

POL211 TA Session 2

Gento Kato

October 11, 2018

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))  
rm(list = ls()) # Remove all objects from workspace.
```

Submitting R Code

If you have problems including R codes within HW documents, submit the **raw R code file (with .R extension)** directly to my E-mail address.

- **NO** PDFs, Microsoft Words, Text files, etc. . .
- Include **your name** in the file name.
- **Only** include codes relevant to the homework.
- Make sure to run codes by yourself. Include results in the main texts (codes are supplemental to your results).

HW 1 Simulations

Q4b

```
# Probability Space  
samplespace <- c(1,-1) # Sample Space  
probmeasure <- c(0.5,0.5) # Prbability Measure  
  
n <- 100 # N of coin flips in each simulation  
m <- 1000 # N of simulation  
  
# Simulation  
set.seed(458789) # make the result replicable  
simres <- replicate(m,replicate(n, sample(samplespace, 1, prob = probmeasure)))  
  
over10 <- sum(abs(colSums(simres))>=10) # N of One Player winning > $10  
over10
```

```
## [1] 382
```

```
pval <- over10/m # Pr that One Player wins $10 or more by chance  
pval
```

```
## [1] 0.382
```

Q7

Using For Loop

VERY basic of the simulation. Intuitive, but **SLOW**.

```

# Identifier for which month the given day falls into
month <- c(rep(1, 31), rep(2, 29), rep(3, 31), rep(4, 30), rep(5, 31), rep(6, 30),
          rep(7, 31), rep(8, 31), rep(9, 30), rep(10, 31), rep(11, 30), rep(12, 31))
table(month)

## month
##  1  2  3  4  5  6  7  8  9 10 11 12
## 31 29 31 30 31 30 31 31 30 31 30 31

# * Shorter way
month <- sort(rep(1:12,31)[-c(2,4,6,9,11,14)])
table(month)

## month
##  1  2  3  4  5  6  7  8  9 10 11 12
## 31 29 31 30 31 30 31 31 30 31 30 31

m <- 1000      # number of simulation runs
success <- rep(NA, m) # create empty vector for success count

set.seed(78990009) # make the simulation replicable

#start of for loop
for(j in 1:m){
  dayrank <- sample(x = 366)
  #randomly permute the days of the year.
  #The new order corresponds to the hypothetical lottery ranking.
  gstar <- rep(NA, 12) # empty vector for monthly av. ranking
  for (i in 1:12){
    gstar[i] <- mean(dayrank[month==i])
    #Calculate the average monthly ranking
  }
  ystar <- sum(abs(gstar - 183.5))
  # sum of abs diff bw gstar and E(monthly av)=183.5
  success[j] <- ystar >= 272.5
  # count as success if ystar >= 272.5
}
#end of for loop

# Calculate P-value
pval <- sum(success)/m
pval

## [1] 0.013

```

What is discussed in Office Hours Yesterday...

```

# Identifier of Month
month <- rep(1:12, c(31,29,31,30,31,30,31,31,30,31,30,31))
month

##      [1]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##     [24]  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2
##     [47]  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  3  3
##     [70]  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  4

```

```
## [93] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [116] 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [139] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6
## [162] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7
## [185] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [208] 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [231] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9
## [254] 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 10 10
## [277] 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
## [300] 10 10 10 10 10 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
## [323] 11 11 11 11 11 11 11 11 11 11 11 11 11 12 12 12 12 12 12 12 12
## [346] 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
```

```
# Set N of Simulation Runs
```

```
m <- 1000
```

```
# Draw Ranking of Days m times
```

```
set.seed(569)
```

```
dayrank <- replicate(m, sample(c(1:366)))
```

```
dim(dayrank)
```

```
## [1] 366 1000
```

```
# Monthly Average Rankings
```

```
gstar <- apply(dayrank, 2, function(k) tapply(k, month, mean))
```

```
dim(gstar)
```

```
## [1] 12 1000
```

```
# Sum of Abs. Deviation from 183.5
```

```
ystar <- colSums(abs(gstar-183.5))
```

```
length(ystar)
```

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

```
# Success Rate (P-value)
```

```
pval2 <- sum(ystar >= 272.5)/m
```

```
pval2
```

```
## [1] 0.019
```

Even Shorter (in one line)...

```
## First version
```

```
m <- 1000 # Set number of simulation runs
```

```
set.seed(78990009) # make the simulation replicable (use the same seed as 1st one)
```

```
pval3 <- mean(replicate(m, sum(abs(tapply(sample(366),
                                         sort(rep(1:12, 31)[-c(2, 4, 6, 9, 11, 14)]),
                                         mean)-183.5)) >= 272.5))
```

```
pval3
```

```
## [1] 0.013
```

```
pval3 == pval # The same p-value as the previous simulation
```

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

```
## Second version
m <- 1000 # Set number of simulation runs
set.seed(569) # make the simulation replicable
pval4 <- sum(colSums(abs(apply(replicate(m, sample(c(1:366))), 2,
                             function(k) tapply (k, month, mean))-183.5)) >= 272.5)/m
pval4

## [1] 0.019
pval4 == pval2 # The same p-value as the previous simulation

## [1] TRUE
```

The Law of large numbers

Conduct Meta simulations with m=10, m=100, m=1000, each for 100 times.

```
n <- 100
m10simu <- replicate(n, mean(replicate(10, sum(abs(tapply(sample(366),
                  sort(rep(1:12, 31)[-c(2, 4, 6, 9, 11, 14)]),
                  mean)-183.5))>=272.5)))
m100simu <- replicate(n, mean(replicate(100, sum(abs(tapply(sample(366),
                  sort(rep(1:12, 31)[-c(2, 4, 6, 9, 11, 14)]),
                  mean)-183.5))>=272.5)))
m1000simu <- replicate(n, mean(replicate(1000, sum(abs(tapply(sample(366),
                  sort(rep(1:12, 31)[-c(2, 4, 6, 9, 11, 14)]),
                  mean)-183.5))>=272.5)))

d <- data.frame(m10 = m10simu,
                m100 = m100simu,
                m1000 = m1000simu)

library(ggplot2)
p <- ggplot(data=d) +
  geom_density(aes(m10, fill="m=10"), alpha=0.5) +
  geom_density(aes(m100, fill="m=100"), alpha=0.5) +
  geom_density(aes(m1000, fill="m=1000"), alpha=0.5) +
  theme_bw() + xlab("Simulated P-values") +
  ggtitle("The variance of p-val distribution is decrease in m")
p
```

The variance of p-val distribution is decrease in m

