Pensions ESG Pilot

OVD

2023-03-16

options(scipen=999)  
  
options(digits = 3)

library(readr)  
  
dfcj <- read\_csv("rds\_prod.experiment.420656.stacked(9).csv")

## Rows: 45528 Columns: 87  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (28): EXPECTED\_PENSION, INVESTS\_IN\_FIREARMS, INVESTS\_IN\_FOSSIL\_FUELS, I...  
## dbl (57): RESPONDENT\_ID, SURVEY\_ID, CHOICE\_SET, LABEL, CHOICE\_INDICATOR, RE...  
## dttm (2): RESPONDENT\_TIME\_OF\_OPENING\_SURVEY, RESPONDENT\_TIME\_OF\_COMPLETING\_...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

library(cregg)  
library(janitor)  
library(tidyverse)

Import data and filter responses 0.5 or 1.5 \* median completion time

dfcj = dfcj %>%   
 clean\_names()

#Transform variables to factors  
dfcj = dfcj %>%   
 mutate(expected\_pension\_num = factor(expected\_pension),  
 firearms = factor(invests\_in\_firearms,  
 levels = c("Invests in firearms", "Does not invest in firearms")),   
 fossil\_fuels = factor(invests\_in\_fossil\_fuels,  
 levels = c("Invests in fossil fuels", "Does not invest in fossil fuels")),  
 may\_employ\_children = factor(invests\_in\_firms\_that\_may\_employ\_children,  
 levels = c("Invests in firms that may employ children", "Invests in firms that ensure no children are employed")),  
 racial\_diversity = factor(advocates\_for\_racial\_diversity\_in\_management,  
 levels = c("Does not advocate for racial diversity in management",  
 "Advocates for racial diversity in management")),  
 gender\_equal\_pay = factor(advocates\_for\_equal\_pay\_for\_men\_and\_women,  
 levels = c("Does not advocate for equal pay for men and women",  
 "Advocates for equal pay for men and women")),  
 choice\_indicator = as.numeric(choice\_indicator),  
 choice = as.factor(choice\_indicator),  
 prior = factor(ifelse(q9\_taking\_into\_account\_esg\_factors\_o1\_yes ==1, "anti-esg", "pro=esg")),  
 republican = factor(ifelse(q11\_party\_id\_o1\_republican==1 |  
 q12\_party\_leanings\_o1\_republican ==1, 1,0)),  
 florida = factor(ifelse(q15\_state\_us\_o1\_florida==1, 1,0)),  
 california = factor(ifelse(q15\_state\_us\_o2\_california==1,1,0)))

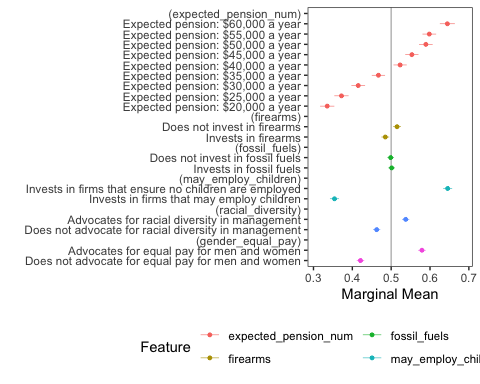
dfcjshort = dfcj %>%   
 select(survey\_id,   
 choice\_set,   
 expected\_pension,  
 expected\_pension\_num,  
 choice\_indicator,   
 choice,  
 firearms,  
 fossil\_fuels,  
 may\_employ\_children,  
 racial\_diversity,  
 gender\_equal\_pay,  
 republican,  
 prior,  
 q22\_openended\_feedback)

dfus = dfcj %>%   
 filter(florida==0 & california==0)

mmus <- cj(dfus, choice\_indicator ~ expected\_pension\_num + firearms + fossil\_fuels + may\_employ\_children + racial\_diversity + gender\_equal\_pay, id = ~ survey\_id,  
 estimate = "mm")

## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.

plot(mmus, vline = 0.5) #'\*# See what is going on with fossil fuels\*



Let’s try with weights

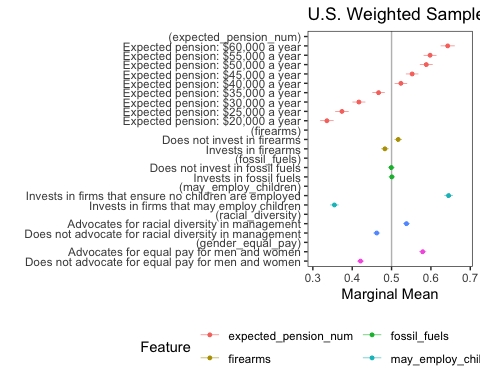
dfcjw = dfcj %>%   
 mutate(weight = case\_when(florida == 1 ~ 0.066,  
 california == 1 ~ 0.12,  
 florida == 0 & california == 0 ~ 0.814))  
  
table(dfcjw$weight) # Looks good

##   
## 0.066 0.12 0.814   
## 8784 9408 27336

# weight = dfcjw$weights  
  
mmusw <- cj(dfcjw, choice\_indicator ~ expected\_pension\_num + firearms + fossil\_fuels + may\_employ\_children + racial\_diversity + gender\_equal\_pay, id = ~ survey\_id,  
 estimate = "mm", weights = ~ weight)

## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.

plot(mmusw, vline = 0.5) +   
 ggtitle("U.S. Weighted Sample")

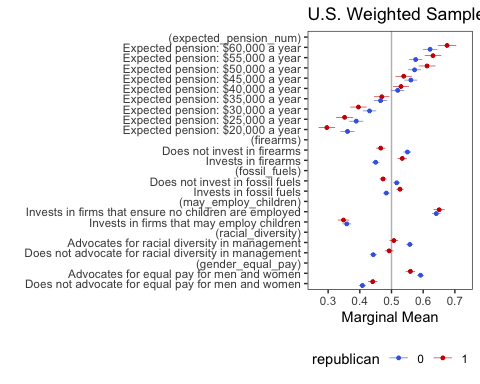


mmusw\_by <- cj(dfcjw, choice\_indicator ~ expected\_pension\_num + firearms + fossil\_fuels + may\_employ\_children + racial\_diversity + gender\_equal\_pay, id = ~ survey\_id,  
 estimate = "mm", weights = ~ weight,  
 by = ~republican)

## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.

plot(mmusw\_by, group = "republican", vline = 0.5) +  
 scale\_colour\_manual(na.translate = F, values = c("royalblue", "red3")) +  
 ggtitle("U.S. Weighted Sample")

## Scale for colour is already present.  
## Adding another scale for colour, which will replace the existing scale.



Vector for AMCE model and plot:

Now trying with larger df

mm\_by <- cj(dfcj, choice\_indicator ~ expected\_pension\_num + firearms + fossil\_fuels + may\_employ\_children + racial\_diversity + gender\_equal\_pay, id = ~ survey\_id,  
 estimate = "mm",   
 by = ~republican)

## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.

amce\_by <- cj(dfcj, choice\_indicator ~ expected\_pension\_num + firearms + fossil\_fuels + may\_employ\_children + racial\_diversity + gender\_equal\_pay, id = ~ survey\_id,  
 estimate = "amce",   
 by = ~republican)

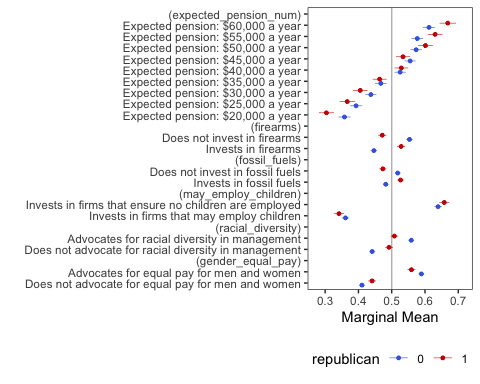
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.

amce <- cj(dfcj, choice\_indicator ~ expected\_pension\_num + firearms + fossil\_fuels + may\_employ\_children + racial\_diversity + gender\_equal\_pay, id = ~ survey\_id,  
 estimate = "amce")

## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.

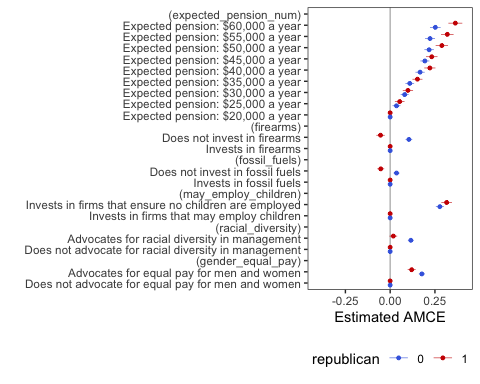
plot(mm\_by, group = "republican", vline = 0.5) +  
 scale\_colour\_manual(na.translate = F, values = c("royalblue", "red3"))

## Scale for colour is already present.  
## Adding another scale for colour, which will replace the existing scale.



plot(amce\_by, group = "republican") +  
 scale\_colour\_manual(na.translate = F, values = c("royalblue", "red3"))

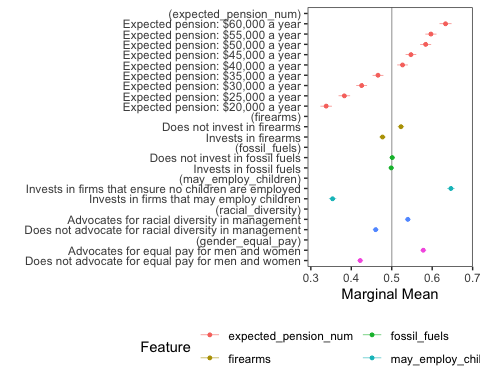
## Scale for colour is already present.  
## Adding another scale for colour, which will replace the existing scale.



mm <- cj(dfcj, choice\_indicator ~ expected\_pension\_num + firearms + fossil\_fuels + may\_employ\_children + racial\_diversity + gender\_equal\_pay, id = ~ survey\_id,  
 estimate = "mm")

## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.

plot(mm, vline = 0.5) #'\*# See what is going on with fossil fuels\*

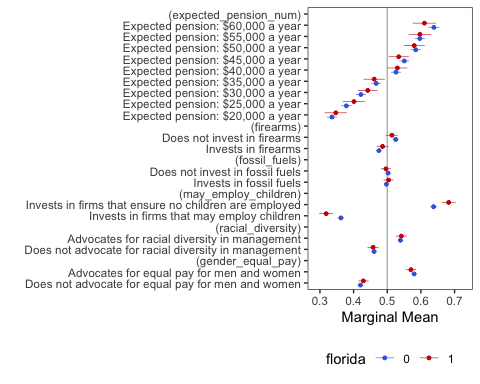


mm\_byfl <- cj(dfcj, choice\_indicator ~ expected\_pension\_num + firearms + fossil\_fuels + may\_employ\_children + racial\_diversity + gender\_equal\_pay, id = ~ survey\_id,  
 estimate = "mm",   
 by = ~florida)

## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.  
  
## Warning in logLik.svyglm(x): svyglm not fitted by maximum likelihood.

plot(mm\_byfl, group = "florida", vline = 0.5) +  
 scale\_colour\_manual(na.translate = F, values = c("royalblue", "red3"))

## Scale for colour is already present.  
## Adding another scale for colour, which will replace the existing scale.



WTP analysis. The logitr package uses a Hierarchical Bayesian model to estimate WTP and utilities from each attribute.

library(logitr)

## Version: 1.0.1  
## Author: John Paul Helveston (George Washington University)  
##   
## Consider submitting praise at  
## https://github.com/jhelvy/logitr/issues/8.  
##   
## Please cite the JSS article in your publications, see:  
## citation("logitr")

dfcjtest = dfcj %>%   
 mutate(pension = expected\_pension)  
  
  
dfcjtest$pension <- recode(dfcjtest$pension,  
 "Expected pension: $20,000 a year" = "20,000",   
 "Expected pension: $25,000 a year" = "25,000",   
 "Expected pension: $30,000 a year" = "30,000",   
 "Expected pension: $35,000 a year" = "35,000",   
 "Expected pension: $40,000 a year" = "40,000",   
 "Expected pension: $45,000 a year" = "45,000",   
 "Expected pension: $50,000 a year" = "50,000",  
 "Expected pension: $55,000 a year" = "55,000",  
 "Expected pension: $60,000 a year" = "60,000"  
 )

#transform to numeric  
  
dfcjtest = dfcjtest %>%  
 mutate(pension\_num = readr::parse\_number(pension))

Not creating var to test MLogit

Careful with grouping!!! Didn’t ungroup before

#This code below worked!  
  
dfcjtest2 = dfcjtest %>%  
 group\_by(survey\_id, choice\_set) %>%  
 mutate(obs\_id = cur\_group\_id()) %>%   
 ungroup() # I had forgotten to do this!

mean(dfcjtest2$pension\_num)

## [1] 39998

dflogitr = dfcjtest2 %>%   
 select(survey\_id,   
 obs\_id,   
 choice\_set,   
 pension\_num,  
 choice\_indicator,   
 choice,  
 firearms,  
 fossil\_fuels,  
 may\_employ\_children,  
 racial\_diversity,  
 gender\_equal\_pay,  
 republican)

dflogitr = dflogitr %>%   
 mutate(price = -1 \* pension\_num,  
 firearms.num = ifelse(firearms == "Invests in firearms", 1, 0),  
 fossil\_fuels.num = ifelse(fossil\_fuels == "Invests in fossil fuels", 1, 0),  
 may\_employ\_children.num = ifelse(may\_employ\_children == "Invests in firms that may employ children", 1, 0),  
 racial\_diversity.num = ifelse(racial\_diversity == "Does not advocate for racial diversity in management", 1, 0),  
 gender\_equal\_pay.num = ifelse(gender\_equal\_pay == "Does not advocate for equal pay for men and women", 1, 0))  
  
dflogitr = dflogitr %>%   
 rename(no\_racial\_diversity.num = racial\_diversity.num,  
 no\_gender\_equal\_pay.num = gender\_equal\_pay.num)

#Word cloud with reliable data  
  
#Create a vector containing only the text  
txt = as.character(dfcj$q22\_openended\_feedback)  
  
  
## Loading packages for word clouds  
library(wordcloud)

## Loading required package: RColorBrewer

library(RColorBrewer)  
library(wordcloud2)  
library(tm)

## Loading required package: NLP

##   
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':  
##   
## annotate

# Create a corpus   
docs <- Corpus(VectorSource(txt))

# clean text data  
docs <- docs %>%  
 tm\_map(removeNumbers) %>%  
 tm\_map(removePunctuation) %>%  
 tm\_map(stripWhitespace)

## Warning in tm\_map.SimpleCorpus(., removeNumbers): transformation drops documents

## Warning in tm\_map.SimpleCorpus(., removePunctuation): transformation drops  
## documents

## Warning in tm\_map.SimpleCorpus(., stripWhitespace): transformation drops  
## documents

docs <- tm\_map(docs, content\_transformer(tolower))

## Warning in tm\_map.SimpleCorpus(docs, content\_transformer(tolower)):  
## transformation drops documents

docs <- tm\_map(docs, removeWords, stopwords("english"))

## Warning in tm\_map.SimpleCorpus(docs, removeWords, stopwords("english")):  
## transformation drops documents

#Document term matrix  
  
dtm <- TermDocumentMatrix(docs)   
matrix <- as.matrix(dtm)   
words <- sort(rowSums(matrix),decreasing=TRUE)   
df <- data.frame(word = names(words),freq=words)

# Create word cloud  
set.seed(1234) # for reproducibility   
  
wordcloud(words = df$word, freq = df$freq, min.freq = 1,   
 max.words=10, random.order=FALSE, rot.per=0.35,   
 colors=brewer.pal(8, "Dark2"))

