

# Pensions ESG Pilot

OVD

2023-03-16

## Contents

```
options(scipen=999)
```

```
options(digits = 3)
```

```
library(readr)
```

```
dfcj <- read_csv("rds_prod.experiment.420656.stacked(7).csv")
```

```
## Rows: 32160 Columns: 83
```

```
## -- Column specification -----
```

```
## Delimiter: ","
```

```
## chr  (24): EXPECTED_PENSION, INVESTS_IN_FIREARMS, INVESTS_IN_FOSSIL_FUELS, I...
```

```
## dbl  (57): RESPONDENT_ID, SURVEY_ID, CHOICE_SET, LABEL, CHOICE_INDICATOR, RE...
```

```
## dtm   (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 may not employ children")),
```

```
         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)

```

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,
            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.

```

```
amce_by <- cj(dfkj, choice_indicator ~ expected_pension_num + firearms + fossil_fuels + may_employ_children,
  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.
```

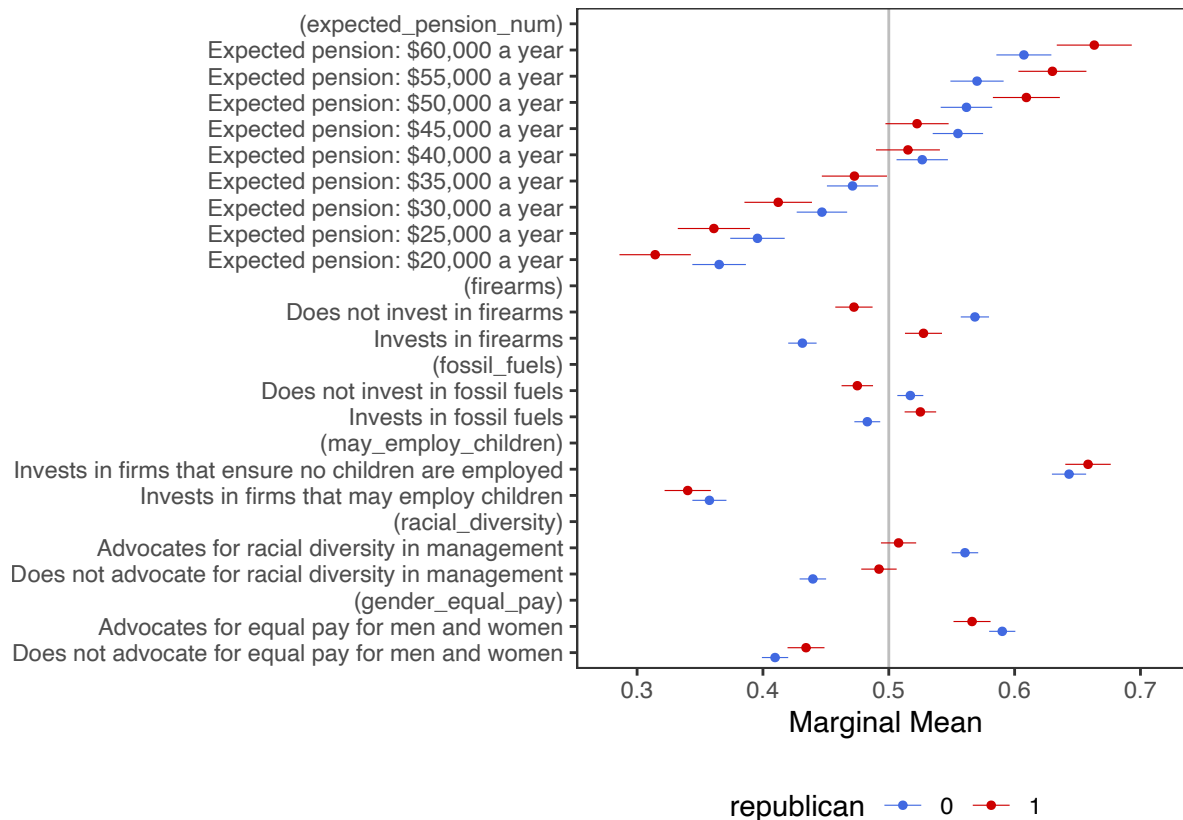
```
amce <- cj(dfkj, choice_indicator ~ expected_pension_num + firearms + fossil_fuels + may_employ_children,
  estimate = "amce")
```

```
## 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.
```

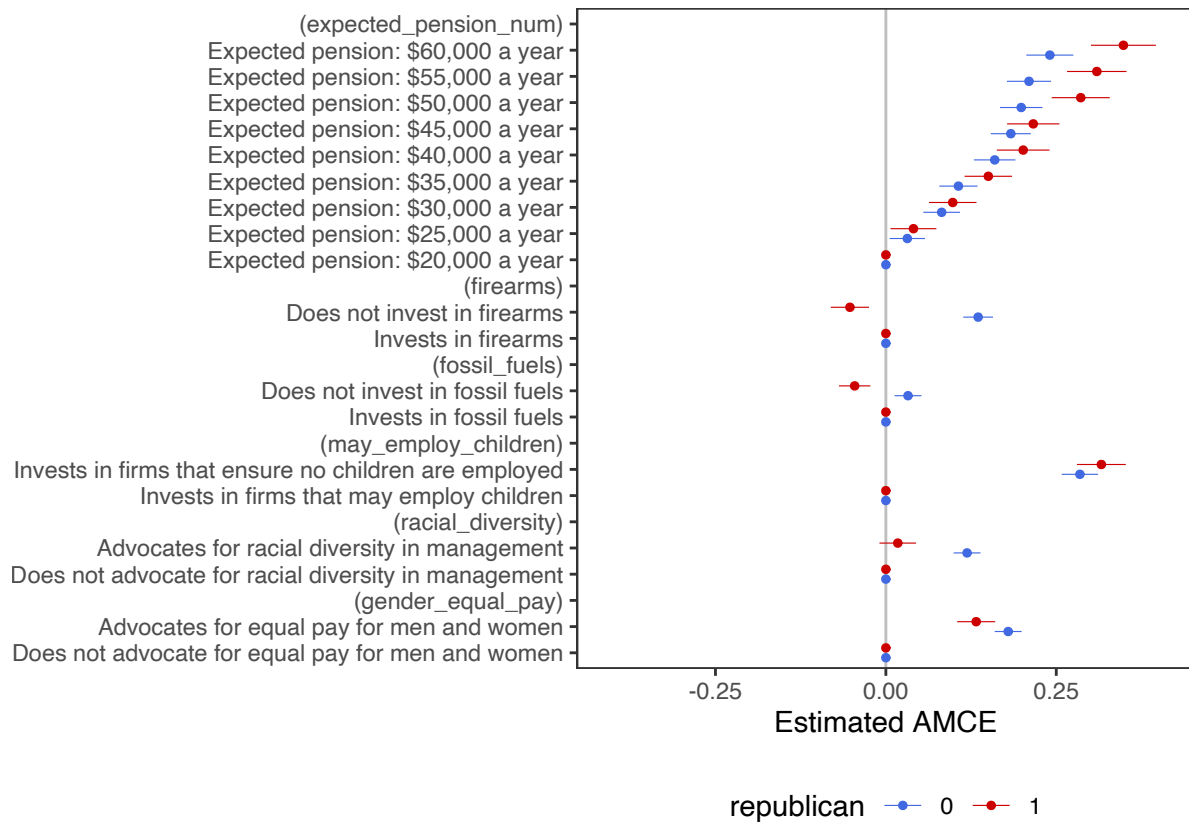
```
## Adding another scale for colour, which will replace the existing scale.
```



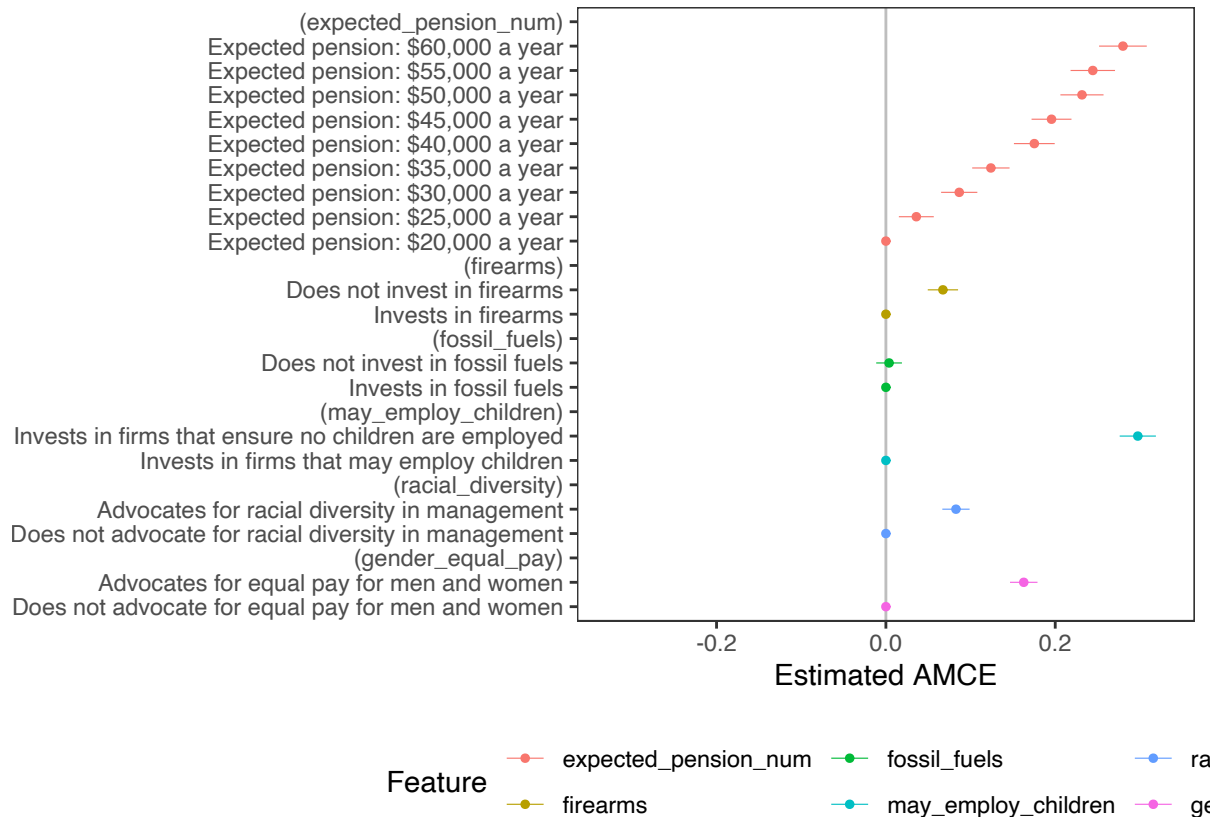
```
plot(amce_by, group = "republican") +
  scale_colour_manual(na.translate = F, values = c("royalblue", "red3"))
```

```
## Scale for colour is already present.
```

```
## Adding another scale for colour, which will replace the existing scale.
```



```
plot(amce)
```



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] 40001
```

```
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)
```

I will filter the data in republican/democrats to calculate the differences in WTP per group

```
# Same for merged df
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(df$dfcjq22_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"))
```

