0.1 Variable documentation

0.1.1 Variables from generic model

- **Scripts:** ultra_main(um), main(m), game(g), headsUp/headsUp2(h), adjustCardValue(acv)

- **playerP1, playerP2, general:** Vector with 4 entries representing all important variables from one player.

  1. risk factor for player
  2. capital from player
  3. card value from player, is a random number between 0 and 1
  4. total bet from player
  5. free variable for learning implementation

- **allData um:** Matrix containing number of Wins from player 1 one for varying risk factors

- **r1, r2 um:** iteration variables representing risk factors for players 1 and 2

- **betValue m,h:** represents the amount a player can bet on his win or the amount by which the pot can be increased per round per person

- **n m:** iteration variable for determining how many games are being simulated, hence determining the accuracy of the monte carlo approach

- **riskfactorP1, riskFactorP2 m:** variables representing the risk factors for both players

- **startCapital m:** determines the capital at the beginning of the game for both players.

- **winsP1 m:** amount of total wins by player 1, used as output to the function main.m

- **winner m,g:** stores the the winner of the game simulated: if 0 winner is player 2, if 1 winner is player 1, used as output to the function game.m

- **counter g:** counts amount of hands played in one game

- **decide_who_starts g:** used to determine whose turn it is to start with betting, if 0 player 2 begins, if 1 player 1 begins

- **pot h:** stores the total amount of money betted by both players

- **capP1, capP2 :** output variables to headsUp.m function storing the capital of the corresponding player after having played the hand

- **newRandValue acv:** output to adjustCardValue.m function, stores the newly generated cardvalue

- **sigma acv:** theoretically the standard deviation of a normally distributed random value, here used to determine the range of adjustement for the function adjustCardValue.m

0.1.2 Variables from learning models

Threshold model

- **Scripts:** ultra_main(um), main(m), game(g), headsUp/headsUp2(h), adjustCardValue(acv), adjustRiskFactor(arf)

- **totalCounter m:** used to store total amount of hands played per game

- **playerP1(5) g:** stores the risk factor of player 2 as it is currently estimated by player 1

- **startRiskFactor h:** input to headsup.m function, stores the risk factor from player 2 as estimated by player 1 before the respective hand

- **estRiskFactor h:** output from headsup.m function, stores the risk factor from player 2 as estimated by player 1 after the respective hand

- **newRiskFactor arf:** output from adjustRiskFactor.m function, stores newly generated risk factor

- **opponentRiskFactor arf:** input to function adjustRiskFactor.m, currently estimated risk factor from opponent player

- **refSurf arf:** two-variable function which represents amount of wins for player 1 in dependence of the respective risk factors

- **funVector arf:** parameterisation of refSurf at point of opponentRiskFactor