# Intro

Cribbage is a card game credited to be created in the early 17th century (Wikipedia)

## Rules/sections of the game

The game is comprised of three sections: the deal, the play, the show

The deal is where the cards are dealt, discarded and the communal card cut from the deck.

The play is where cards are played alternately by each player in order for the values of the cards to reach certain totals or score runs and pairs.

The show is where the players hands are scored in conjunction with the communal cut card as well as a further hand called the crib, which is comprised of the discarded cards from the players hands, that is added to the dealers points.

Further information can be found at <https://en.wikipedia.org/wiki/Cribbage>

# Aims of this program

* Create a program that advises on which two cards to discard in order to achieve the ‘best’ hand.
* Use as an exercise to increase familiarity with R.
* Develop better coding practices with better documentation and testing.

## What this is

A tool to show various statistical outputs based on all the combinations of hands and cut cards possible from a dealt hand

## What this is not

A tool that takes into consideration the play or crib aspects of the show.

A functional Cribbage game.

# Rules of the show

In the show the remaining cards from the original hand are scored in conjunction with the cut card, though this card can sometimes act differently from the cards in the hand.

The rules for scoring the hand are as follows:

## 15’s

Every combination of cards that adds up to 15 scores 2 points. Cards can be used multiple times but the combinations of all sets must be unique i.e. a hand with one 8 and two 7's can have to sets of 15's one for each 7 with the 8.

## Pairs

Every combination of pairs will be awarded 2 points. Each card can be used multiple times but the combination of both cards must be unique i.e. a hand with 3 7's will score 6 points; 2 for each combination of pairs.

## Flush

To score a flush each card in the hand (excluding the cut card) must be of the same suit and will score 4 points if this condition is met. If the cut card is also the same suit then an extra point will be awarded.

## Runs

Runs are any combination of 3 or more cards in sequential order and score one point for each card. cards can be used multiple times but the combination to make the runs must be unique i.e. a hand that has the cards 2h, 3h, 3s, 4h, 9h can have 2 runs of:

2h, 3h, 4h,

2h, 3s, 4h

A further rule is that runs must be the maximum length that they can be i.e. a run of 1,2,3,4 cannot be split into 3 runs of:

1,2,3

2,3,4

1,2,3,4

## One for his nob

If a hand has a jack that has the same suit as the cut card then an extra point is awarded

# Program functionality

The function will be started using the optimise\_hand function and will produce different results depending on the input.

The inputs can be empty or a data\_frame/table with two columns:

Value: 1L:13 to represent the card value

Suit: c, d, h, s to represent the suit of the card

The output of the function will be as follows:

Optimise\_hand() create a random hand and calculate all combinations of discards and cut cards

Optimise\_hand(hand with 4 cards) calculate all combination of cut cards with the

Optimise\_hand(hand with 5 cards) calculate the score of the hand assuming the final card is the cut card

Optimise\_hand(hand with 6 cards) calculate all combinations of discards and cut cards

# Future developments

Once the aims of this project have been achieved further ideas could be implemented in following versions such as:

Shiny implementation

Creating a functional game simulator