Stat405

Simulation

Hadley Wickham

1. Simulation
2. Mathematical approach
3. Homework
4. Feedback

Symposium?

Bootstrap estimate

Casino claims that slot machines have prize payout of 92%. Is this claim true?

Payoff for 345 attempts is 0.67.

What does that tell us about true distribution? Can now generate payoffs distribution for that many pulls. Is 0.92 likely or unlikely?

```
calculate_prize <- function(windows) {
  payoffs <- c("DD" = 800, "7" = 80, "BBB" = 40,
    "BB" = 25, "B" = 10, "C" = 10, "0" = 0)

  same <- length(unique(windows)) == 1
  allbars <- all(windows %in% c("B", "BB", "BBB"))

  if (same) {
    prize <- payoffs[windows[1]]
  } else if (allbars) {
    prize <- 5
  } else {
    cherries <- sum(windows == "C")
    diamonds <- sum(windows == "DD")

    prize <- c(0, 2, 5)[cherries + 1] *
      c(1, 2, 4)[diamonds + 1]
  }
  prize
}
```

Your turn

Now want to generate a new draw from the same distribution (a bootstrap sample).

Write a function that returns the prize from a randomly generated new draw. Call the function `random_prize`

Hint: `sample(slot$w1, 1)` will draw a single sample randomly from the original data

```r
slots <- read.csv("slots.csv", stringsAsFactors = F)

w1 <- sample(slots$w1, 1)
w2 <- sample(slots$w2, 1)
w3 <- sample(slots$w3, 1)

calculate_prize(c(w1, w2, w3))
```

```r
random_prize <- function() {
  w1 <- sample(slots$w1, 1)
  w2 <- sample(slots$w2, 1)
  w3 <- sample(slots$w3, 1)

  calculate_prize(c(w1, w2, w3))
}

# What is the implicit assumption here?
# How could we test that assumption?
```

Your turn

Write a function to do this n times

Use a for loop: create an empty vector, and then fill with values

Draw a histogram of the results

```
n <- 100
prizes <- rep(NA, n)

for(i in seq_along(prizes)) {
  prizes[i] <- random_prize()
}
```

```
random_prizes <- function(n) {
  prizes <- rep(NA, n)

  for(i in seq_along(prizes)) {
    prizes[i] <- random_prize()
  }
  prizes
}

library(ggplot2)
qplot(random_prizes(100))
```

Your turn

Create a function that returns the payoff (mean prize) for n draws (`payoff`)

Then create a function that draws from the distribution of mean payoffs m times (`payoff_distr`)

Explore the distribution. What happens as m increases? What about n?

```
payoff <- function(n) {
  mean(random_prizes(n))
}

m <- 100
n <- 10

payoffs <- rep(NA, m)
  for(i in seq_along(payoffs)) {
  payoffs[i] <- payoff(n)
}

qplot(payoffs, binwidth = 0.1)
```

```r
payoff_distr <- function(m, n) {
  payoffs <- rep(NA, m)
  for(i in seq_along(payoffs)) {
    payoffs[i] <- payoff(n)
  }
  payoffs
}

dist <- payoff_distr(100, 1000)
qplot(dist, binwidth = 0.05)
```

```
# Speeding things up a bit
system.time(payoff_distr(200, 100))

payoff <- function(n) {
  w1 <- sample(slots$w1, n, rep = T)
  w2 <- sample(slots$w2, n, rep = T)
  w3 <- sample(slots$w2, n, rep = T)

  prizes <- rep(NA, n)
  for(i in seq_along(prizes)) {
    prizes[i] <- calculate_prize(c(w1[i], w2[i], w3[i]))
  }
  mean(prizes)
}
system.time(payoff_distr(200, 100))
```

```
# Me - to save time
payoffs <- payoff_distr(10000, 345)
save(payoffs, file = "payoffs.rdata")

# You
load("payoffs.rdata")
str(payoffs)
```

```
qplot(payoffs, binwidth = 0.02)

mean(payoffs)
sd(payoffs)
mean(payoffs < mean(slots$prize))
quantile(payoffs, c(0.025, 0.975))
mean(payoffs) + c(-2, 2) * sd(payoffs)

t.test(slots$prize, mu = 0.92)
sd(slots$prize) / sqrt(nrow(slots))
```

Mathematical approach

Why are we doing this simulation? Could work out the expected value and variance mathematically. So let's do it!

Simplifying assumption: slots are iid.

```
# Calculate empirical distribution
dist <- table(c(slots$w1, slots$w2, slots$w3))
dist <- dist / sum(dist)

distdf <- as.data.frame(dist)
names(distdf) <- c("slot", "probability")
distdf$slot <- as.character(distdf$slot)
```

```
poss <- with(distdf, expand.grid(
  w1 = slot, w2 = slot, w3 = slot,
  stringsAsFactors = FALSE
))

poss$prize <- NA
for(i in seq_len(nrow(poss))) {
  window <- as.character(poss[i, 1:3])
  poss$prize[i] <- calculate_prize(window)
}
```

Your turn

How can you calculate the probability of each combination?

(**Hint**: think about subsetting. **More hint**: think about the table and character subsetting. **Final hint**: you can do this in one line of code)

Then work out the expected value and variance.

```
poss$prob <- with(poss,
  dist[w1] * dist[w2] * dist[w3])

(poss_mean <- with(poss, sum(prob * prize)))

# How do we determine the variance of this
# estimator?
```