CRRA

1. Using the CRRA utility function:

$$U(x) = \begin{cases} \frac{x^{1-r}}{1-r} & \text{if} \quad r \neq 1 \\ \ln(x) & \text{if} \quad r = 1 \end{cases}$$

   where x is the payoff and r is the CRRA coefficient.

2. For each lottery, the participant should choose the lottery if the expected utility of the lottery exceeds the utility of the sure payment:

$$0.5 \times U(lottery_{high}) + 0.5 \times U(lottery_{low}) > U(10)$$

3. Using maximum likelihood estimation approach and assume a logistic error model for choice stochasticity:

$$P(choose\ lottery) = \frac{1}{1 + exp(-\lambda \cdot (EU_{lottery} - U_{sure})}$$

4. Maximizing the log-likelihood over all trials for each participant to estimate r.

Load necessary libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
library(haven)
```

Upload the data

```
setwd("~/Desktop/Stata/CRRA")
df <- read_dta("new_data.dta")
```

Shift all payoffs (e.g. +21, so the minimum is 1)

```r
offset <- 21
df <- df %>%
  mutate(
    lot_low_pos = lottery_low + offset,
    lot_high_pos = lottery_high + offset,
    sure_pos = sure + offset
  )
```

CRRA utility function

```r
crra_utility <- function(x, r) {
  if (abs(r - 1) < 1e-6) {
    return(log(x))
  } else {
    return((x^(1 - r) - 1) / (1 - r))
  }
}
```

Negative log-likelihood function for one participant

```r
neg_log_lik <- function(r, data, lambda = 1) {
  # Avoid negative or zero values by ensuring positive payoffs
  if (any(data$lot_low_pos <= 0) || any(data$lot_high_pos <= 0) || any(data$sure_pos <= 0)) return(Inf)
  # Calculate expected utility difference
  util_lot <- 0.5 * crra_utility(data$lot_high_pos, r) + 0.5 * crra_utility(data$lot_low_pos, r)
  util_sure <- crra_utility(data$sure_pos, r)
  diff <- util_lot - util_sure
  p_lottery <- 1 / (1 + exp(-lambda * diff))
  # Avoid log(0)
  p_lottery <- pmin(pmax(p_lottery, 1e-8), 1 - 1e-8)
  # Log-likelihood
  loglik <- data$choice * log(p_lottery) + (1 - data$choice) * log(1 - p_lottery)
  return(-sum(loglik))
}
```

Estimate CRRA for each participant

```r
results <- df %>%
  group_by(id) %>%
  group_modify(~{
    res <- optim(par = 0.5, fn = neg_log_lik, data = ., method = "Brent", lower = -2, upper = 2)
    tibble(crra = res$par, negloglik = res$value)
  })
```

Results

```
print(results)
```

```
## # A tibble: 300 x 3
## # Groups:   id [300]
##    id          crra negloglik
##    <dbl+lbl> <dbl>     <dbl>
##  1  1 [S001] 0.320      9.83
##  2  2 [S002] 0.320      9.83
##  3  3 [S003] 0.317      9.57
##  4  4 [S004] 0.351     11.1 
##  5  5 [S005] 0.314      9.23
##  6  6 [S006] 0.320      9.83
##  7  7 [S007] 0.324     10.2 
##  8  8 [S008] 0.324     10.2 
##  9  9 [S009] 0.317      9.57
## 10 10 [S010] 0.315      9.37
## # i 290 more rows
```