Homework 1 solutions

Ciaran Evans

Practice with for loops

Question 1

```r
x <- seq(from=0, to=1, by=0.1)
cube_root_x <- rep(0, length(x))
for(i in 1:length(x)){
  cube_root_x[i] <- x[i]^(1/3)
}
cube_root_x
```

```
 [1] 0.0000000 0.4641589 0.5848035 0.6694330 0.7368063 0.7937005 0.8434327
 [8] 0.8879040 0.9283178 0.9654894 1.0000000
```

Note: the exponent needs to be in parentheses! E.g. `5^1/3` is not the same as `5^(1/3)`

Question 2

```r
x <- seq(from=0, to=2, by=0.05)
cube_root_x <- rep(0, length(x))
for(i in 1:length(x)){
  cube_root_x[i] <- x[i]^(1/3)
}
cube_root_x
```

```
 [1] 0.0000000 0.3684031 0.4641589 0.5313293 0.5848035 0.6299605 0.6694330
 [8] 0.7047299 0.7368063 0.7663094 0.7937005 0.8193213 0.8434327 0.8662391
[15] 0.8879040 0.9085603 0.9283178 0.9472682 0.9654894 0.9830476 1.0000000
[22] 1.0163964 1.0322801 1.0476896 1.0626586 1.0772173 1.0913929 1.1052094
[29] 1.1186889 1.1318512 1.1447142 1.1572945 1.1696071 1.1816658 1.1934832
[36] 1.2050711 1.2164404 1.2276010 1.2385623 1.2493330 1.2599210
```

Question 3

```
x <- seq(from=0, to=1, by=0.1)
cube_root_x <- x^(1/3)
cube_root_x
```

```
 [1] 0.0000000 0.4641589 0.5848035 0.6694330 0.7368063 0.7937005 0.8434327
 [8] 0.8879040 0.9283178 0.9654894 1.0000000
```

Questions 4 and 5

There are two different ways students could approach the problem here (the wording of the problem is slightly ambiguous). *I am fine with either approach.*

The simpler approach is to have the robots pull simultaneously, in which case the game is fair (each robot wins 50% of the time):

```
nsim <- 1000
results <- rep(NA, nsim)

for(i in 1:nsim){
  marker <- 0
  while(abs(marker) < 0.5){
    robotA <- runif(1, 0, 0.5)
    robotB <- runif(1, 0, 0.5)
    marker <- marker + robotA - robotB
  }
  results[i] <- marker >= 0.5
}

# fraction of the time that robot A wins
mean(results)
```

```
[1] 0.491
```

The slightly more complicated approach is to have the robots take *turns* pulling, and update the marker each time. Now there is a distinct advantage to robot A for pulling first:

```
nsim <- 1000
results <- rep(NA, nsim)
```

```r
for(i in 1:nsim){
  marker <- 0
  current_robot <- "A"
  while(abs(marker) < 0.5){
    pull <- runif(1, 0, 0.5)
    marker <- ifelse(current_robot == "A", marker + pull, marker - pull)
    current_robot <- ifelse(current_robot == "A", "B", "A")
  }
  results[i] <- marker >= 0.5
}

# fraction of the time that robot A wins
mean(results)
```

```
[1] 0.64
```

Question 9

- **Aims:** Assess the importance of the constant variance assumption in simple linear regression models
- **Data generation:** Data $(X_1, Y_1), ..., (X_n, Y_n)$ will be generated according to the following model:
  - $X_i \sim Uniform(0, 1)$
  - $\varepsilon_i \sim N(0, \sigma_i^2)$
  - $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$
  - To assess the importance of the constant variance assumption, we will consider several different standard deviations for ε_i: $\sigma_i = 1$ (constant variance), $\sigma_i = X_i$, and $\sigma_i = X_i^2$ (the latter two incorporating a relationship between σ_i and the explanatory variable X_i)
- **Estimands:** We will estimate β_1
- **Methods:** We will fit a linear regression model with the `lm` function in R, and calculate a 95% confidence interval for β_1
- **Performance measures:** observed coverage of the confidence intervals in repeated simulations

Question 10

```r
n <- 100
beta0 <- 0.5
beta1 <- 1
```

```r
x <- runif(n, min=0, max=1)
noise <- rnorm(n, mean=0, sd=x)
y <- beta0 + beta1*x + noise

plot(x,y)
```