Probability with Venn Diagrams

Patrick Kelly

June 8, 2016

Probability with Venn Diagrams

Vidéo en français

Probabilité “Diagrammes de Venn”

English video

Probability) “Venn Diagrams”

A standard deck of cards

There are 4 suits: {Spades, Diamonds, Clubs, Hearts}

There are 13 types cards in each suit (rank)

{Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King}

Pick a random card. What is the probability that it is a Jack?

p(Jack) = Number of Jacks / Cards in deck = 4/52 = 1/13 = 0.077 = 7.7%

Place the card back in the deck and shuffle

Pick a random card. What is the probability that it is a Heart?

p(Hearts) = Number of Hearts / Cards in deck = 13/52 = 1/4 = 0.25 = 25%

Place the card back in the deck and shuffle

Pick a random card. What is the probability that it is the Jack of Hearts?

p(Jack of Hearts) = 1/52 = 0.019 = 1.9%

Place the card back in the deck and shuffle

Pick a random card. What is the probability that it is a Jack or Heart?

p(Jack or Hearts) = (4/52 + 13/52 - 1/52) = 4/13 = 0.308 = 30.8%

Let's code this exercise in R

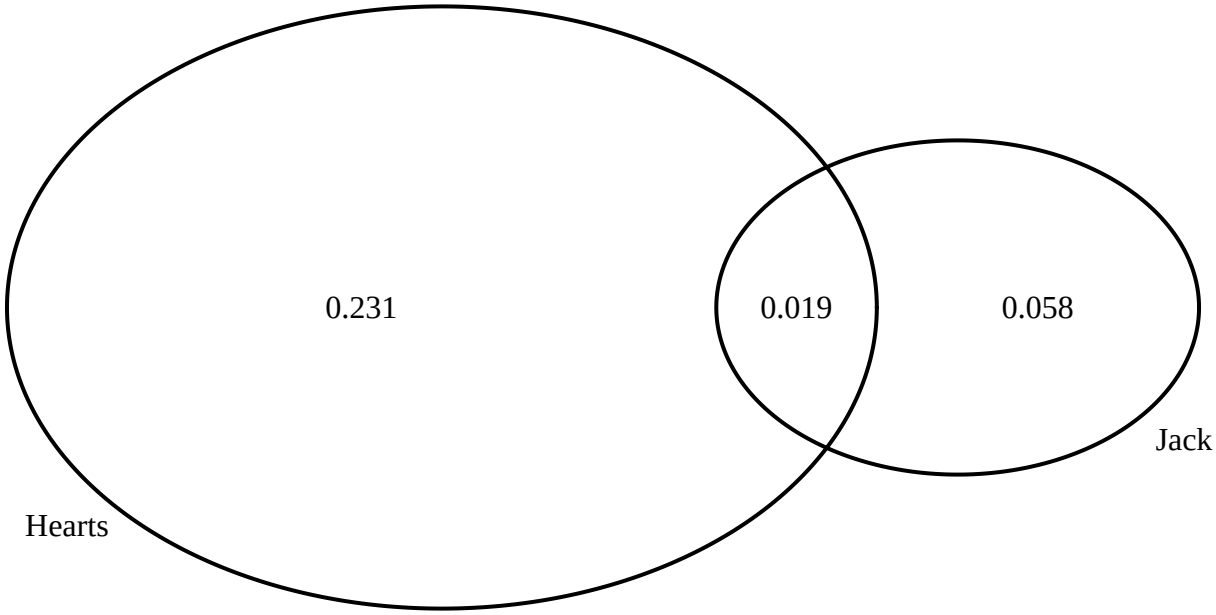
```
# install.packages("VennDiagram")
library(VennDiagram)
```

```
## Warning: package 'VennDiagram' was built under R version 3.2.5
```

```
## Loading required package: grid
```

```
## Loading required package: futile.logger
```

```
grid.newpage()
# Create Venn Diagram
j <- round(1/13,3)
h <- round(1/4,3)
intersection <- round(1/52,3)
draw.pairwise.venn(j, h, intersection,
                   category = c("Jack", "Hearts"))
```




```
## (polygon[GRID.polygon.1], polygon[GRID.polygon.2], polygon[GRID.polygon.3], polygon[GRID.polygon.4],
```

```
# p(Jack or Heart) <- p(Jack) + p(Hearts) - p(Jack of Hearts)
p_j_or_h <- j+h-intersection
p_j_or_h
```

```
## [1] 0.308
```

Now let's create a deck of 52 cards

```
suit <- c("H" ,"C" ,"D", "S")
rank <- c(2:9, "T", "J", "Q", "K", "A")
deck <- NULL #create the deck
for(r in rank){
    deck <- c(deck, paste(r, suit))
}
deck
```

```
##  [1] "2 H" "2 C" "2 D" "2 S" "3 H" "3 C" "3 D" "3 S" "4 H" "4 C" "4 D"
## [12] "4 S" "5 H" "5 C" "5 D" "5 S" "6 H" "6 C" "6 D" "6 S" "7 H" "7 C"
## [23] "7 D" "7 S" "8 H" "8 C" "8 D" "8 S" "9 H" "9 C" "9 D" "9 S" "T H"
## [34] "T C" "T D" "T S" "J H" "J C" "J D" "J S" "Q H" "Q C" "Q D" "Q S"
## [45] "K H" "K C" "K D" "K S" "A H" "A C" "A D" "A S"
```

Create a shuffle function

```
shuffle <- function(deck) {
    return(sample(deck,length(deck)))
}
```

Simulate drawing one card 10,000 times

```
n <- 10000
iterations <- 0
while(iterations < n)  {
    rn <- sample(1:5,1) # Create a random number from 1 to 5
    random <- 0
    while(random < rn){
    s_deck <- shuffle(deck) # Shuffle the deck rn times
    random <- random + 1
    }
    trial <- sample(s_deck, n, replace=TRUE) # Draw a card 10,000 times
    iterations <- iterations + 1 }
iterations
```

```
## [1] 10000
```

```
rn
```

```
## [1] 2
```

Probability of drawing a Jack

```r
p_Jack <- ifelse(grep("J", trial), 1)
p_Jack <- sum(p_Jack) / n
p_Jack
```

```
## [1] 0.0756
```

```r
p_Jack = round(p_Jack * 100, digits = 1)
paste("Probability of drawing a Jack =", p_Jack,"%")
```

```
## [1] "Probability of drawing a Jack = 7.6 %"
```

```r
paste("Probability of drawing a Jack =",
      round(sum(ifelse(grep("J", trial), 1)) / n * 100, digits = 1),"%")
```

```
## [1] "Probability of drawing a Jack = 7.6 %"
```

Probability of drawing a Heart

```r
p_Hearts <- ifelse(grep("H", trial), 1)
p_Hearts <- sum(p_Hearts) / n
p_Hearts
```

```
## [1] 0.2418
```

```r
p_Hearts = round(p_Hearts * 100, digits = 1)
paste("Probability of drawing a Heart =", p_Hearts,"%")
```

```
## [1] "Probability of drawing a Heart = 24.2 %"
```

Probability of drawing the Jack of Hearts

```r
p_JH <- ifelse(trial == "J H", 1,0)
p_JH <- sum(p_JH) / n
p_JH
```

```
## [1] 0.018
```

```r
p_JH = round(p_JH * 100, digits = 1)
paste("Probability of drawing the Jack of Hearts =", p_JH,"%")
```

```
## [1] "Probability of drawing the Jack of Hearts = 1.8 %"
```

Probability of drawing a Jack or Heart

```r
p_J_or_H <- p_Jack + p_Hearts - p_JH
p_J_or_H
```

```
## [1] 30
```

```r
paste("Probability of drawing a Jack or Heart =", p_J_or_H,"%")
```

```
## [1] "Probability of drawing a Jack or Heart = 30 %"
```

Probability of drawing 2 cards of the same suit

Monte Carlo Simulation

```r
same_suit <- function(card1, card2) {
    # substr(x, start, stop) Substrings of a Character Vector
    return(substr(card1, 3,3) == substr(card2, 3,3))
}
sscount <- 0
iterations <- 0
while(iterations < 10000) { # do 10,000 simulations
    sdeck <- shuffle(deck)
    twocards <- sample(sdeck, 2, replace=FALSE)
    if(same_suit(twocards[1], twocards[2])){  # TRUE or FALSE
        sscount <- sscount + 1 }
        iterations <- iterations + 1 }
p <- sscount/iterations #the estimated probability of 2 cards same rank
p <- round(p * 100, digits = 1)
paste("Probability of drawing 2 cards of the same suit =", p,"%")
```

```
## [1] "Probability of drawing 2 cards of the same suit = 23.5 %"
```

Probability of drawing 2 cards of the same rank.

```r
same_rank <- function(card1, card2) {
    # substr(x, start, stop) Substrings of a Character Vector
    return(substr(card1, 1,1) == substr(card2, 1,1))
}
srcount <- 0
iterations <- 0
while(iterations < 10000) { # do 10,000 simulations
    sdeck <- shuffle(deck)
    twocards <- sample(sdeck, 2, replace=FALSE)
    if(same_rank(twocards[1], twocards[2])){  # TRUE or FALSE
        srcount <- srcount + 1 }
        iterations <- iterations + 1 }
```

```r
p <- srcount/iterations #the estimated probability of 2 cards same suit
p <- round(p * 100, digits = 1)
paste("Probability of drawing 2 cards of the same rank =", p,"%")
```

```
## [1] "Probability of drawing 2 cards of the same rank = 5.8 %"
```