W3

Celina Jang

2025-06-25

Week 3

```
chs2020_raw <- read.csv('chs2020_clean_raw.csv')

# new cleaned dataset with additional variables
chs2020_cleaned <- chs2020_raw |>
   # Only using those who answered version 1 survey
  filter(!is.na(strata_q1)) |>

  select(strata_q1, survey, mood1, mood2, mood3, mood4, mood5, mood6, helpneighbors20_q1,
         discussissues, helpcommproj, trustkeys, proudneigh, agegroup,
         birthsex, newrace6, education, employment20, insured, usborn, maritalstatus20,
         sexualid20, ipvphy, bmi, didntgetcare20, fruitveg20, generalhealth, pcp20,
         skiprxcost, insure5, imputed_povgroup3, imputed_pov200,
         imputed_povertygroup, hhsize, delaypayrent, rodentsstreet, k6, nspd,
         wt21_dual_q1
         ) |>
  # recoding demographic variables
  mutate(
    age_band = case_when (
      agegroup == 1 | agegroup==2 ~ '18-44',
      agegroup == 3 ~ '45-64',
      agegroup == 4 ~ '65+'
    ) ,
    gender = as.factor(case_when (
      birthsex==1 ~ 'male',
      birthsex==2 ~ 'female'
    )),
    race_ethnicity = as.factor(case_when (
      newrace6 == 1 ~ 'White',
      newrace6 == 2 ~ 'Black',
      newrace6 == 3 ~ 'Hispanic',
      newrace6 == 4 ~ 'Asian/Pacific Islander',
      newrace6 == 5 ~ 'North African/Mid Eastern',
      newrace6 == 6 ~ 'Other'
    ))
  )


# compute k6 total and social cohesion score
```

```r
chs2020_cleaned$k6_total <-
  rowSums(chs2020_cleaned[, c("mood1", "mood2", "mood3", "mood4", "mood5", "mood6")],
                            na.rm = TRUE)

sc_items <- c("helpneighbors20_q1", "discussissues","helpcommproj", "trustkeys","proudneigh")

for(i in 1:5) {
  chs2020_cleaned <-
    chs2020_cleaned |>
    mutate(
      helpneighbors20_q1_rev = case_when(helpneighbors20_q1==1 ~ 5,
                                         helpneighbors20_q1==2 ~ 4,
                                         helpneighbors20_q1==3 ~ 3,
                                         helpneighbors20_q1==4 ~ 2,
                                         helpneighbors20_q1==5 ~ 1,
                                         is.na(helpneighbors20_q1) ~ NA),
      discussissues_rev = case_when(discussissues==1 ~ 4,
                                    discussissues==2 ~ 3,
                                    discussissues==3 ~ 2,
                                    discussissues==4 ~ 1,
                                    is.na(discussissues) ~ NA),
      helpcommproj_rev = case_when(helpcommproj==1 ~ 4,
                                   helpcommproj==2 ~ 3,
                                   helpcommproj==3 ~ 2,
                                   helpcommproj==4 ~ 1,
                                   is.na(helpcommproj) ~ NA),
      trustkeys_rev = case_when(trustkeys==1 ~ 4,
                                trustkeys==2 ~ 3,
                                trustkeys==3 ~ 2,
                                trustkeys==4 ~ 1,
                                is.na(trustkeys) ~ NA),
      proudneigh_rev = case_when(proudneigh==1 ~ 4,
                                 proudneigh==2 ~ 3,
                                 proudneigh==3 ~ 2,
                                 proudneigh==4 ~ 1,
                                 is.na(proudneigh) ~ NA)
    )
}


chs2020_cleaned$social_cohesion_rev<- rowMeans(chs2020_cleaned[, c("helpneighbors20_q1_rev",
                                               "discussissues_rev",
                                               "helpcommproj_rev",
                                               "trustkeys_rev",
                                               "proudneigh_rev")], na.rm = TRUE)
chs2020_cleaned$social_cohesion <- rowMeans(chs2020_cleaned[, c("helpneighbors20_q1",
                                               "discussissues",
                                               "helpcommproj",
                                               "trustkeys",
                                               "proudneigh")], na.rm = TRUE)

# save cleaned dataset as new csv file
write.csv(chs2020_cleaned, "chs2020_working_w3.csv", row.names = FALSE)
```

Construct Validity

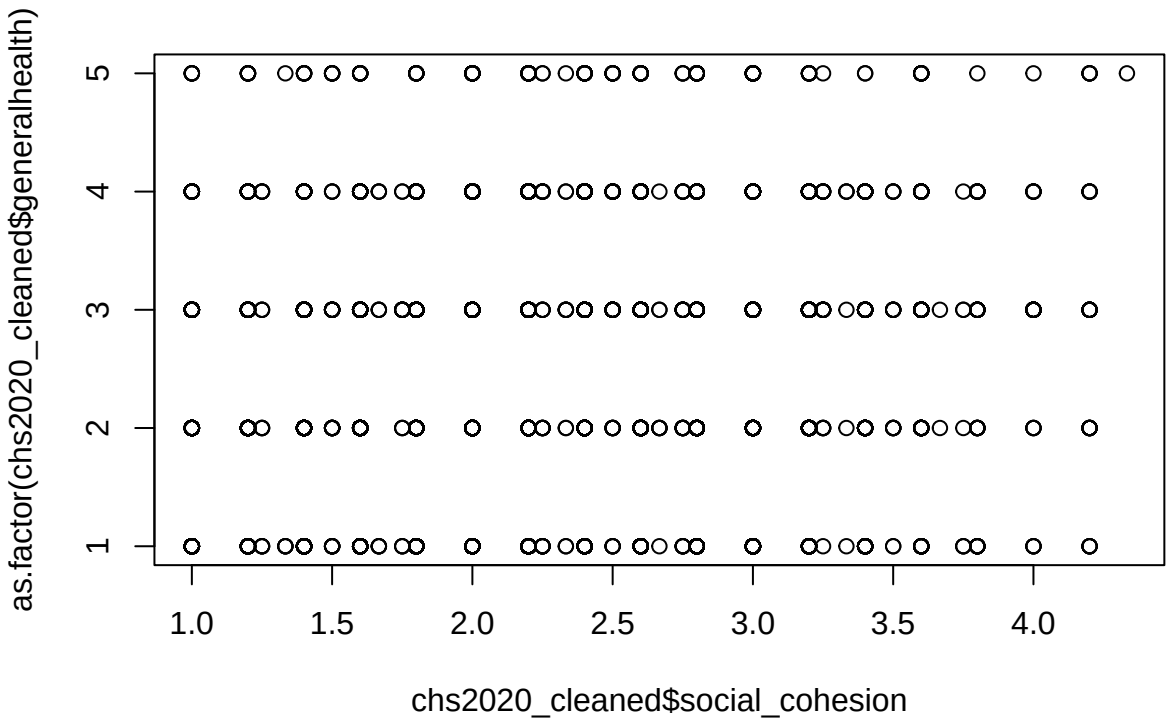
```
# general health
cor(chs2020_cleaned$social_cohesion, chs2020_cleaned$generalhealth, use = 'complete.obs', method = 'spea
```

```
## [1] 0.06383923
```

```
summary(lm(generalhealth ~ social_cohesion, data=chs2020_cleaned))
```

```
##
## Call:
## lm(formula = generalhealth ~ social_cohesion, data = chs2020_cleaned)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7079 -0.5800 -0.3883  0.5531  2.6330
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.26039    0.05586  40.467  < 2e-16 ***
## social_cohesion  0.10655    0.02341   4.551 5.49e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.09 on 4317 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.004774,   Adjusted R-squared:  0.004544
## F-statistic: 20.71 on 1 and 4317 DF,  p-value: 5.495e-06
```

```
plot(y=as.factor(chs2020_cleaned$generalhealth), x=chs2020_cleaned$social_cohesion)
```




```
#fruitveg20
cor(chs2020_cleaned$social_cohesion, chs2020_cleaned$fruitveg20, use = 'complete.obs', method = 'pearso
```

```
## [1] -0.1272561
```

```
summary(lm(fruitveg20 ~ social_cohesion, data=chs2020_cleaned))
```

```
##
## Call:
## lm(formula = fruitveg20 ~ social_cohesion, data = chs2020_cleaned)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10836 -0.06089 -0.01341  0.04989  1.14484
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.187485   0.022576  96.896   <2e-16 ***
## social_cohesion -0.079125   0.009467  -8.358   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4367 on 4244 degrees of freedom
##   (90 observations deleted due to missingness)
## Multiple R-squared:  0.01619,    Adjusted R-squared:  0.01596
```

```
## F-statistic: 69.86 on 1 and 4244 DF,  p-value: < 2.2e-16
```

```
#delaypayrent
cor(chs2020_cleaned$social_cohesion, chs2020_cleaned$delaypayrent, use = 'complete.obs', method = 'pears
```

```
## [1] -0.06127506
```

```
chs2020_cleaned<- chs2020_cleaned |> mutate(delaypayrent0 = case_when(
  delaypayrent == 1 ~ 0,
  delaypayrent == 2 ~ 1,
  is.na(delaypayrent) ~ NA
))

rent_glm <- glm(delaypayrent0 ~ social_cohesion, data=chs2020_cleaned, family = binomial)
exp(-0.23553  )
```

```
## [1] 0.790152
```

```
summary(rent_glm)
```

```
##
## Call:
## glm(formula = delaypayrent0 ~ social_cohesion, family = binomial,
##     data = chs2020_cleaned)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      2.23453    0.14496  15.415  < 2e-16 ***
## social_cohesion -0.23553    0.05882  -4.004 6.22e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3714.1  on 4289  degrees of freedom
## Residual deviance: 3698.2  on 4288  degrees of freedom
##   (46 observations deleted due to missingness)
## AIC: 3702.2
##
## Number of Fisher Scoring iterations: 4
```

```
#didntgetcare20
cor(chs2020_cleaned$social_cohesion, chs2020_cleaned$didntgetcare20, use = 'complete.obs', method = 'pea
```

```
## [1] -0.0452685
```

```
chs2020_cleaned<- chs2020_cleaned |> mutate(didntgetcare0 = case_when(
  didntgetcare20 == 1 ~ 0,
  didntgetcare20 == 2 ~ 1,
  is.na(didntgetcare20) ~ NA
```

```
))

care_glm<- glm(didntgetcare0 ~ social_cohesion, data=chs2020_cleaned, family = binomial)
exp(-0.19354)
```

```
## [1] 0.8240369
```

```
summary(care_glm)
```

```
##
## Call:
## glm(formula = didntgetcare0 ~ social_cohesion, family = binomial,
##     data = chs2020_cleaned)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       2.42901    0.16031  15.152  < 2e-16 ***
## social_cohesion -0.19354    0.06518  -2.969  0.00299 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3187.0  on 4313  degrees of freedom
## Residual deviance: 3178.2  on 4312  degrees of freedom
##   (22 observations deleted due to missingness)
## AIC: 3182.2
##
## Number of Fisher Scoring iterations: 4
```

```
# -------------------------REVERSE CODED--------------------------------


# general health
chs2020_cleaned<- chs2020_cleaned |> mutate(generalhealth_r = case_when(
  generalhealth == 1 ~ 5,
  generalhealth == 2 ~ 4,
  generalhealth == 3 ~ 3,
  generalhealth == 4 ~ 2,
  generalhealth == 5 ~ 1,
  is.na(generalhealth) ~ NA
))

cor(chs2020_cleaned$social_cohesion_rev, chs2020_cleaned$generalhealth_r, use = 'complete.obs', method =
```

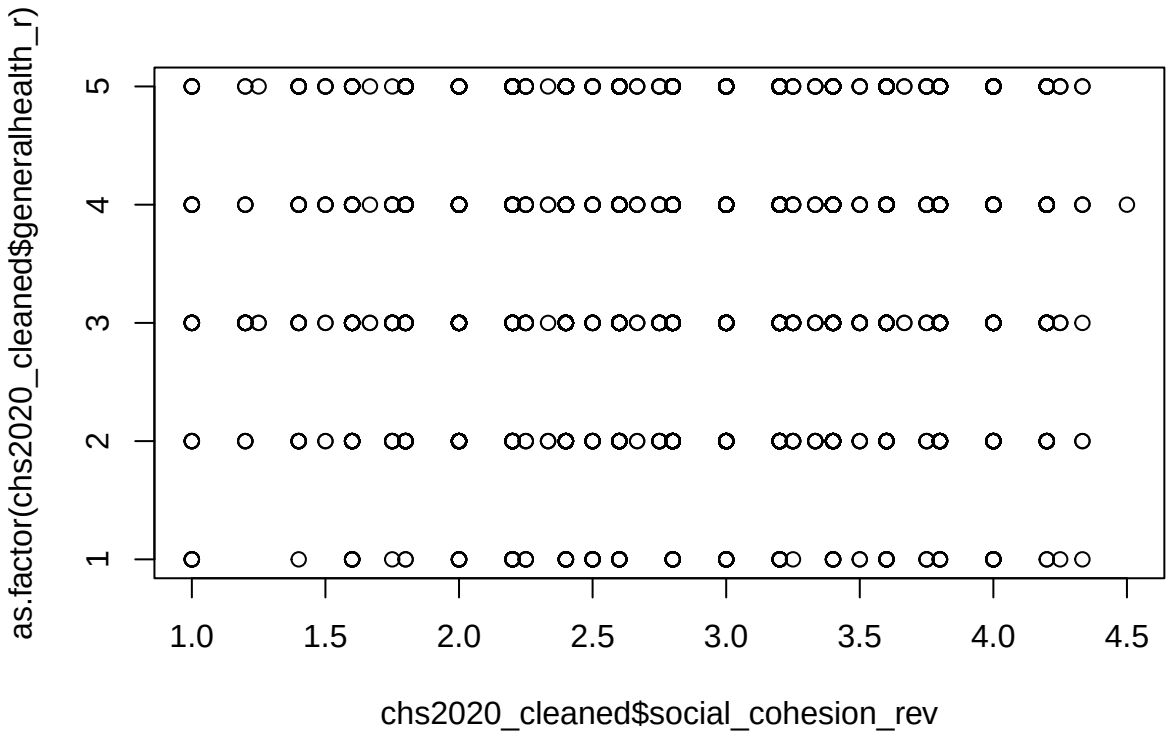
```
## [1] 0.06647859
```

```
summary(lm(generalhealth_r ~ social_cohesion_rev, data=chs2020_cleaned))
```

```
##
## Call:
```

```
## lm(formula = generalhealth_r ~ social_cohesion_rev, data = chs2020_cleaned)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6532 -0.5611  0.3837  0.5825  1.7151
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.17439    0.07003   45.33  < 2e-16 ***
## social_cohesion_rev  0.11049    0.02331    4.74 2.21e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.09 on 4317 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.005177,   Adjusted R-squared:  0.004946
## F-statistic: 22.46 on 1 and 4317 DF,  p-value: 2.21e-06
```

```
plot(y=as.factor(chs2020_cleaned$generalhealth_r), x=chs2020_cleaned$social_cohesion_rev)
```




```
#fruitveg20
cor(chs2020_cleaned$social_cohesion_rev, chs2020_cleaned$fruitveg20,
    use = 'complete.obs', method = 'pearson')
```

```
## [1] 0.1278015
```

```r
summary(lm(fruitveg20 ~ social_cohesion_rev, data=chs2020_cleaned))
```

```
##
## Call:
## lm(formula = fruitveg20 ~ social_cohesion_rev, data = chs2020_cleaned)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.11260 -0.06117 -0.01370  0.04564  1.14453
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.776349   0.028317  62.732   <2e-16 ***
## social_cohesion_rev 0.079117   0.009425   8.395   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4367 on 4244 degrees of freedom
##   (90 observations deleted due to missingness)
## Multiple R-squared:  0.01633,    Adjusted R-squared:  0.0161
## F-statistic: 70.47 on 1 and 4244 DF,  p-value: < 2.2e-16
```

```r
#delaypayrent
cor(chs2020_cleaned$social_cohesion_rev, chs2020_cleaned$delaypayrent, use = 'complete.obs', method = '
```

```
## [1] 0.06158955
```

```r
chs2020_cleaned<- chs2020_cleaned |> mutate(delaypayrent0 = case_when(
  delaypayrent == 1 ~ 1,
  delaypayrent == 2 ~ 0,
  is.na(delaypayrent) ~ NA
))

rent_glm <- glm(delaypayrent0 ~ social_cohesion_rev, data=chs2020_cleaned, family = binomial)
exp(-0.23553)
```

```
## [1] 0.790152
```

```r
summary(rent_glm)
```

```
##
## Call:
## glm(formula = delaypayrent0 ~ social_cohesion_rev, family = binomial,
##     data = chs2020_cleaned)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.01055    0.17137  -5.897  3.7e-09 ***
## social_cohesion_rev -0.23558    0.05853  -4.025  5.7e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3714.1  on 4289  degrees of freedom
## Residual deviance: 3698.0  on 4288  degrees of freedom
##   (46 observations deleted due to missingness)
## AIC: 3702
##
## Number of Fisher Scoring iterations: 4
```

```
exp(coef(rent_glm))
```

```
##         (Intercept) social_cohesion_rev
##           0.3640191           0.7901134
```

```
#didntgetcare20
chs2020_cleaned<- chs2020_cleaned |> mutate(didntgetcare0 = case_when(
  didntgetcare20 == 1 ~ 1,
  didntgetcare20 == 2 ~ 0,
  is.na(didntgetcare20) ~ NA
))
cor(chs2020_cleaned$social_cohesion_rev, chs2020_cleaned$didntgetcare20,
    use = 'complete.obs', method = 'pearson')
```

```
## [1] 0.04498655
```

```
care_glm<- glm(didntgetcare0 ~ social_cohesion_rev, data=chs2020_cleaned, family = binomial)
exp(-0.19153)
```

```
## [1] 0.8256949
```

```
summary(care_glm)
```

```
##
## Call:
## glm(formula = didntgetcare0 ~ social_cohesion_rev, family = binomial,
##     data = chs2020_cleaned)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.42903    0.19041  -7.505 6.15e-14 ***
## social_cohesion_rev -0.19153    0.06491  -2.951  0.00317 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3187.0  on 4313  degrees of freedom
## Residual deviance: 3178.3  on 4312  degrees of freedom
##   (22 observations deleted due to missingness)
## AIC: 3182.3
##
## Number of Fisher Scoring iterations: 4
```

Criterion validity

```
cor(chs2020_cleaned$social_cohesion_rev, chs2020_cleaned$k6, use = 'complete.obs', method = 'pearson')
```

```
## [1] -0.09647054
```

```
summary(lm(k6 ~ social_cohesion ,data=chs2020_cleaned))
```

```
##
## Call:
## lm(formula = k6 ~ social_cohesion, data = chs2020_cleaned)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.428 -3.361 -1.124  1.994 20.469
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.93863    0.22057  13.323  < 2e-16 ***
## social_cohesion  0.59279    0.09245   6.412 1.59e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.308 on 4328 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.009409,   Adjusted R-squared:  0.009181
## F-statistic: 41.11 on 1 and 4328 DF,  p-value: 1.592e-10
```

Summary TABLE A (MAYBE)

```
library(dplyr)
library(purrr)

# Recode helper
recode_var <- function(data, var) {
  # Only add age_band if it doesn't exist
  if (!"age_band" %in% names(data) && "agegroup" %in% names(data)) {
    data <- data |> mutate(
      age_band = case_when(
        agegroup == 1 | agegroup == 2 ~ '18-44',
        agegroup == 3 ~ '45-64',
        agegroup == 4 ~ '65+',
        TRUE ~ NA_character_
      )
    )
  }
  if (var == "education") {
    data <- data |> mutate(
```

```
      education_lev = case_when(
        education == 1 ~ 'Less than high school',
        education == 2 ~ 'High school graduate',
        education == 3 ~ 'Some college/technical school',
        education == 4 ~ 'College graduate',
        education == '.d' ~ 'Dont know',
        education == '.r' ~ 'Refused',
        TRUE ~ NA_character_
      )
    )
  } else if (var == "employment20") {
    data <- data |> mutate(
      employment = case_when(
        employment20 == 1 ~ 'Employed for wages or salary',
        employment20 == 2 ~ 'Self-employed',
        employment20 == 3 ~ 'Unemployed for 1 year or more',
        employment20 == 4 ~ 'Unemployed for less than 1 year',
        employment20 == 5 ~ 'A homemaker',
        employment20 == 6 ~ 'A student',
        employment20 == 7 ~ 'Retired',
        employment20 == 8 ~ 'Unable to work',
        employment20 == '.d' ~ 'Dont know',
        employment20 == '.r' ~ 'Refused',
        TRUE ~ NA_character_
      )
    )
  } else if (var == "imputed_povertygroup") {
    data <- data |> mutate(
      poverty_level = case_when(
        imputed_povertygroup == 1 ~ '<100% poverty',
        imputed_povertygroup == 2 ~ '100% - <200% poverty',
        imputed_povertygroup == 3 ~ '200% - <400% poverty',
        imputed_povertygroup == 4 ~ '400% - <600% poverty',
        imputed_povertygroup == 5 ~ '>=600% poverty',
        TRUE ~ NA_character_
      )
    )
  }
  return(data)
}

# Variables to summarize
vars_to_summarize <- list(
  age_band = "ageband",
  gender = "gender",
  race_ethnicity = "race_ethnicity",
  education = "education",
  employment20 = "employment",
  imputed_povertygroup = "poverty_group",
  hhsize = "householdsize"
)

chs2020_cleaned_recoded <- recode_var(chs2020_cleaned, "education")
```

```r
chs2020_cleaned_recoded <- recode_var(chs2020_cleaned_recoded, "employment20")
chs2020_cleaned_recoded <- recode_var(chs2020_cleaned_recoded, "imputed_povertygroup")

# Now get_summary just uses this data as is, without recoding:
get_summary <- function(data, var_name, display_name) {

  # Map var_name to actual column name after recode if needed
  col_name <- if(var_name %in% c("education", "employment20", "imputed_povertygroup")) {
    if(var_name == "education") "education_lev" else
    if(var_name == "employment20") "employment" else
    if(var_name == "imputed_povertygroup") "poverty_level"
  } else var_name

  # Remove NAs in col_name
  data_clean <- data |> filter(!is.na(.data[[col_name]]))

  if (col_name == "age_band") {
    overall <- data_clean |>
      count(age_band) |>
      rename(level = age_band) |>
      mutate(
        percent = round(n / sum(n) * 100, 2),
        ageband = "Overall",
        variable = display_name
      )

    by_ageband <- overall |>
      mutate(ageband = level)

  } else {
    overall <- data_clean |>
      count(!!sym(col_name)) |>
      mutate(
        percent = round(n / sum(n) * 100, 2),
        ageband = "Overall",
        variable = display_name
      ) |>
      rename(level = !!sym(col_name))|>
      mutate(level = as.character(level))

    by_ageband <- data_clean |>
      group_by(age_band, !!sym(col_name)) |>
      count() |>
      ungroup() |>
      group_by(age_band) |>
      mutate(percent = round(n / sum(n) * 100, 2)) |>
      mutate(variable = display_name) |>
      rename(level = !!sym(col_name), ageband = age_band)|>
      mutate(level = as.character(level))
  }

  combined <- bind_rows(overall, by_ageband) |>
    select(variable, ageband, level, count = n, percent)
```

```
  return(combined)
}

# Then run map2_dfr with pre-recoded data:
summary_all <- map2_dfr(
  names(vars_to_summarize),
  vars_to_summarize,
  ~ get_summary(chs2020_cleaned_recoded, .x, .y)
)
# View result
print(summary_all)

if (!dir.exists("summary_tables")) {
  dir.create("summary_tables")
}
summary_all %>%
  split(.$variable) %>%               # split into list by variable
  imap(~ write.csv(.x,
                   file = paste0("summary_tables/", .y, "_summary.csv"),
                   row.names = FALSE))



colnames(chs2020_cleaned)
```

Descriptive analysis of key variables

```
## k6_total
chs2020_cleaned |>
  ggplot(aes(x=k6_total)) + geom_boxplot()
summary(chs2020_cleaned$k6_total)

## nspd
chs2020_cleaned |>
  mutate(nspd_response = case_when(nspd==1 ~ 'Yes', nspd==2 ~ 'No', is.na(nspd) ~ NA)) |>
  group_by(nspd_response, ageband) |>
  count() |>
  summarize(
    count=n,
    .groups='drop'
    ) |>
  group_by(ageband) |>
  mutate(proportion = count / sum(count)*100)

## ageband
chs2020_cleaned |>
  count(age_band) |>
  summarize(
    ageband=age_band,
    percent = n/sum(n) *100,
```

```
    count=n)

## usborn
chs2020_cleaned |>
  count(usborn) |>
  summarize(
    usborn=usborn,
    percent = n/sum(n) *100,
    count=n)

## gender
chs2020_cleaned |>
  count(gender) |>
  summarize(
    gender=gender,
    percent = n/sum(n) *100,
    count=n)

chs2020_cleaned |>
  group_by(gender, age_band) |>
  count() |>
  summarize(
    count=n,
    .groups='drop'
    ) |>
  group_by(age_band) |>
  mutate(proportion_by_age = count / sum(count)*100)


## race/ethnicity
chs2020_cleaned |>
  count(race_ethnicity) |>
  summarize(
    race_ethnicity=race_ethnicity,
    percent = round(n/sum(n) *100,3),
    count=n)

chs2020_cleaned |>
  group_by(race_ethnicity, age_band) |>
  count() |>
  summarize(
    count=n,
    .groups='drop'
    ) |>
  group_by(age_band) |>
  mutate(proportion_by_age = count / sum(count)*100)

## education
chs2020_cleaned |>
  mutate(
    education_lev =
      case_when(education==1 ~ 'Less than high school',
                education==2 ~ 'High school graduate',
```

```
                education==3 ~ 'Some college/technical school',
                education==4 ~ 'College graduate',
                education=='.d' ~ 'Dont know',
                education=='.r' ~ 'Refused',
                is.na(education) ~ NA)
    ) |>
  count(education_lev) |>
  summarize(
    education=education_lev,
    percent = round(n/sum(n) *100,3),
    count=n)

chs2020_cleaned |>
   mutate(
    education_lev =
      case_when(education==1 ~ 'Less than high school',
                education==2 ~ 'High school graduate',
                education==3 ~ 'Some college/technical school',
                education==4 ~ 'College graduate',
                education=='.d' ~ 'Dont know',
                education=='.r' ~ 'Refused',
                is.na(education) ~ NA)
    ) |>
  group_by(education_lev, age_band) |>
  count() |>
  summarize(
    count=n,
    .groups='drop'
    ) |>
  group_by(age_band) |>
  mutate(proportion_by_age = count / sum(count)*100)


## employment20
chs2020_cleaned |>
  mutate(
    employment =
      case_when(employment20==1 ~ 'Employed for wages or salary',
                employment20==2 ~ 'Self-employed',
                employment20==3 ~ 'Unemployed for 1 year or more',
                employment20==4 ~ 'Unemployed for less than 1 year',
                employment20==5 ~ 'A homemaker',
                employment20==6 ~ 'A student',
                employment20==7 ~ 'Retired',
                employment20==8 ~ 'Unable to work',
                employment20=='.d' ~ 'Dont know',
                employment20=='.r' ~ 'Refused',
                is.na(employment20) ~ NA)
    )|>
  count(employment) |>
  summarize(
    employment=employment,
    percent = round(n/sum(n) *100,3),
```

```
    count=n)

chs2020_cleaned |>
   mutate(
    employment =
      case_when(employment20==1 ~ 'Employed for wages or salary',
                employment20==2 ~ 'Self-employed',
                employment20==3 ~ 'Unemployed for 1 year or more',
                employment20==4 ~ 'Unemployed for less than 1 year',
                employment20==5 ~ 'A homemaker',
                employment20==6 ~ 'A student',
                employment20==7 ~ 'Retired',
                employment20==8 ~ 'Unable to work',
                employment20=='.d' ~ 'Dont know',
                employment20=='.r' ~ 'Refused',
                is.na(employment20) ~ NA)
    )|>
  group_by(employment, age_band) |>
  count() |>
  summarize(
    count=n,
    .groups='drop'
    ) |>
  group_by(age_band) |>
  mutate(proportion_by_age = count / sum(count)*100)

## imputed poverty group
chs2020_cleaned |>
  mutate(
    poverty_level =
      case_when(imputed_povertygroup==1 ~ '<100% poverty',
                imputed_povertygroup==2 ~ '100% - <200% poverty',
                imputed_povertygroup==3 ~ '200% - <400% poverty',
                imputed_povertygroup==4 ~ '400% - <600% poverty',
                imputed_povertygroup==5~ '>=600% poverty',
                is.na(imputed_povertygroup) ~ NA)
    )|>
  count(poverty_level) |>
  summarize(
    poverty_group=poverty_level,
    percent =round( n/sum(n) *100,3),
    count=n)

chs2020_cleaned |>
   mutate(
    poverty_level =
      case_when(imputed_povertygroup==1 ~ '<100% poverty',
                imputed_povertygroup==2 ~ '100% - <200% poverty',
                imputed_povertygroup==3 ~ '200% - <400% poverty',
                imputed_povertygroup==4 ~ '400% - <600% poverty',
                imputed_povertygroup==5~ '>=600% poverty',
                is.na(imputed_povertygroup) ~ NA)
    )|>
```

```
  group_by(poverty_level, age_band) |>
  count() |>
  summarize(
    count=n,
    .groups='drop'
    ) |>
  group_by(age_band) |>
  mutate(proportion_by_age = count / sum(count)*100)

## hhsize
chs2020_cleaned |>
  count(hhsize) |>
  summarize(
    householdsize=hhsize,
    percent = round(n/sum(n) *100,3),
    count=n)

## delaypayrent
chs2020_cleaned |>
  mutate(delaypayrent_response = case_when(
    delaypayrent == 1 ~ 'Yes',
    delaypayrent == 2 ~ 'No',
    delaypayrent == '.d' ~ 'Dont know',
    delaypayrent == '.r' ~ 'Refused'
  )) |>
  count(delaypayrent_response) |>
  summarize(
    delaypayrent_response=delaypayrent_response,
    percent = round(n/sum(n) *100,3),
    count=n)

## rodentsstreet
chs2020_cleaned |>
  mutate(rodent = case_when(
    rodentsstreet == 1 ~ 'Yes',
    rodentsstreet == 2 ~ 'No',
    rodentsstreet == '.d' ~ 'Dont know',
    rodentsstreet == '.r' ~ 'Refused'
  )) |>
  count(rodent) |>
  summarize(
    rodentsstreet_response=rodent,
    percent =round( n/sum(n) *100,3),
    count=n)

chs2020_cleaned |>
   mutate(rodent = case_when(
    rodentsstreet == 1 ~ 'Yes',
    rodentsstreet == 2 ~ 'No',
    rodentsstreet == '.d' ~ 'Dont know',
    rodentsstreet == '.r' ~ 'Refused'
  ))|>
  group_by(rodent, age_band) |>
```

```
  count() |>
  summarize(
    count=n,
    .groups='drop'
    ) |>
  group_by(age_band) |>
  mutate(proportion_by_age = count / sum(count)*100)

## smellcigsmoke20_q1
chs2020_cleaned |>
  mutate(smellcig = case_when(
    smellcigsmoke20_q1 == 1 ~ 'Everyday',
    smellcigsmoke20_q1 == 2 ~ 'A few times per week',
    smellcigsmoke20_q1 == 3 ~ 'A few times per month',
    smellcigsmoke20_q1 == 4 ~ 'A few times per year',
    smellcigsmoke20_q1 == 5 ~ 'Never',
    smellcigsmoke20_q1 == '.d' ~ 'Dont know',
    smellcigsmoke20_q1 == '.r' ~ 'Refused',
    is.na(smellcigsmoke20_q1) ~ NA
  )) |>
  count(smellcig) |>
  summarize(
    smellcigsmoke=smellcig,
    percent =round(n/sum(n) *100,3),
    count=n)

chs2020_cleaned |>
   mutate(smellcig = case_when(
    smellcigsmoke20_q1 == 1 ~ 'Everyday',
    smellcigsmoke20_q1 == 2 ~ 'A few times per week',
    smellcigsmoke20_q1 == 3 ~ 'A few times per month',
    smellcigsmoke20_q1 == 4 ~ 'A few times per year',
    smellcigsmoke20_q1 == 5 ~ 'Never',
    smellcigsmoke20_q1 == '.d' ~ 'Dont know',
    smellcigsmoke20_q1 == '.r' ~ 'Refused',
    is.na(smellcigsmoke20_q1) ~ NA
  )) |>
  group_by(smellcig, age_band) |>
  count() |>
  summarize(
    count=n,
    .groups='drop'
    ) |>
  group_by(age_band) |>
  mutate(proportion_by_age = count / sum(count)*100)
```

```
library(dplyr)
library(knitr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
summarize_variable_clean <- function(data, var, var_label) {
  tab <- data %>%
    count(!!sym(var)) %>%
    filter(!is.na(!!sym(var))) %>%
    mutate(
      percent = round(n / sum(n) * 100, 2),
      Variable = var_label,
      Category = str_to_title(as.character(!!sym(var))),
      n = n
    ) %>%
    select(Variable, Category, n, percent) %>%
    arrange(desc(percent))  # <- Sort by percent

  # Blank out the repeated variable label (except first)
  tab$Variable[-1] <- ""
  tab
}

gender_table <- summarize_variable_clean(chs2020_cleaned, "gender", "Gender")
ageband_table <- summarize_variable_clean(chs2020_cleaned, "age_band", "Age Band")
race_table <- summarize_variable_clean(chs2020_cleaned, "race_ethnicity", "Race/Ethnicity")
hhsize_table <- summarize_variable_clean(chs2020_cleaned, "hhsize", "Household Size")

# Recoded: Usborn
birthplace_table <- chs2020_cleaned |>
  mutate(usborn = case_when(
    usborn == 1 ~ 'US Born',
    usborn == 2 ~ 'Foreign Born',
    TRUE ~ NA_character_
  ))  %>%
  summarize_variable_clean("usborn", "Birthplace")

# Recoded: Education
education_table <- chs2020_cleaned %>%
  mutate(education = case_when(
    education == 1 ~ 'Less than high school',
    education == 2 ~ 'High school graduate',
    education == 3 ~ 'Some college/technical school',
    education == 4 ~ 'College graduate',
    education == ".d" ~ 'Don't know',
    education == ".r" ~ 'Refused',
    TRUE ~ NA_character_
  )) %>%
  summarize_variable_clean("education", "Education")

# Recoded: Employment
employment_table <- chs2020_cleaned %>%
  mutate(employment = case_when(
    employment20 == 1 ~ 'Employed for wages or salary',
    employment20 == 2 ~ 'Self-employed',
```

```
    employment20 == 3 ~ 'Unemployed 1+ year',
    employment20 == 4 ~ 'Unemployed <1 year',
    employment20 == 5 ~ 'Homemaker',
    employment20 == 6 ~ 'Student',
    employment20 == 7 ~ 'Retired',
    employment20 == 8 ~ 'Unable to work',
    employment20 == ".d" ~ 'Don't know',
    employment20 == ".r" ~ 'Refused',
    TRUE ~ NA_character_
  )) %>%
  summarize_variable_clean("employment", "Employment")

# Recoded: Poverty Level
poverty_table <- chs2020_cleaned %>%
  mutate(poverty_level = case_when(
    imputed_povertygroup == 1 ~ '<100% poverty',
    imputed_povertygroup == 2 ~ '100% - <200% poverty',
    imputed_povertygroup == 3 ~ '200% - <400% poverty',
    imputed_povertygroup == 4 ~ '400% - <600% poverty',
    imputed_povertygroup == 5 ~ '>=600% poverty',
    TRUE ~ NA_character_
  )) %>%
  summarize_variable_clean("poverty_level", "Poverty Group")


summary_table <- bind_rows(
  gender_table,
  birthplace_table,
  ageband_table,
  race_table,
  education_table,
  employment_table,
  poverty_table,
  hhsize_table
)
```

```r
summary_table <- summary_table |>
  rename('%' = percent)

summary_table %>%
  kable(caption = "Demographic Distribution") %>%
  kable_styling(full_width = FALSE)
```

Table 1: Demographic Distribution

Variable	Category	n	%
Gender	Female	2436	56.38
	Male	1885	43.62
Birthplace	Us Born	2323	53.81
	Foreign Born	1994	46.19
Age Band	18-44	1906	44.05
	45-64	1466	33.88
	65+	955	22.07
Race/Ethnicity	White	1367	31.53
	Hispanic	1231	28.39
	Black	907	20.92
	Asian/Pacific Islander	644	14.85
	Other	144	3.32
	North African/Mid Eastern	43	0.99
Education	College Graduate	1987	46.05
	High School Graduate	891	20.65
	Some College/Technical School	833	19.30
	Less Than High School	604	14.00
Employment	Employed For Wages Or Salary	1951	45.29
	Retired	743	17.25
	Unemployed <1 Year	451	10.47
	Self-Employed	378	8.77
	Unable To Work	312	7.24
	Student	180	4.18
	Homemaker	165	3.83
	Unemployed 1+ Year	128	2.97
Poverty Group	<100% Poverty	1020	23.52
	>=600% Poverty	986	22.74
	100% - <200% Poverty	854	19.70
	200% - <400% Poverty	755	17.41
	400% - <600% Poverty	721	16.63
Household Size	1	1112	25.65
	2	1105	25.48
	3	720	16.61
	4	674	15.54
	5	384	8.86
	6	194	4.47
	7	147	3.39

TABLE B

```
library(dplyr)
library(knitr)
library(kableExtra)

summary_B <- tibble(
  Variable = c("General Health", "Fruit/Veg Intake", "Delayed Rent", "Unmet Care"),
  Correlation = c(
    cor(chs2020_cleaned$social_cohesion_rev, chs2020_cleaned$generalhealth_r,
        use = "complete.obs", method = "spearman"),
    cor(chs2020_cleaned$social_cohesion_rev, chs2020_cleaned$fruitveg20,
        use = "complete.obs", method = "pearson"),
    cor(chs2020_cleaned$social_cohesion_rev, chs2020_cleaned$delaypayrent0,
        use = "complete.obs", method = "pearson"),
    cor(chs2020_cleaned$social_cohesion_rev, chs2020_cleaned$didntgetcare0,
        use = "complete.obs", method = "pearson")),
  Regression_Coefficient = c(
    coef(summary(lm(generalhealth_r ~ social_cohesion_rev, data=chs2020_cleaned)))[2, "Estimate"],
    coef(summary(lm(fruitveg20 ~ social_cohesion_rev, data=chs2020_cleaned)))[2, "Estimate"],
    coef(summary(glm(delaypayrent0 ~ social_cohesion_rev, data=chs2020_cleaned,
                     family=binomial)))[2, "Estimate"],
    coef(summary(glm(didntgetcare0 ~ social_cohesion_rev, data=chs2020_cleaned,
                     family=binomial)))[2, "Estimate"]),
  Odds_Ratio = c(
    NA, NA,
    exp(coef(summary(glm(delaypayrent0 ~ social_cohesion_rev, data=chs2020_cleaned,
                         family=binomial)))[2, "Estimate"]),
    exp(coef(summary(glm(didntgetcare0 ~ social_cohesion_rev, data=chs2020_cleaned,
                         family=binomial)))[2, "Estimate"])),
  pValue = c(
    coef(summary(lm(generalhealth_r ~ social_cohesion_rev, data=chs2020_cleaned)))[2, "Pr(>|t|)"],
    coef(summary(lm(fruitveg20 ~ social_cohesion_rev, data=chs2020_cleaned)))[2, "Pr(>|t|)"],
    coef(summary(glm(delaypayrent0 ~ social_cohesion_rev, data=chs2020_cleaned,
                     family=binomial)))[2, "Pr(>|z|)"],
    coef(summary(glm(didntgetcare0 ~ social_cohesion_rev, data=chs2020_cleaned,
                     family=binomial)))[2, "Pr(>|z|)"])
)

summary_B <- summary_B %>%
  # Round numeric columns except Odds_Ratio and pValue first
  mutate(
    Correlation = round(Correlation, 3),
    Regression_Coefficient = round(Regression_Coefficient, 3),
    Odds_Ratio = ifelse(
      is.na(Odds_Ratio),
      ".",                              # placeholder for the first two
      sprintf("%.3f", Odds_Ratio)       # round and format as string
    ),
    pValue = sapply(pValue, function(p) {
      if      (is.na(p))      NA_character_
      else if (p < 1e-10)     "<1e-10"
      else sprintf("%.3f", p)
```

```
    })
  ) %>%
  rename(
    `Odds Ratio` = Odds_Ratio,
    "$\\beta$" = Regression_Coefficient,  # unicode beta
     p = pValue
  )

# print with kable
summary_B %>%
  kable(caption = "Construct Validity", format='markdown',escape = FALSE,
        col.names = c(
      "Variable",
      "Correlation",
      "Beta",             # your beta column
      "Odds Ratio",
      "p"
    )) %>%
  kable_styling(full_width=F)
```

Table 2: Construct Validity

Variable	Correlation	Beta	Odds Ratio	p
General Health	0.066	0.110	.	0.000
Fruit/Veg Intake	0.128	0.079	.	<1e-10
Delayed Rent	-0.062	-0.236	0.790	0.000
Unmet Care	-0.045	-0.192	0.826	0.003

```
#write.csv(summary_B, "construct_validity_table.csv", row.names = FALSE, na = "")
```

TABLE C

```
summary_C <- tibble(
  Correlation = c(
  cor(chs2020_cleaned$social_cohesion_rev, chs2020_cleaned$k6,
      use = 'complete.obs', method = 'pearson') ),
  Regression_Coefficient = c(
    coef(summary(lm(k6 ~ social_cohesion_rev,data=chs2020_cleaned)))[2, "Estimate"]),
  pValue=c(
    coef(summary(lm(k6 ~ social_cohesion_rev ,data=chs2020_cleaned)))[2, "Pr(>|t|)"]),
  )


summary_C %>%
  mutate(pValue=ifelse((pValue < 1e-10), "<1e-10", sprintf("%.3f", pValue))) |>
  kable(caption = "Criterion Validity (Relationship between K6 and Social Cohesion)",
        escape = FALSE,
        col.names = c(
```

```
    "Correlation",
    "Beta",            # your beta column
    "p"), digits=3)
```

Table 3: Criterion Validity (Relationship between K6 and Social Cohesion)

Correlation	Beta	p
-0.096	-0.587	0.000