

# Chapter 5

Gavin McCorry

2024-03-18

## Chapter 5: Modifying Values

### Loading Deck

```
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,]  
}
```

### Changing Values In Place

```
vec <- c(0, 0, 0, 0, 0, 0)  
vec
```

```
## [1] 0 0 0 0 0 0
```

```
# replacing on value  
vec[1] <- 1000  
vec
```

```
## [1] 1000 0 0 0 0 0
```

```
# replacing multiple values  
vec[c(1, 3, 5)] <- c(1, 1, 1)  
vec
```

```
## [1] 1 0 1 0 1 0
```

```
# crating new values
vec[7] <- 0
vec
```

```
## [1] 1 0 1 0 1 0 0
```

Adding new variables to data set

```
deck$new <- 1:52
head(deck)
```

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

```
# Can also remove variables
deck$new <- NULL
head(deck)
```

```
##   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
```

Working with the deck - Card game War, replacing aces value of 1 to 14

```
# singling out value of the aces
deck[c(13, 26, 39, 52), ]
```

```
##   face   suit value
## 13 ace  spades    1
## 26 ace  clubs     1
## 39 ace diamonds   1
## 52 ace  hearts     1
```

```
# can just get the values
deck$value[c(13, 26, 39, 52)]
```

```
## [1] 1 1 1 1
```

```
# replacing ace value
deck$value[c(13, 26, 39, 52)] <- 14
head(deck, 13)
```

```
##      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
## 11 three spades    3
## 12  two spades     2
## 13   ace spades   14
```

## Logical Subsetting

```
# Logical Opereators Examples:
1 > 2
```

```
## [1] FALSE
```

```
1 > c(0, 1, 2)
```

```
## [1] TRUE FALSE FALSE
```

```
c(1, 2, 3) == c(3, 2, 1)
```

```
## [1] FALSE TRUE FALSE
```

```
# %in% operator
1 %in% c(3, 4, 5)
```

```
## [1] FALSE
```

```
c(1, 2, 3, 4) %in% c(3, 4, 5)
```

```
## [1] FALSE FALSE TRUE TRUE
```

**Exercise:** Extract the face column of `deck` and test whether each value is equal to ace and count how many cards are equal to ace.

```
sum(deck$face == "ace")
```

```
## [1] 4
```

Exercise: Lets put logical subsetting to use with a new gam: hearts. In haearts everey card has a valu of zero:

```
deck4 <- deck
deck4$value <- 0

# Assign a value of 1 to every card in dck4 that has a suit of harts
deck4$value[deck4$suit == "hearts"] <- 1
deck4$value[deck4$suit == "hearts"]
```

```
## [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Boolean Operators: We can use boolean opeerators to find the queen of spades and assign it the value of 13 for the game of hearts

```
queenSpades <- deck4$face == "queen" & deck4$suit == "spades"
deck4$value[queenSpades] <- 13
deck4[queenSpades, ]
```

```
##   face   suit value
## 2 queen spades    13
```

One last game: Black Jack:

```
deck5 <- deck

facecards <- deck5$face %in% c("king", "queen", "jack")
deck5[facecards, ]
```

```
##   face   suit value
## 1  king  spades    13
## 2 queen  spades    12
## 3  jack  spades    11
## 14 king  clubs     13
## 15 queen clubs     12
## 16 jack  clubs     11
## 27 king  diamonds   13
## 28 queen diamonds   12
## 29 jack  diamonds   11
## 40 king  hearts     13
## 41 queen hearts     12
## 42 jack  hearts     11
```

```
deck5$value[facecards] <- 10
head(deck5, 13)
```

```
##      face  suit value
## 1   king spades   10
## 2  queen spades   10
## 3   jack spades   10
## 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
## 11 three spades    3
## 12  two spades     2
## 13   ace spades   14
```

## Missing Informtaion:

```
1 + NA
```

```
## [1] NA
```

```
NA == 1
```

```
## [1] NA
```

**na.rm**

```
c(NA, 1:50)
```

```
## [1] NA  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
## [26] 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
## [51] 50
```

```
# cant get mean regularly
mean(c(NA, 1:50))
```

```
## [1] NA
```

```
# Use special parameters
mean(c(NA, 1:50), na.rm = TRUE)
```

```
## [1] 25.5
```

**is.na**

```
# May want to identify if a value is NA  
NA == NA
```

```
## [1] NA
```

```
# Regular comparison operators don't work so use is.na  
is.na(NA)
```

```
## [1] TRUE
```

Last thing we will finish the blackjack deck by making the values NA since we don't know the final value of the ace

```
deck5$value[deck5$face == "ace"] <- NA  
head(deck5, 13)
```

```
##      face    suit value  
## 1   king spades    10  
## 2  queen spades    10  
## 3   jack spades    10  
## 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  
## 11 three spades     3  
## 12  two spades      2  
## 13   ace spades    NA
```