Chapter_6

Gavin McCorry

2024-04-10

Environments

Loading Deck

```
deck <- read.csv("C:/Users/gwmcc/OneDrive/Documents/GitHub/Data-332/Chapter-3/cards.csv")
Deck <- read.csv("C:/Users/gwmcc/OneDrive/Documents/GitHub/Data-332/Chapter-3/cards.csv")
```

Shuffle Function from Chapter 4

```
shuffle <- function(cards) {
  random <- sample(c(1:52), size = 52)
  cards[random,]
}
```

Deal Function

```
deal <- function(cards){
  cards[1, ]
}
```

Vieing environment

```
parenvs(all = T)
```

```
##    label                            name
## 1  <environment: R_GlobalEnv>       ""
## 2  <environment: package:pryr>      "package:pryr"
## 3  <environment: package:stats>     "package:stats"
## 4  <environment: package:graphics>  "package:graphics"
## 5  <environment: package:grDevices> "package:grDevices"
## 6  <environment: package:utils>     "package:utils"
## 7  <environment: package:datasets>  "package:datasets"
## 8  <environment: package:methods>   "package:methods"
## 9  <environment: 0x00000250b8df3218> "Autoloads"
## 10 <environment: base>              ""
## 11 <environment: R_EmptyEnv>        ""
```

Working With Environments

```
as.environment("package:stats")
```

```
## <environment: package:stats>
## attr(,"name")
## [1] "package:stats"
## attr(,"path")
## [1] "C:/Program Files/R/R-4.2.1/library/stats"
```

```
globalenv()
```

```
## <environment: R_GlobalEnv>
```

```
baseenv()
```

```
## <environment: base>
```

```
emptyenv()
```

```
## <environment: R_EmptyEnv>
```

```
# look up environments parnts
parent.env(globalenv())
```

```
## <environment: package:pryr>
## attr(,"name")
## [1] "package:pryr"
## attr(,"path")
## [1] "C:/Users/gwmcc/AppData/Local/R/win-library/4.2/pryr"
```

```
# view saved objects in an environment
ls(emptyenv())
```

```
## character(0)
```

```
ls(globalenv())
```

```
## [1] "deal"    "deck"    "Deck"    "shuffle"
```

Active Environments

```
# see current activ environment
environment()
```

```
## <environment: R_GlobalEnv>
```

Evaluation

```
# R runs your functions in a runtime environment then returns the results back to your environment
# See here:
show_env <- function(){
  list(ran.in = environment(),
       parent = parent.env(environment()),
       objects = ls.str(environment())
  )
}
# Runs in different environment each time
show_env()
```

```
## $ran.in
## <environment: 0x00000250bde2b590>
##
## $parent
## <environment: R_GlobalEnv>
##
## $objects
```

Exercise:

```
# redefine the deal function by taking out the parameter
# Deal still works cuz deck is in the global environment

deal <- function() {
  deck[1, ]
}
deal()
```

```
##   face   suit value
## 1 king spades    13
```

Exercise 2:

```
# mak it so that after dealing card is removed from the deck

deal <- function(){
  card <- deck[1, ]
  assign("deck", deck[-1, ], envir = globalenv())
  card
}

deal()
```

```
##   face   suit value
## 1 king spades    13
```

```
deal()
```

```
##    face   suit value
## 2 queen spades    12
```

```
deal()
```

```
##   face   suit value
## 3 jack spades    11
```

Exercise 3:

```
# Rewrite shuffle so that it replaces the copy of deck that lives in the global environment with a shuf

shuffle <- function(){
  random <- sample(1:52, size = 52)
  assign("deck", Deck[random, ], envir = globalenv())
}

shuffle()
```

Closure

```
shuffle()
deal()
```

```
##     face  suit value
## 19 eight clubs     8
```

```
deal()
```

```
##    face   suit value
## 47  six hearts     6
```

```
# this would be better if we could store deck in a safe and out of the way place

setup <- function(deck){
  DEAL <- deck

  DEAL <- function(){
  card <- deck[1, ]
  assign("deck", deck[-1, ], envir = globalenv())
  card
```

```
  }

  SHUFFLE <- function(){
  random <- sample(1:52, size = 52)
  assign("deck", Deck[random, ], envir = globalenv())
  }

  list(deal = DEAL, shuffle = SHUFFLE)
}

cards <- setup(deck)
deal <- cards$deal
shuffle <- cards$shuffle

# finally have complete card game
```