

PC_Pilot_1

2024-11-18

Cleaning

```
library(readr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v purrr      1.0.2
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(mlogit)
```

```
## Loading required package: dfidx
##
## Attaching package: 'dfidx'
##
## The following object is masked from 'package:stats':
##
##   filter
```

```
data <- read_csv("PC_Pilot-English_Prolific_December 12, 2024_17.20.csv")
```

```
## Rows: 403 Columns: 81
## -- Column specification -----
## Delimiter: ","
## chr (81): StartDate, EndDate, Status, IPAddress, Progress, Duration (in seco...
##
## 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.
```

```
View(data)
```

```
data_header_rm <- data[-c(1,2),] #remove qualtrics duplicate headers
data_header_rm[data_header_rm == ""] <- NA #empty string = NA
```

```

View(data_header_rm)

# Prolific, pass consent, and prescreen
data_passed <- data_header_rm %>%
  filter(
    Finished == 1,
    consent == 1,
    str_length(Prolific_ID) == 24, # Prolific IDs
  )

#rows with fraud
data_fraud <- data_passed %>%
  filter(
    Q_BallotBoxStuffing == "True",
    Q_RecaptchaScore <= 0.5,
    Q_RelevantIDDuplicate == "True",
    Q_RelevantIDDuplicateScore >= 75,
    Q_RelevantIDFraudScore >= 30
  )

#remove unnecessary columns
data_passed1 <- data_passed %>%
  filter(!PROLIFIC_PID %in% data_fraud$PROLIFIC_PID) %>% #remove fraud
  select(-c(StartDate,
            EndDate,
            Status,
            IPAddress,
            Progress,
            Finished,
            RecordedDate,
            ResponseId,
            RecipientLastName,
            RecipientFirstName,
            RecipientEmail,
            ExternalReference,
            LocationLatitude,
            LocationLongitude,
            DistributionChannel,
            UserLanguage,
            'Duration (in seconds)',
            Q_BallotBoxStuffing,
            Q_RecaptchaScore,
            Q_RelevantIDDuplicate,
            Q_RelevantIDDuplicateScore,
            Q_RelevantIDFraudScore,
            PROLIFIC_PID,
            consent,
            not_pc_label,
            pc_label
          )) %>%
  mutate(ppt_ID = row_number())

```

```

# Separate demographics
data_demo <- data_passed1 %>%
  select(ppt_ID, Prolific_ID,
         label,
         starts_with("Demo_"),
         comment
        )

write.csv(data_demo, "demo_12102024.csv", row.names = FALSE)

# Main data
data_main <- data_passed1 %>%
  select(-c(Prolific_ID, # de-identified
            starts_with("Demo_"),
            comment)) %>%
  mutate(across(where(is.character), # Apply to character columns only
                ~ ifelse(!is.na(suppressWarnings(as.numeric(.))),
                          as.numeric(.),
                          .))) %>%

  mutate(
    experience_1 = as.numeric(experience_1),
    experience_2 = as.numeric(experience_2),
    experience_3 = as.numeric(experience_3),
    SliderOverlapValue = as.numeric(SliderOverlapValue)
  )

# Attention check
data_main <- data_main[data_main$check == 2, ]

```

Recode Data

```

# In primary and early_language, change 4=English and other to 3 = English and other.
data_main <- data_main %>%
  mutate(
    primary = if_else(primary == 4, 3, primary),
    early_language = if_else(early_language == 4, 3, early_language)
  )

# For all stigma_, 5 = the most stigmatized, 1 = the least stigmatized
## Recode stigma_resp_2, stigma_dist_2, stigma_dist_3
data_main_recode <- data_main %>%
  mutate(stigma_resp_2_mut = 6 - stigma_resp_2,
         stigma_dist_2_mut = 6 - stigma_dist_2,
         stigma_dist_3_mut = 6 - stigma_dist_3) %>%
  select(-c(stigma_resp_2, stigma_dist_2, stigma_dist_3))

# For all stertyp_, 5 = the most stereotyped (negatively)
## Reverse all values in stertyp_ columns
data_main_recode <- data_main_recode %>%
  mutate(across(
    starts_with("stertyp_"), # Select columns starting with "stertyp_"
    ~ 6 - ., # Reverse the values (assuming range is 1 to 5)
  ))

```

```

    .names = "{.col}_rev" # Add "_rev" to the new column names
  )) %>%
  select(-starts_with("stertyp_")|ends_with("_rev"))

data_main_noFR <- data_main_recode %>%
  select(-c(general_1, general_2, impression_1, impression_2, impression_3_4_TEXT, like_fr, offens_fr))
write.csv(data_main_noFR, "data_main_noFR.csv", row.names = FALSE)

```

Analysis - Simple

```

# Overlap compare
data_main_recode %>%
  group_by(label) %>%
  summarise(overlap_avg = mean(SliderOverlapValue))

```

```

## # A tibble: 2 x 2
##   label                                overlap_avg
##   <chr>                                <dbl>
## 1 people who experience housing insecurity      NA
## 2 the homeless                                NA

```

Stigma: PC vs. PIC

```

## average stigma
data_main_recode %>%
  mutate(stigma_avg_ppt = rowSums(select(., starts_with('stigma_')))/6) %>%
  group_by(label) %>%
  summarise(stigma_avg_label = mean(stigma_avg_ppt))

```

```

## # A tibble: 2 x 2
##   label                                stigma_avg_label
##   <chr>                                <dbl>
## 1 people who experience housing insecurity      2.07
## 2 the homeless                                2.28

```

```

## stigma_resp
data_main_recode %>%
  mutate(stigma_resp_ppt = rowSums(select(., starts_with('stigma_resp')))/2) %>%
  group_by(label) %>%
  summarise(stigma_resp_label = mean(stigma_resp_ppt))

```

```

## # A tibble: 2 x 2
##   label                                stigma_resp_label
##   <chr>                                <dbl>
## 1 people who experience housing insecurity      2.22
## 2 the homeless                                2.16

```

```
## stigma_dist
data_main_recode %>%
  mutate(stigma_dist_ppt = rowSums(select(., starts_with('stigma_dist')))/3) %>%
  group_by(label) %>%
  summarise(stigma_dist_label = mean(stigma_dist_ppt))
```

```
## # A tibble: 2 x 2
##   label                                stigma_dist_label
##   <chr>                                <dbl>
## 1 people who experience housing insecurity      1.98
## 2 the homeless                                2.25
```

```
## stigma_danger
data_main_recode %>%
  group_by(label) %>%
  summarise(stigma_danger_label = mean(stigma_danger))
```

```
## # A tibble: 2 x 2
##   label                                stigma_danger_label
##   <chr>                                <dbl>
## 1 people who experience housing insecurity      2.04
## 2 the homeless                                2.61
```

Stereotype: PC vs. PIC

```
## average stertyp
data_main_recode %>%
  mutate(stertyp_avg_ppt = rowSums(select(., starts_with('stertyp_')))/8) %>%
  group_by(label) %>%
  summarise(stertyp_avg_label = mean(stertyp_avg_ppt))
```

```
## # A tibble: 2 x 2
##   label                                stertyp_avg_label
##   <chr>                                <dbl>
## 1 people who experience housing insecurity      3.37
## 2 the homeless                                3.51
```

```
## stertyp_cmptition
data_main_recode %>%
  mutate(stertyp_cmptition_ppt = rowSums(select(., starts_with('stertyp_cmptition_')))/2) %>%
  group_by(label) %>%
  summarise(stertyp_cmptition_label = mean(stertyp_cmptition_ppt))
```

```
## # A tibble: 2 x 2
##   label                                stertyp_cmptition_label
##   <chr>                                <dbl>
## 1 people who experience housing insecurity      3.90
## 2 the homeless                                4.10
```

```
## stertyp_comp
data_main_recode %>%
  mutate(stertyp_comp_ppt = rowSums(select(., starts_with('stertyp_comp_')))/2) %>%
  group_by(label) %>%
  summarise(stertyp_comp_label = mean(stertyp_comp_ppt))
```

```
## # A tibble: 2 x 2
##   label                                stertyp_comp_label
##   <chr>                                <dbl>
## 1 people who experience housing insecurity      3.27
## 2 the homeless                                3.33
```

```
## stertyp_warm
data_main_recode %>%
  mutate(stertyp_warm_ppt = rowSums(select(., starts_with('stertyp_warm_')))/2) %>%
  group_by(label) %>%
  summarise(stertyp_warm_label = mean(stertyp_warm_ppt))
```

```
## # A tibble: 2 x 2
##   label                                stertyp_warm_label
##   <chr>                                <dbl>
## 1 people who experience housing insecurity      2.56
## 2 the homeless                                2.74
```

```
## stertyp_status
data_main_recode %>%
  mutate(stertyp_status_ppt = rowSums(select(., starts_with('stertyp_status_')))/2) %>%
  group_by(label) %>%
  summarise(stertyp_status_label = mean(stertyp_status_ppt))
```

```
## # A tibble: 2 x 2
##   label                                stertyp_status_label
##   <chr>                                <dbl>
## 1 people who experience housing insecurity      3.74
## 2 the homeless                                3.89
```

```
# Average donation: PC vs. PIC
data_main_recode %>%
  group_by(label) %>%
  summarise(donate_homeless_label = mean(donate_1))
```

```
## # A tibble: 2 x 2
##   label                                donate_homeless_label
##   <chr>                                <dbl>
## 1 people who experience housing insecurity      40.3
## 2 the homeless                                42.1
```

```
# concept_1, concept_2, concept_3
data_main_recode %>%
  group_by(label) %>%
  summarise(concept_shelter_label = mean(concept_1))
```

```
## # A tibble: 2 x 2
##   label                                concept_shelter_lable
##   <chr>                                <dbl>
## 1 people who experience housing insecurity    3.04
## 2 the homeless                               2.94
```

```
data_main_recode %>%
  group_by(label) %>%
  summarise(concept_unemploy_lable = mean(concept_2))
```

```
## # A tibble: 2 x 2
##   label                                concept_unemploy_lable
##   <chr>                                <dbl>
## 1 people who experience housing insecurity    2.67
## 2 the homeless                               3.18
```

```
data_main_recode %>%
  group_by(label) %>%
  summarise(concept_duration_lable = mean(concept_3))
```

```
## # A tibble: 2 x 2
##   label                                concept_duration_lable
##   <chr>                                <dbl>
## 1 people who experience housing insecurity    3.26
## 2 the homeless                               3.32
```

```
# Average like, offenses: if distinct from 1.5
data_main_recode %>%
  summarise(like_avg_ppt = mean(like),
            offense_avg_ppt = mean(offens))
```

```
## # A tibble: 1 x 2
##   like_avg_ppt offense_avg_ppt
##   <dbl>         <dbl>
## 1      1.5      1.77
```

```
## like, offenses: if correlated with label
data_main_recode %>%
  group_by(label) %>%
  summarise(like_avg_label = mean(like),
            offense_avg_label = mean(offens))
```

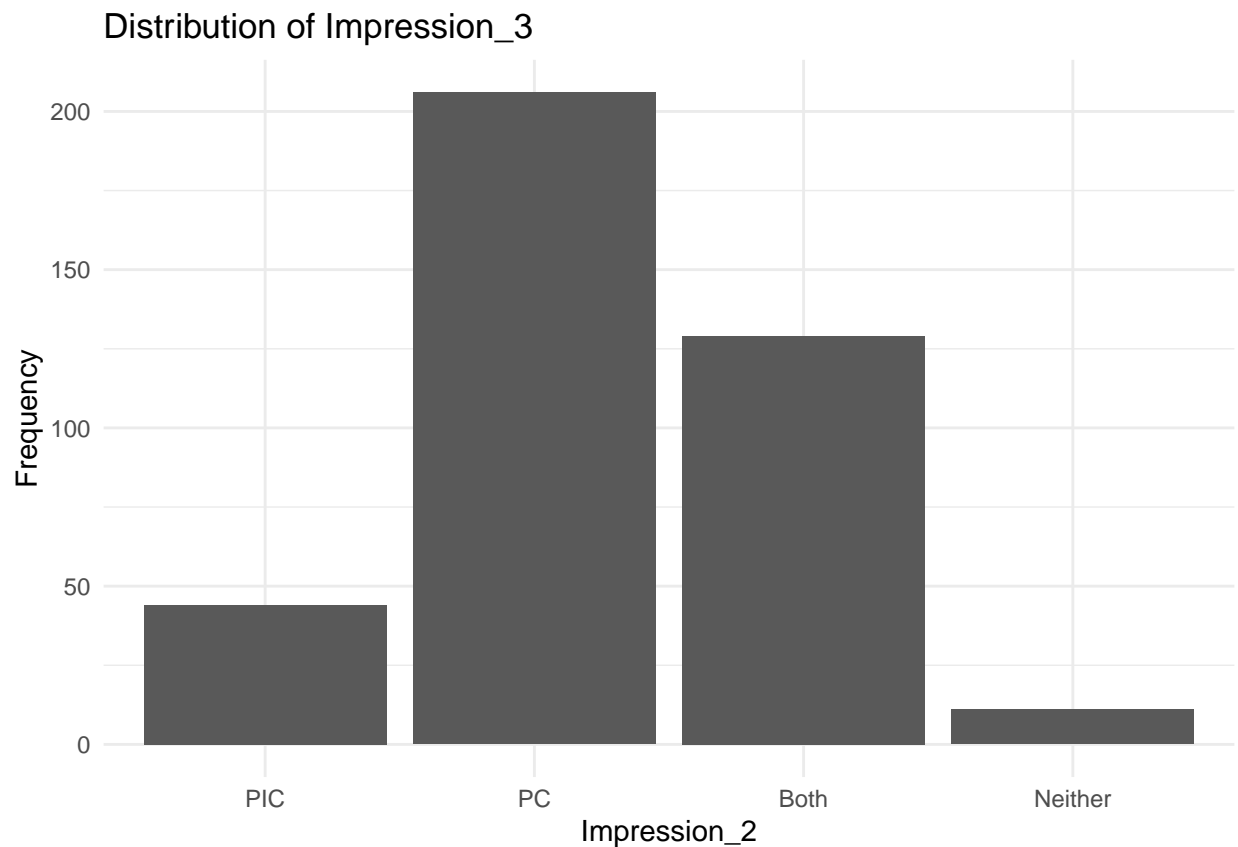
```
## # A tibble: 2 x 3
##   label                                like_avg_label offense_avg_label
##   <chr>                                <dbl>         <dbl>
## 1 people who experience housing insecurity    1.59         1.84
## 2 the homeless                               1.41         1.70
```

```
# Distribution of impression_3
data_main_recode %>%
  summarise(impression_3_avg = mean(impression_3))
```

```
## # A tibble: 1 x 1
##   impression_3_avg
##             <dbl>
## 1             2.27
```

```
data_main_recode$impression_3 <- factor(data_main$impression_3,
                                       levels = c(1, 2, 3, 4),
                                       labels = c("PIC", "PC", "Both", "Neither"))

ggplot(data_main_recode, aes(x = impression_3)) +
  geom_bar() +
  labs(title = "Distribution of Impression_3",
       x = "Impression_2",
       y = "Frequency") +
  theme_minimal()
```



Analysis - F Test

```
# Calculate stigma and stereotype as averages
data_main_recode <- data_main_recode %>%
  mutate(
    stigma_avg = rowMeans(select(., starts_with("stigma_")), na.rm = TRUE),
    stereotype_avg = rowMeans(select(., starts_with("stertyp_")), na.rm = TRUE)
```



```
)

# Perform F-tests for each variable
# 1. Stigma
stigma_aov <- aov(stigma_avg ~ label, data = data_main_recode)
summary(stigma_aov)

##              Df Sum Sq Mean Sq F value    Pr(>F)
## label          1    4.32    4.315    8.245 0.00431 **
## Residuals     388 203.06    0.523
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# 2. Stereotype
stereotype_aov <- aov(stereotype_avg ~ label, data = data_main_recode)
summary(stereotype_aov)

##              Df Sum Sq Mean Sq F value    Pr(>F)
## label          1    2.05    2.0482    7.433 0.00669 **
## Residuals     388 106.91    0.2755
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# 3. Supportive Behavior
supportive_aov <- aov(donate_1 ~ label, data = data_main_recode)
summary(supportive_aov)

##              Df Sum Sq Mean Sq F value    Pr(>F)
## label          1    303    302.6    0.705  0.402
## Residuals     388 166470    429.0
```

Analysis - Multinomial Logistic Regression

```
# Model 1: Stata

# Model 2: Cleaning
## political orientation
data_main_recode <- data_main_recode %>%
  mutate(political = rowMeans(select(., c(political_social, political_econ)), na.rm = TRUE))

## label preference
### like: 1 = the homeless/PIC, 2 = people who experience housing insecurity/PC

## experience: 1 = have personal or close experience, 0 = don't have experience
data_main_recode <- data_main_recode %>%
  mutate(experience = ifelse(experience_1 == 1 | experience_2 == 1, 1, 0))

# Replace NA with 0 in 'experience'
data_main_recode$experience[is.na(data_main_recode$experience)] <- 0
```

```
## empathy
data_main_recode <- data_main_recode %>%
  mutate(empathy = rowMeans(select(., starts_with("empathy_")), na.rm = TRUE))

data_main_noFR <- data_main_recode %>%
  select(-c(general_1, general_2, impression_1, impression_2, impression_3_4_TEXT, like_fr, offens_fr))
write.csv(data_main_noFR, "data_main_noFR.csv", row.names = FALSE)
```