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)

data <- read_csv("PC_Pilot-English_November 19, 2024_10.12.csv")
```

```
## Rows: 33 Columns: 74
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (74): 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(ID) == 24, # Prolific IDs
  )
```

```r
#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())
```

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

write.csv(data_demo, "demo_11082024.csv", row.names = FALSE)

# Main data
```

```r
data_main <- data_passed1 %>%
  select(-c(ID, #de-identified
            starts_with("Demo_"),
            comment)) %>%
  mutate(across(where(~ all(!is.na(as.numeric(.)))), as.numeric))
```

```
## Warning: There were 6 warnings in `mutate()`.
## The first warning was:
## i In argument: `across(where(~all(!is.na(as.numeric(.)))), as.numeric)`.
## Caused by warning in `predicate()`:
## ! NAs introduced by coercion
## i Run `dplyr::last_dplyr_warnings()` to see the 5 remaining warnings.
```

Analysis - Simple

```r
# Overlap compare
data_main %>%
  group_by(label) %>%
  summarise(overlap_avg = mean(SliderOverlapValue))
```

```
## # A tibble: 2 x 2
##   label                                    overlap_avg
##   <chr>                                          <dbl>
## 1 people who experience housing insecurity        60
## 2 the homeless                                    58.4
```

```r
# Stigma: PC vs. PIC

## 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))

## 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             1.67
## 2 the homeless                                         2.5
```

```r
## 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               2.1
## 2 the homeless                                           2.3
```

```
## 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              1.53
## 2 the homeless                                          2.67
```

```
## 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                 1.2
## 2 the homeless                                             2.4
```

```
# Stereotype: PC vs. PIC

## average stertyp
data_main %>%
  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              2.65
## 2 the homeless                                          2.45
```

```
## stertyp_cmptition
data_main %>%
  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                     1.6
## 2 the homeless                                                 2.3
```

```
## stertyp_comp
data_main %>%
  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
## 2 the homeless                                            2.4
```

```
## stertyp_warm
data_main %>%
  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                3.8
## 2 the homeless                                            3
```

```
## stertyp_status
data_main %>%
  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                  2.2
## 2 the homeless                                              2.1
```

```
# Average donation: PC vs. PIC
data_main %>%
  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                  35.2
## 2 the homeless                                              45
```

```r
# concept_1, concept_2, concept_3
data_main %>%
  group_by(label) %>%
  summarise(concept_shelter_lable = mean(concept_1))
```

```
## # A tibble: 2 x 2
##   label                                    concept_shelter_lable
##   <chr>                                                    <dbl>
## 1 people who experience housing insecurity                   2.2
## 2 the homeless                                               3
```

```r
data_main %>%
  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                   2.2
## 2 the homeless                                               2.8
```

```r
data_main %>%
  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                   2.8
## 2 the homeless                                               3.4
```

```r
# Average like, offens: if distinct from 1.5
data_main %>%
  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          1.6             1.9
```

```r
## like, offens: if correlated with label
data_main %>%
  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            1.8               2
## 2 the homeless                                        1.4               1.8
```

```r
# Distribution of impression_2
data_main %>%
  summarise(impression_2_avg = mean(impression_2))
```

```
## # A tibble: 1 x 1
##   impression_2_avg
##              <dbl>
## 1              2.4
```

```r
data_main$impression_2 <- factor(data_main$impression_2,
                                 levels = c(1, 2, 3, 4),
                                 labels = c("PIC", "PC", "Both", "Neither"))

ggplot(data_main, aes(x = impression_2)) +
  geom_bar() +
  labs(title = "Distribution of Impression_2",
       x = "Impression_2",
       y = "Frequency") +
  theme_minimal()
```