Playing_Cards

Saurav Singh

17 January 2017

Playing Cards

First I have imported **deck.csv** file which have playing cards data in format:

```
deck=read.table("deck.csv",sep = ",",header = TRUE,stringsAsFactors = FALSE)
head(deck,10)
```

```
##     face   suit value
## 1   king spades    13
## 2  queen spades    12
## 3   jack spades    11
## 4    ten spades    10
## 5   nine spades     9
## 6  eight spades     8
## 7  seven spades     7
## 8    six spades     6
## 9   five spades     5
## 10  four spades     4
```

Next, I have setup function which does the following things:

- Saves **deck** to **DECK**,so as to preserve the original copy.
- Declares DEAL() function which deals the card on the top and removes it from the deck.
- Declares SHUFFLE() function which shuffles the deck.

Finally,setting up the deck accordingly.

```
setup<-function(deck){
DECK<-deck

DEAL<-function(){
  card<-deck[1,]
  assign("deck",deck[-1,],envir = parent.env(environment()))
  card
}

SHUFFLE<-function(){
  random<-sample(1:52,size=52)
  assign("deck",DECK[random,],envir = parent.env(environment()))
}

list(deal=DEAL,shuffle=SHUFFLE)
}
cards<-setup(deck)
deal<-cards$deal
shuffle<-cards$shuffle
```

Game Of War

Arranging a deck according to Game of War.
(i.e giving all Aces a value of 14.)

```
deck2<-deck
deck2$value[c(13,26,39,52)]<-14
```

The above can only work if we have the original arranged set of playing cards.
If we had shuffled deck then:

```
deck3<-shuffle()
deck3$value[deck3$face=="ace"]<-14
```

Hearts

Arranging a deck according to the Game of Hearts.
(i.e setting the value of each suit of hearts as 1, for Queen of Spades the value will be 13 and for rest all the value will be 0.)

```
deck4<-deck
deck4$value<-0
deck4$value[deck4$suit=="hearts"]<-1
queenOfSpades<-deck4$face=="queen" & deck4$suit=="spades"
deck4$value[queenOfSpades]<-13
```

BlackJack

Arranging a deck according to the Game of BlackJack. (i.e setting the value of each facecard as 10 and for aces the value is NA as its value depends on the outcome of the game.)

```
deck5<-deck
facecard<-deck5$face %in% c("king","queen","jack")
deck5$value[facecard]<-10
deck5$value[deck5$face=="ace"]<-NA
```

Environments

For understanding environments I have used the following function:

```
show_env<-function(){
  a=15
  b=6
  c=4
  list(ran.in=environment(),
       parent=parent.env(environment()),
       objects=ls.str(environment()))
}
show_env()
```

```
## $ran.in
## <environment: 0x46ccaa0>
##
```

```
## $parent
## <environment: R_GlobalEnv>
##
## $objects
## a :  num 15
## b :  num 6
## c :  num 4
```

It shows the reference of the running environment created while calling show_env() function.Also, shows the parent of the environment i.e Global Environment.Lastly, it tells us about how functions behave, as everytime when they are called creates a running environment in which these variable a,b,c are with-held.
Understanding environments was important because of which I was able to create SHUFFLE() and DEAL() function.