

Loops

coop711

2018 5 30

```
load("./S3.RData")
```

Expected Values

```
die <- 1:6
```

expand.grid

```
rolls <- expand.grid(die, die)
# rolls
expand.grid(1:2, 1:2, 1:2)
```

```
##   Var1 Var2 Var3
## 1    1    1    1
## 2    2    1    1
## 3    1    2    1
## 4    2    2    1
## 5    1    1    2
## 6    2    1    2
## 7    1    2    2
## 8    2    2    2
```

```
rolls$value <- rolls$Var1 + rolls$Var2
head(rolls, 3)
```

```
##   Var1 Var2 value
## 1    1    1     2
## 2    2    1     3
## 3    3    1     4
```

```
prob <- c("1" = 1/8, "2" = 1/8, "3" = 1/8, "4" = 1/8, "5" = 1/8, "6" = 3/8)
prob
```

```
##      1      2      3      4      5      6
## 0.125 0.125 0.125 0.125 0.125 0.375
```

```
rolls$Var1
```

```
## [1] 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5
## [36] 6
```

```
prob[rolls$Var1]
```

```
##      1      2      3      4      5      6
## 0.125 0.125 0.125 0.125 0.125 0.375 0.125 0.125 0.125 0.125 0.125 0.375
##      1      2      3      4      5      6
## 0.125 0.125 0.125 0.125 0.125 0.375 0.125 0.125 0.125 0.125 0.125 0.375
##      1      2      3      4      5      6
## 0.125 0.125 0.125 0.125 0.125 0.375 0.125 0.125 0.125 0.125 0.125 0.375
```

```
rolls$prob1 <- prob[rolls$Var1]
rolls$prob2 <- prob[rolls$Var2]
rolls
```

```
##      Var1 Var2 value prob1 prob2
## 1      1    1     2 0.125 0.125
## 2      2    1     3 0.125 0.125
## 3      3    1     4 0.125 0.125
## 4      4    1     5 0.125 0.125
## 5      5    1     6 0.125 0.125
## 6      6    1     7 0.375 0.125
## 7      1    2     3 0.125 0.125
## 8      2    2     4 0.125 0.125
## 9      3    2     5 0.125 0.125
## 10     4    2     6 0.125 0.125
## 11     5    2     7 0.125 0.125
## 12     6    2     8 0.375 0.125
## 13     1    3     4 0.125 0.125
## 14     2    3     5 0.125 0.125
## 15     3    3     6 0.125 0.125
## 16     4    3     7 0.125 0.125
## 17     5    3     8 0.125 0.125
## 18     6    3     9 0.375 0.125
## 19     1    4     5 0.125 0.125
## 20     2    4     6 0.125 0.125
## 21     3    4     7 0.125 0.125
## 22     4    4     8 0.125 0.125
## 23     5    4     9 0.125 0.125
## 24     6    4    10 0.375 0.125
## 25     1    5     6 0.125 0.125
## 26     2    5     7 0.125 0.125
## 27     3    5     8 0.125 0.125
## 28     4    5     9 0.125 0.125
## 29     5    5    10 0.125 0.125
## 30     6    5    11 0.375 0.125
## 31     1    6     7 0.125 0.375
## 32     2    6     8 0.125 0.375
## 33     3    6     9 0.125 0.375
## 34     4    6    10 0.125 0.375
## 35     5    6    11 0.125 0.375
## 36     6    6    12 0.375 0.375
```

```
rolls$prob <- rolls$prob1 * rolls$prob2
rolls
```

```
##      Var1 Var2 value prob1 prob2      prob
## 1      1      1      2 0.125 0.125 0.015625
## 2      2      1      3 0.125 0.125 0.015625
## 3      3      1      4 0.125 0.125 0.015625
## 4      4      1      5 0.125 0.125 0.015625
## 5      5      1      6 0.125 0.125 0.015625
## 6      6      1      7 0.375 0.125 0.046875
## 7      1      2      3 0.125 0.125 0.015625
## 8      2      2      4 0.125 0.125 0.015625
## 9      3      2      5 0.125 0.125 0.015625
## 10     4      2      6 0.125 0.125 0.015625
## 11     5      2      7 0.125 0.125 0.015625
## 12     6      2      8 0.375 0.125 0.046875
## 13     1      3      4 0.125 0.125 0.015625
## 14     2      3      5 0.125 0.125 0.015625
## 15     3      3      6 0.125 0.125 0.015625
## 16     4      3      7 0.125 0.125 0.015625
## 17     5      3      8 0.125 0.125 0.015625
## 18     6      3      9 0.375 0.125 0.046875
## 19     1      4      5 0.125 0.125 0.015625
## 20     2      4      6 0.125 0.125 0.015625
## 21     3      4      7 0.125 0.125 0.015625
## 22     4      4      8 0.125 0.125 0.015625
## 23     5      4      9 0.125 0.125 0.015625
## 24     6      4     10 0.375 0.125 0.046875
## 25     1      5      6 0.125 0.125 0.015625
## 26     2      5      7 0.125 0.125 0.015625
## 27     3      5      8 0.125 0.125 0.015625
## 28     4      5      9 0.125 0.125 0.015625
## 29     5      5     10 0.125 0.125 0.015625
## 30     6      5     11 0.375 0.125 0.046875
## 31     1      6      7 0.125 0.375 0.046875
## 32     2      6      8 0.125 0.375 0.046875
## 33     3      6      9 0.125 0.375 0.046875
## 34     4      6     10 0.125 0.375 0.046875
## 35     5      6     11 0.125 0.375 0.046875
## 36     6      6     12 0.375 0.375 0.140625
```

```
sum(rolls$value * rolls$prob)
```

```
## [1] 8.25
```

```
wheel <- c("DD", "7", "BBB", "BB", "B", "C", "0")
combos <- expand.grid(wheel, wheel, wheel,
                     stringsAsFactors = FALSE)

# combos
get_symbols <- function() {
  wheel <- c("DD", "7", "BBB", "BB", "B", "C", "0")
  sample(wheel,
         size = 3,
         replace = TRUE,
         prob = c(0.03, 0.03, 0.06, 0.1, 0.25, 0.01, 0.52))
}
prob <- c("DD" = 0.03,
         "7" = 0.03,
         "BBB" = 0.06,
         "BB" = 0.1,
         "B" = 0.25,
         "C" = 0.01,
         "0" = 0.52)
combos$prob1 <- prob[combos$Var1]
combos
```

Slot Machine

##	Var1	Var2	Var3	probl
## 1	DD	DD	DD	0.03
## 2	7	DD	DD	0.03
## 3	BBB	DD	DD	0.06
## 4	BB	DD	DD	0.10
## 5	B	DD	DD	0.25
## 6	C	DD	DD	0.01
## 7	0	DD	DD	0.52
## 8	DD	7	DD	0.03
## 9	7	7	DD	0.03
## 10	BBB	7	DD	0.06
## 11	BB	7	DD	0.10
## 12	B	7	DD	0.25
## 13	C	7	DD	0.01
## 14	0	7	DD	0.52
## 15	DD	BBB	DD	0.03
## 16	7	BBB	DD	0.03
## 17	BBB	BBB	DD	0.06
## 18	BB	BBB	DD	0.10
## 19	B	BBB	DD	0.25
## 20	C	BBB	DD	0.01
## 21	0	BBB	DD	0.52
## 22	DD	BB	DD	0.03
## 23	7	BB	DD	0.03
## 24	BBB	BB	DD	0.06
## 25	BB	BB	DD	0.10
## 26	B	BB	DD	0.25
## 27	C	BB	DD	0.01
## 28	0	BB	DD	0.52
## 29	DD	B	DD	0.03
## 30	7	B	DD	0.03
## 31	BBB	B	DD	0.06
## 32	BB	B	DD	0.10
## 33	B	B	DD	0.25
## 34	C	B	DD	0.01
## 35	0	B	DD	0.52
## 36	DD	C	DD	0.03
## 37	7	C	DD	0.03
## 38	BBB	C	DD	0.06
## 39	BB	C	DD	0.10
## 40	B	C	DD	0.25
## 41	C	C	DD	0.01
## 42	0	C	DD	0.52
## 43	DD	0	DD	0.03
## 44	7	0	DD	0.03
## 45	BBB	0	DD	0.06
## 46	BB	0	DD	0.10
## 47	B	0	DD	0.25
## 48	C	0	DD	0.01
## 49	0	0	DD	0.52
## 50	DD	DD	7	0.03
## 51	7	DD	7	0.03
## 52	BBB	DD	7	0.06
## 53	BB	DD	7	0.10
## 54	B	DD	7	0.25
## 55	C	DD	7	0.01
## 56	0	DD	7	0.52

## 57	DD	7	7	0.03
## 58	7	7	7	0.03
## 59	BBB	7	7	0.06
## 60	BB	7	7	0.10
## 61	B	7	7	0.25
## 62	C	7	7	0.01
## 63	0	7	7	0.52
## 64	DD	BBB	7	0.03
## 65	7	BBB	7	0.03
## 66	BBB	BBB	7	0.06
## 67	BB	BBB	7	0.10
## 68	B	BBB	7	0.25
## 69	C	BBB	7	0.01
## 70	0	BBB	7	0.52
## 71	DD	BB	7	0.03
## 72	7	BB	7	0.03
## 73	BBB	BB	7	0.06
## 74	BB	BB	7	0.10
## 75	B	BB	7	0.25
## 76	C	BB	7	0.01
## 77	0	BB	7	0.52
## 78	DD	B	7	0.03
## 79	7	B	7	0.03
## 80	BBB	B	7	0.06
## 81	BB	B	7	0.10
## 82	B	B	7	0.25
## 83	C	B	7	0.01
## 84	0	B	7	0.52
## 85	DD	C	7	0.03
## 86	7	C	7	0.03
## 87	BBB	C	7	0.06
## 88	BB	C	7	0.10
## 89	B	C	7	0.25
## 90	C	C	7	0.01
## 91	0	C	7	0.52
## 92	DD	0	7	0.03
## 93	7	0	7	0.03
## 94	BBB	0	7	0.06
## 95	BB	0	7	0.10
## 96	B	0	7	0.25
## 97	C	0	7	0.01
## 98	0	0	7	0.52
## 99	DD	DD	BBB	0.03
## 100	7	DD	BBB	0.03
## 101	BBB	DD	BBB	0.06
## 102	BB	DD	BBB	0.10
## 103	B	DD	BBB	0.25
## 104	C	DD	BBB	0.01
## 105	0	DD	BBB	0.52
## 106	DD	7	BBB	0.03
## 107	7	7	BBB	0.03
## 108	BBB	7	BBB	0.06
## 109	BB	7	BBB	0.10
## 110	B	7	BBB	0.25
## 111	C	7	BBB	0.01
## 112	0	7	BBB	0.52
## 113	DD	BBB	BBB	0.03
## 114	7	BBB	BBB	0.03

##	115	BBB	BBB	BBB	0.06
##	116	BB	BBB	BBB	0.10
##	117	B	BBB	BBB	0.25
##	118	C	BBB	BBB	0.01
##	119	0	BBB	BBB	0.52
##	120	DD	BB	BBB	0.03
##	121	7	BB	BBB	0.03
##	122	BBB	BB	BBB	0.06
##	123	BB	BB	BBB	0.10
##	124	B	BB	BBB	0.25
##	125	C	BB	BBB	0.01
##	126	0	BB	BBB	0.52
##	127	DD	B	BBB	0.03
##	128	7	B	BBB	0.03
##	129	BBB	B	BBB	0.06
##	130	BB	B	BBB	0.10
##	131	B	B	BBB	0.25
##	132	C	B	BBB	0.01
##	133	0	B	BBB	0.52
##	134	DD	C	BBB	0.03
##	135	7	C	BBB	0.03
##	136	BBB	C	BBB	0.06
##	137	BB	C	BBB	0.10
##	138	B	C	BBB	0.25
##	139	C	C	BBB	0.01
##	140	0	C	BBB	0.52
##	141	DD	0	BBB	0.03
##	142	7	0	BBB	0.03
##	143	BBB	0	BBB	0.06
##	144	BB	0	BBB	0.10
##	145	B	0	BBB	0.25
##	146	C	0	BBB	0.01
##	147	0	0	BBB	0.52
##	148	DD	DD	BB	0.03
##	149	7	DD	BB	0.03
##	150	BBB	DD	BB	0.06
##	151	BB	DD	BB	0.10
##	152	B	DD	BB	0.25
##	153	C	DD	BB	0.01
##	154	0	DD	BB	0.52
##	155	DD	7	BB	0.03
##	156	7	7	BB	0.03
##	157	BBB	7	BB	0.06
##	158	BB	7	BB	0.10
##	159	B	7	BB	0.25
##	160	C	7	BB	0.01
##	161	0	7	BB	0.52
##	162	DD	BBB	BB	0.03
##	163	7	BBB	BB	0.03
##	164	BBB	BBB	BB	0.06
##	165	BB	BBB	BB	0.10
##	166	B	BBB	BB	0.25
##	167	C	BBB	BB	0.01
##	168	0	BBB	BB	0.52
##	169	DD	BB	BB	0.03
##	170	7	BB	BB	0.03
##	171	BBB	BB	BB	0.06
##	172	BB	BB	BB	0.10

##	173	B	BB	BB	0.25
##	174	C	BB	BB	0.01
##	175	0	BB	BB	0.52
##	176	DD	B	BB	0.03
##	177	7	B	BB	0.03
##	178	BBB	B	BB	0.06
##	179	BB	B	BB	0.10
##	180	B	B	BB	0.25
##	181	C	B	BB	0.01
##	182	0	B	BB	0.52
##	183	DD	C	BB	0.03
##	184	7	C	BB	0.03
##	185	BBB	C	BB	0.06
##	186	BB	C	BB	0.10
##	187	B	C	BB	0.25
##	188	C	C	BB	0.01
##	189	0	C	BB	0.52
##	190	DD	0	BB	0.03
##	191	7	0	BB	0.03
##	192	BBB	0	BB	0.06
##	193	BB	0	BB	0.10
##	194	B	0	BB	0.25
##	195	C	0	BB	0.01
##	196	0	0	BB	0.52
##	197	DD	DD	B	0.03
##	198	7	DD	B	0.03
##	199	BBB	DD	B	0.06
##	200	BB	DD	B	0.10
##	201	B	DD	B	0.25
##	202	C	DD	B	0.01
##	203	0	DD	B	0.52
##	204	DD	7	B	0.03
##	205	7	7	B	0.03
##	206	BBB	7	B	0.06
##	207	BB	7	B	0.10
##	208	B	7	B	0.25
##	209	C	7	B	0.01
##	210	0	7	B	0.52
##	211	DD	BBB	B	0.03
##	212	7	BBB	B	0.03
##	213	BBB	BBB	B	0.06
##	214	BB	BBB	B	0.10
##	215	B	BBB	B	0.25
##	216	C	BBB	B	0.01
##	217	0	BBB	B	0.52
##	218	DD	BB	B	0.03
##	219	7	BB	B	0.03
##	220	BBB	BB	B	0.06
##	221	BB	BB	B	0.10
##	222	B	BB	B	0.25
##	223	C	BB	B	0.01
##	224	0	BB	B	0.52
##	225	DD	B	B	0.03
##	226	7	B	B	0.03
##	227	BBB	B	B	0.06
##	228	BB	B	B	0.10
##	229	B	B	B	0.25
##	230	C	B	B	0.01

##	231	0	B	B	0.52
##	232	DD	C	B	0.03
##	233	7	C	B	0.03
##	234	BBB	C	B	0.06
##	235	BB	C	B	0.10
##	236	B	C	B	0.25
##	237	C	C	B	0.01
##	238	0	C	B	0.52
##	239	DD	0	B	0.03
##	240	7	0	B	0.03
##	241	BBB	0	B	0.06
##	242	BB	0	B	0.10
##	243	B	0	B	0.25
##	244	C	0	B	0.01
##	245	0	0	B	0.52
##	246	DD	DD	C	0.03
##	247	7	DD	C	0.03
##	248	BBB	DD	C	0.06
##	249	BB	DD	C	0.10
##	250	B	DD	C	0.25
##	251	C	DD	C	0.01
##	252	0	DD	C	0.52
##	253	DD	7	C	0.03
##	254	7	7	C	0.03
##	255	BBB	7	C	0.06
##	256	BB	7	C	0.10
##	257	B	7	C	0.25
##	258	C	7	C	0.01
##	259	0	7	C	0.52
##	260	DD	BBB	C	0.03
##	261	7	BBB	C	0.03
##	262	BBB	BBB	C	0.06
##	263	BB	BBB	C	0.10
##	264	B	BBB	C	0.25
##	265	C	BBB	C	0.01
##	266	0	BBB	C	0.52
##	267	DD	BB	C	0.03
##	268	7	BB	C	0.03
##	269	BBB	BB	C	0.06
##	270	BB	BB	C	0.10
##	271	B	BB	C	0.25
##	272	C	BB	C	0.01
##	273	0	BB	C	0.52
##	274	DD	B	C	0.03
##	275	7	B	C	0.03
##	276	BBB	B	C	0.06
##	277	BB	B	C	0.10
##	278	B	B	C	0.25
##	279	C	B	C	0.01
##	280	0	B	C	0.52
##	281	DD	C	C	0.03
##	282	7	C	C	0.03
##	283	BBB	C	C	0.06
##	284	BB	C	C	0.10
##	285	B	C	C	0.25
##	286	C	C	C	0.01
##	287	0	C	C	0.52
##	288	DD	0	C	0.03

##	289	7	0	C	0.03
##	290	BBB	0	C	0.06
##	291	BB	0	C	0.10
##	292	B	0	C	0.25
##	293	C	0	C	0.01
##	294	0	0	C	0.52
##	295	DD	DD	0	0.03
##	296	7	DD	0	0.03
##	297	BBB	DD	0	0.06
##	298	BB	DD	0	0.10
##	299	B	DD	0	0.25
##	300	C	DD	0	0.01
##	301	0	DD	0	0.52
##	302	DD	7	0	0.03
##	303	7	7	0	0.03
##	304	BBB	7	0	0.06
##	305	BB	7	0	0.10
##	306	B	7	0	0.25
##	307	C	7	0	0.01
##	308	0	7	0	0.52
##	309	DD	BBB	0	0.03
##	310	7	BBB	0	0.03
##	311	BBB	BBB	0	0.06
##	312	BB	BBB	0	0.10
##	313	B	BBB	0	0.25
##	314	C	BBB	0	0.01
##	315	0	BBB	0	0.52
##	316	DD	BB	0	0.03
##	317	7	BB	0	0.03
##	318	BBB	BB	0	0.06
##	319	BB	BB	0	0.10
##	320	B	BB	0	0.25
##	321	C	BB	0	0.01
##	322	0	BB	0	0.52
##	323	DD	B	0	0.03
##	324	7	B	0	0.03
##	325	BBB	B	0	0.06
##	326	BB	B	0	0.10
##	327	B	B	0	0.25
##	328	C	B	0	0.01
##	329	0	B	0	0.52
##	330	DD	C	0	0.03
##	331	7	C	0	0.03
##	332	BBB	C	0	0.06
##	333	BB	C	0	0.10
##	334	B	C	0	0.25
##	335	C	C	0	0.01
##	336	0	C	0	0.52
##	337	DD	0	0	0.03
##	338	7	0	0	0.03
##	339	BBB	0	0	0.06
##	340	BB	0	0	0.10
##	341	B	0	0	0.25
##	342	C	0	0	0.01
##	343	0	0	0	0.52

```
combos$prob2 <- prob[combos$Var2]
combos$prob3 <- prob[combos$Var3]
head(combos, 3)
```

```
##   Var1 Var2 Var3 prob1 prob2 prob3
## 1   DD   DD   DD  0.03  0.03  0.03
## 2    7   DD   DD  0.03  0.03  0.03
## 3  BBB   DD   DD  0.06  0.03  0.03
```

```
combos$prob <- combos$prob1 * combos$prob2 * combos$prob3
head(combos, 3)
```

```
##   Var1 Var2 Var3 prob1 prob2 prob3   prob
## 1   DD   DD   DD  0.03  0.03  0.03 2.7e-05
## 2    7   DD   DD  0.03  0.03  0.03 2.7e-05
## 3  BBB   DD   DD  0.06  0.03  0.03 5.4e-05
```

```
sum(combos$prob)
```

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

```
symbols <- c(combos[1, 1], combos[1, 2], combos[1, 3])
score(symbols)
```

```
## [1] 800
```

for Loops

```
for (value in c("My", "First", "for", "loop")) {
  print("one run")
}
```

```
## [1] "one run"
## [1] "one run"
## [1] "one run"
## [1] "one run"
```

```
value
```

```
## [1] "loop"
```

```
for (value in c("My", "First", "for", "loop")) {
  print(value)
}
```

```
## [1] "My"
## [1] "First"
## [1] "for"
## [1] "loop"
```

```
for (word in c("My", "First", "for", "loop")) {
  print(word)
}
```

```
## [1] "My"
## [1] "First"
## [1] "for"
## [1] "loop"
```

```
for (string in c("My", "First", "for", "loop")) {
  print(string)
}
```

```
## [1] "My"
## [1] "First"
## [1] "for"
## [1] "loop"
```

```
for (i in c("My", "First", "for", "loop")) {
  print(i)
}
```

```
## [1] "My"
## [1] "First"
## [1] "for"
## [1] "loop"
```

```
for (value in c("My", "First", "for", "loop")) {
  value
}
chars <- vector(length = 4)
chars
```

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

```
words <- c("My", "First", "for", "loop")
for (i in 1:4) {
  chars[i] <- words[i]
}
chars
```

```
## [1] "My"      "First"   "for"     "loop"
```

```
combos$prize <- NA
head(combos, 3)
```

```
##   Var1 Var2 Var3 prob1 prob2 prob3   prob prize
## 1   DD   DD   DD  0.03  0.03  0.03 2.7e-05    NA
## 2    7   DD   DD  0.03  0.03  0.03 2.7e-05    NA
## 3  BBB   DD   DD  0.06  0.03  0.03 5.4e-05    NA
```

```
for (i in 1:nrow(combos)) {
  symbols <- c(combos[i, 1], combos[i, 2], combos[i, 3])
  combos$prize[i] <- score(symbols)
}
head(combos, 3)
```

```
##   Var1 Var2 Var3 prob1 prob2 prob3   prob prize
## 1   DD   DD   DD  0.03  0.03  0.03 2.7e-05   800
## 2    7   DD   DD  0.03  0.03  0.03 2.7e-05    0
## 3  BBB   DD   DD  0.06  0.03  0.03 5.4e-05    0
```

```
sum(combos$prize * combos$prob)
```

```
## [1] 0.538014
```

```
score <- function(symbols) {
  diamonds <- sum(symbols == "DD")
  cherries <- sum(symbols == "C")

  # identify case
  # since diamonds are wild, only nondiamonds
  # matter for three of a kind and all bars
  slots <- symbols[symbols != "DD"]
  same <- length(unique(slots)) == 1
  bars <- slots %in% c("B", "BB", "BBB")

  # assign prize
  if (diamonds == 3) {
    prize <- 100
  } else if (same) {
    payouts <- c("7" = 80,
                 "BBB" = 40,
                 "BB" = 25,
                 "B" = 10,
                 "C" = 10,
                 "0" = 0)
    prize <- unname(payouts[slots[1]])
  } else if (all(bars)) {
    prize <- 5
  } else if (cherries > 0) {
    # diamonds count as cherries
    # so long as there is one real cherry
    prize <- c(0, 2, 5)[cherries + diamonds + 1]
  } else {
    prize <- 0
  }

  # double for each diamond
  prize * 2^diamonds
}
score(c("BB", "DD", "B"))
```

```
## [1] 10
```

```
for (i in 1:nrow(combos)) {
  symbols <- c(combos[i, 1], combos[i, 2], combos[i, 3])
  combos$prize[i] <- score(symbols)
}
head(combos, 3)
```

```
##   Var1 Var2 Var3 prob1 prob2 prob3   prob prize
## 1   DD   DD   DD  0.03  0.03  0.03 2.7e-05   800
## 2    7   DD   DD  0.03  0.03  0.03 2.7e-05   320
## 3  BBB   DD   DD  0.06  0.03  0.03 5.4e-05   160
```

```
sum(combos$prize * combos$prob)
```

```
## [1] 0.934356
```

```
save.image(file = "./Loops.RData")
```