want to consider universities)
- Do you want to include hospitals? (1- no, 2-yes, I want to include both orgs that are and are not hospitals, 3 - yes, I only want to consider hospitals)
- Range of Total Expenses c(min expenses, max expenses) (default is c(0,Inf))
- Range of Full Time equivalent Expenses c(min expenses, max expenses)(default is c(0,Inf))

If any search criteria is a NA value, there will be no hard matching on that criteria and every option for that criteria will be included in the soft matching process. (This will mostly come up with NTEE and NTEE-CC Codes but is also used as a fail-safe if a user doesn't include a search input).

Like the organization list, the search list also needs to be in a specific format. But for State, MajorGroup, NTEE, and NTEE.CC are vectors with (potentially) multiple options (i.e. using pickerInput(multiple=TRUE)). . Users can decide if they want to filter by total expenses and total employees if they want to. TotalEmployee and TotalExpense are a vector with minimum and maximum search criteria, e.g. c(min_number, max_number). The minimum number of employees that can be entered is 0 and the maximum number is 50,000. The minimum number of expenses that can be entered is 0 and the maximum number is 6 trillion. (See the helpfiles/SearchTotalEmployee and helpfiles/SearchTotalExpense for examples) Here is a table of the list names and the available inputs for the search criteria:

Input Name	**Potential Input Options**
State	A vector containing any state abbreviations that the user wants to include as a character vector (including D.C. and Puerto Rico): "AL" , "AK", ... "WY"
Loc	Either "Rural" or "Metropolitan" or c("Rural", "Metropolitan")
MajorGroup	A vector containing any of the 10 Broad Category options: 1, 2, ... 10
NTEE	A vector containing any of the 26 Major Group options: "A", "B", ... , "Z"
NTEE.CC	A vector containing any of the 7 available common codes: "01", "02", "03", "05", "11", "12", "19", "00"
UNIV	Do you want to include universities in your search? 1 = No, I do not want to include universities, 2 = Yes, I want to include both universities and non-universities, 3 = Yes, I exclusively want to include universities.
HOSP	Are you a hospital? Do you want to include hospitals in your search? 1 = No, I do not want to include hospitals, 2 = Yes, I want to include both hospitals and non-hospitals, 3 = Yes, I exclusively want to include hospitals.
TotalExpense	Range of total expenses the user wants to include as format c(min_number, max_number). If not filtering on this criteria then default to c(0, Inf).
TotalEmployee	Range of total employee the user wants to include as format c(min_number, max_number). If not filtering on this criteria then default to c(0, Inf).

Currently users can only select to filter by state or location (not both), and they can choose to filter by braod categroy, major group, or common code (not an combo of the 2).

For each criteria, the user then specifies if they want a hard or soft match. If the user chooses hard match, we will filter out orgs that do not fit the search criteria they provide. There is no hard or soft options for UNIV and HOSP as these are true and false options. If the user wants to hard match, set the option equal to TRUE, if the user wants to soft match set the option equal to FALSE.

A pretend search criteria and hard/soft filters is below:

```r
#search criteria
search <- list(State = c("VA", "DC", "MD"),
               MajorGroup = 2,
               Loc = c("Metropolitan"),
               NTEE = c("B"),
               NTEE.CC = NA,
               UNIV = 1,
               HOSP = 1,
               TotalExpense = c(1000000, 6000000),
               TotalEmployee = c(0, 100)
)
```

```
#TRUE is hard match, FALSE is soft match
#TotalExpense and TotalEmployee do not have hard/soft options
hard <- list(State = FALSE,
             Loc = TRUE,
             MajorGroup = TRUE,
             NTEE = TRUE,
             NTEE.CC = FALSE,
             TotalExpense = TRUE,
             TotalEmployee = TRUE
)
```

Creating the Comparision set We first filter the data to match their hard search criteria using the `dat_filtering_hard` function in the func/dat-filtering-hard.R directory. This function is why we need the organization, search, and hard lists to be in a specific format.

```
dat.filtered <- dat_filtering_hard(search, hard)
```

Row Selection This is not implemented into the shiny app yet be we would like implement it in the final product.

We next allow the user to deselect organizations they do not want to include in their search. In the shiny app we will use the DT::dataTable options.

```
#say for whatever reason the user does not want to include these orgs
# rows.select <- c(1,2,3,4,5)
# dat.filtered <- dat.filtered[ -rows.select, ]
```

Just a note for when/if this gets implemented, when you allow users to see the data set for selecting/deselecting ogs to include in the comparison set, suppress the CEOCompensation and Gender Columns. This way the user can not cherry pick orgs with higher CEO pay or only male lead orgs. I highly doubt many people would do something nafarious like this, but we want to make sure we make it impossible for them to do it.

We intend this function for users to be able to deselect orgs that might fit their filtering criteria but are still "different" from them. e.g. If the user's org is a small animal shelter, but the ASPCA makes it into their search criteria, they can take the ASPCA out of their search because while the ASPCA has local branches, they are supported by a larger central organization.

Get the distances We then take these selected options and find the distance of each org in the search criteria to the user's org. We use the `HEOM_with_weights` function in the funcs/distance-metric.R file.

See the HEOM vignette for details about how we are measuring distance.

There are 9 matching criteria: $\log_{10}$(Total Expenses +1), $\log_{10}$(Total Employee +1), State, Major Group, NTEE, NTEE.CC, UNIV, HOSP, and Location type.

```
results <- HEOM_with_weights(org, search, dat.filtered)
```

Get the weighted average using the distance metric Let d_i be the distance metric for each organization, i, in the comparison set. Let $w_i = \frac{1-d_i}{\sum_i (1-d_i)}$ be the weight for each organization. Let C_i be the CEO compensation (in 2022 dollars) of organization i.

We then use the weighted average as th suggested pay as $\sum_i w_i C_i$.

```
d_i <- results$dist
sum_w <- sum(1 - d_i)
w_i <- (1-d_i) / sum_w
#sum(w_i) = 1

weighted.average <- sum(w_i * results$CEOCompensation)
```

Range of Pay We next get the range of pay by

1. Making a regression model using the same characteristics used for the data imputation in the 990EZ total employees plus gender and location.
2. Getting the residuals of this model as percentages of their original statistics.
3. Find the 10% and 90% quantiles of these percentage residuals. (Sometimes CEO get bonus' or maybe only get paid for part of the year so this gets ride of any of those edge cases)
4. Report final suggested pay range as weighted.average + weighted.average * quants.

```
dat.model <- results %>%
  #only select the variables we need for regression
  select(FormYr, TotalAssests, GrossReceipts, TotalExpense, TotalEmployee, MajorGroup, State, UNIV, HOS
  #make numierics log
  dplyr::mutate(log.assests = log(TotalAssests + 1, 10)) %>%
  dplyr::mutate(log.gross = log(GrossReceipts + 1, 10)) %>%
  dplyr::mutate(log.expense = log(TotalExpense + 1,  10)) %>%
  dplyr::mutate(log.employee= log(TotalEmployee + 1, 10)) %>%
  #make categoricals factors
  dplyr::mutate(FormYr = as.factor(FormYr )) %>%
  dplyr::mutate(MajorGroup = as.factor(MajorGroup )) %>%
  dplyr::mutate(UNIV = as.factor(UNIV )) %>%
  dplyr::mutate(HOSP = as.factor(HOSP )) %>%
  dplyr::mutate(State = as.factor(State )) %>%
  dplyr::mutate(LocationType = as.factor(LocationType )) %>%
  dplyr::mutate(Gender = as.factor(Gender )) %>%

  #get ride of things not needed in the model
  select(-c(TotalAssests, GrossReceipts, TotalExpense, TotalEmployee ))

#only use complete data
dat.model <- na.omit(dat.model)

#only use factors with more than one category (this is a `lm` issue not a theory of regression issue)
values_count <- sapply(lapply(dat.model, unique), length)

#get the model
mod <- lm(CEOCompensation ~ ., dat.model[ , values_count > 1])

#get residuals as a percentage of their original value
resids <- mod$residuals
per <- resids / dat.model$CEOCompensation / 100
```

```r
quants <- quantile(per, c(0.10, 0.90)) #get rid of the bonus years

#get suggested pay range
suggested.range<- weighted.average + weighted.average * quants

#the range of values is
suggested.range
```

```
##      10%      90%
## 178257.6 180743.0
```

```r
weighted.average
```

```
## [1] 180059.2
```

In the report we need to include basic summary information about the regression model used.

Gender adjusted This is not implemented yet, but could be later. We adjust for gender by adding the coefficient of the "Gender" variable to only the women CEO's.

```r
#get the beta-hat for being a male
#side note the p-value changes greatly with the inputs but this makes sense

diff <- unname(mod$coefficients["GenderM"])

### Exactly the same as doing this
#
# #make all the women coded as men
# dat.women <- dat.model %>%
#   filter(Gender == "F")
#
# dat.women.as.men <- dat.women%>%
#   mutate(Gender = "M")
#
# #get the predicted values with women coded as men
# y.hat.women <- predict(mod, dat.women.as.men)
#
# #get the average difference between the gender adjusted predicted value and the original model predic
# y.hat.orig <- predict(mod, dat.women)
# diff <- mean(abs(y.hat.women - y.hat.orig))


#get the percentage residuals of the women
y.hat.orig <- mod$fitted.values
is.female <- dat.model$Gender == "F"

#only add diff if the CEO is a women
y.hat.adjusted <- y.hat.orig + diff*is.female

#get new residual
resid.adjusted <- dat.model$CEOCompensation - y.hat.adjusted
```

```r
per.adjusted <- resid.adjusted /( dat.model$CEOCompensation + diff*is.female) / 100
quants.adjusted <- quantile(per.adjusted, c(0.10, 0.90))

#output gender adjusted suggested range
suggested.range.adjusted <- weighted.average + weighted.average * quants.adjusted

#the range of values is
suggested.range.adjusted
```

```
##      10%      90%
## 178257.8 180592.7
```

```r
weighted.average
```

```
## [1] 180059.2
```

```r
###### exploring the differences

# plot(y.hat.orig, y.hat.adjusted)
# abline(0, 1)
#
# #lower quantiles comes up
# quantile(y.hat.orig, c(0, 10, 25, 50, 75, 90, 100)/100)
# quantile(y.hat.adjusted, c(0, 10, 25, 50, 75, 90, 100)/100)
#
# quantile(per, c(0, 10, 25, 50, 75, 90, 100)/100)
# quantile(per.adjusted, c(0, 10, 25, 50, 75, 90, 100)/100)
```

Side note about p-values: sometimes the p-value for the indicator for Male/Female can get up to something like .2. Even though that is conventionally high, I don't think the traditional p-value cut offs are appropriate in this scenario. In my opinion, this is enough for us to consider worth looking at. Just because the gender pay gap is happening at a rate of above p-value = 0.05 doesn't mean its not happening or not having an affect on people's lives. I am also very opinionated about p-values so many may not agree with this stance.

Exploring the data From here you can easily create any types of plots, tables, or other visualizations that you want to.

```r
#
# plot(results$TotalExpense, results$CEOCompensation)
# plot(results$TotalEmployee, results$CEOCompensation)
#
#
# hist(results$CEOCompensation, breaks = 20)
# boxplot(results$CEOCompensation)
# quantile(results$CEOCompensation, c(0, 10, 25, 50, 75, 90, 100)/ 100)
# mean(results$CEOCompensation)
#
#
# ggplot(results) +
#   geom_point(aes(x = TotalExpense, y = CEOCompensation, color = dist))
#
```

```
# results1 <- results[results$TotalEmployee > 0, ]
# hist(results1$CEOCompensation, breaks = 50)
#
# mean(results$CEOCompensation[results1$dist <= 0.1])
#
# plot(results1$TotalExpense, results1$CEOCompensation)
# plot(results1$TotalEmployee, results1$CEOCompensation)
#
#
# boxplot(results1$CEOCompensation)
# quantile(results1$CEOCompensation, c(0, 10, 25, 50, 75, 90, 100)/ 100)
# mean(results1$CEOCompensation)
#
#
# ggplot(results1) +
#   geom_point(aes(x = TotalExpense, y = CEOCompensation, color = dist))
#
# #take the average of CEO pay whose distance is <0.1
# mean(results1$CEOCompensation[results1$dist <= 0.1])
```

Removing the orgs with 0 employees Exploring the distributions of distance metrics

```
# hist(results$dist, breaks = 20)
# boxplot(results$dist)
# #quantile(results$dist, c(0, 10, 25, 50, 75, 90, 100)/ 100)
```

Running the Shiny App.

The shiny app can be run from any machine without downloading the repo by running the following code:

```
# install.packages("shiny")
# shiny::runGitHub("compensation-appraisal",  "Nonprofit-Open-Data-Collective")
```

Future Plans

To whoever is picking up this project, apologies for the confusion about the differences in between broad category, major group, and common codes names, and the MajorGroup, NTEE, and NTEE.CC variables names. In colloquial conversations, we used MajorGroup to refer to the 10 groups labeled I, II, . . . , X. We used NTEE to refer to the 26 letter groups. We used common code to refer to the (Letter)(2 digit number) e.g. A04. So we wrote up the code using those names, but once we went back and read the post about NTEE codes on Urban's website, we realized we got the names wrong, and the clearest way to display the names for the user was to use Broad Category as the 10 groups labeled I, II, . . . , X. Major group as the 26 letters, and common codes as the 2 digit numbers. Here is table of variable names, what the user sees, and the group label names.

Urban's NTEE article can be found here: https://nccs.urban.org/project/national-taxonomy-exempt-entities-ntee-codes

The entire code book is found here: https://nccs.urban.org/publication/irs-activity-codes

What Urban NTEE Page says	Variable Name	Group Labels
Broad Category	MajorGroup	I, II, ..., X
Major Group	NTEE	A, B, ..., Z
Common Code	NTEE.CC	A01, A02, ..., Z99

These variable names can be changed to better reflect what they actually represent, but admittidly, it will take quite a bit of work.

Additional Functionality Here is a list of things we want to be able to do in the future. f

- Want to expand this to more than just CEO's. Want to do this exact method with CFO's, COO's or other "high-ranking" positions. Need to wait for Jesse and his other collaborators for the job title identification data set from the IRS data.
- We would like to look at transitional pay with the difference-in-difference model like the worksheet that Jesse has (on the Trello resources column).

Advanced Features Here is a list of features we want to implement in the shiny app that are not there currently.

- We want to expand to more common code options. We wanted to make sure these specialty organizations could also use the tool, hence the current common code options only reflects these specialty orgs. We want eventually to allow users to input any of the common codes.
- Using shiny validate on all user inputs, some input validations are implemented at the beginning of the server.R to get an idea of what this should look like, but most of the inputs do not have this implemented yet.
- Inside shinyglide::screen(), using the next_condition option to make the next condition reactive to the current screen inputs. An example of this is commented on the organization size questions. The main issue I had implementing this was I do not know the equivalent of is.na() is java. If a pickerInput is blank then the variable is assigned value NA. next_condition uses java script as the verifier See the help file for ?shinyglide::screen() more details on the java script part.
- We would eventually like to report a "gender-neutral" CEO pay suggestion. I talked briefly about what this might look like in the Report section, but how to do this is still in the conversational phase.
- We would like to have at the bottom of every screen saying something like "There are X orgs left in your comarison set". Code wise, this is very easy to implement, but for reasons described in the "Known Issues" section, it is not currently implemented.
  - A super advanced version of this would be to have every option list how many orgs are in the comparison set and in that category. For example if you say I want to filter by location
- Not generating the report if there aren't enough orgs in the comaprison set

Known Issues

- Shiny glide pages

There is a error in the shiny glide screens where if a DT object is not loaded (or any calculations that depend on dat.filtered.pre are loaded on screens before “the report” screen) where the screens are not the width of the entire screen and if you scroll through the screens they get more and more off. If you run the shiny app exactly as is except only comment out the very last screen with the DTOutput then you will see what I am talking about. My best guess as to why this is happening is with the conditionalPanels. Something about how a conditionalPanel gets rendered in the css document conflicts with how the shinyglide screen gets rendered in the css. However, if you output a DTOutput, it overrides this error in the hierarchy of the css elements. This is obviously not the ideal solution, but I do not understand enough about css to know how to resolve this issue. I’ve narrowed down the issue to the conditionalPanels and shiny glide conflict by commenting out everything and adding back in one element at a time.