Stats 380

Sooyong Choi

07/08/2020

Question 1a

```
seq(5.5, 32.5, by = 3)
```

```
##  [1]  5.5  8.5 11.5 14.5 17.5 20.5 23.5 26.5 29.5 32.5
```

Question 1b

```
cumsum(2:8)
```

```
## [1]  2  5  9 14 20 27 35
```

Question 1c

```
rev(cumprod(1:5) + 0.3)
```

```
## [1] 120.3  24.3   6.3   2.3   1.3
```

Question 1d

```
rep(0:4,each=3)[2:13]
```

```
##  [1] 0 0 1 1 1 2 2 2 3 3 3 4
```

Question 1e

```
seq(2.3, 3.8, by = 0.3)
```

```
## [1] 2.3 2.6 2.9 3.2 3.5 3.8
```

Question 1f

```
rep(seq(0.1, 0.6, by = 0.1), rep(3:1,2))
```

```
##  [1] 0.1 0.1 0.1 0.2 0.2 0.3 0.4 0.4 0.4 0.5 0.5 0.6
```

Question 1g

```
a = rep(0,14)
a[cumsum(2:5)] = 1:4
```

Question 1h

```
paste("x^", rep(3:1, 1:3), sep = "")
```

```
## [1] "x^3" "x^2" "x^2" "x^1" "x^1" "x^1"
```

Question 2a

```
sum1 = function(x) sum(exp(x)/((seq(along = x))^x))
```

```
> sum1(1)
[1] 2.718282
> sum1(2)
[1] 7.389056
> sum1(30)
[1] 1.068647e+13
```

Question 2b

```
sum2 = function(x) sum(exp(seq(along = x))/factorial(seq(along = x)))
```

```
> sum2(numeric(0))
[1] 0
> sum2(c(2,3,5))
[1] 9.760399
```

Question 3a

```
bmi_function = function(weight, height)
  round((weight/height^2), 3)
```

```
> bmi_function(80, 1.8)
[1] 24.691
```

Question 3b

```
weight.category = function(bmi) {
  if (bmi < 18.5) {
    cat("This corresponds to category: underweight")
  }

  else if (18.5 <= bmi & bmi < 25) {
    cat("This corresponds to category: normal weight")
  }

  else if (25 <= bmi & bmi < 33) {
    cat("This corresponds to category: overweight")
  }

  else
    cat("This corresponds to category: obesity")
}
```

```
> weight.category(15)
This corresponds to category: underweight
> weight.category(20)
This corresponds to category: normal weight
> weight.category(26)
This corresponds to category: overweight
> weight.category(36)
This corresponds to category: obesity
```

Question 4a

```
rwalk = function(steps) {
  if (steps > 0) {
    # to account for origin 0
    total_distance = numeric(steps+1)
    # generates random number between 1 and -1 with biased probability
    random = sample(c(-1,1), steps, prob = c(0.45, 0.55), replace = TRUE)
    x_n = cumsum(random)
    # to include origin 0 at the start
    total_distance[-1] = x_n
    return(total_distance)
  }

  else if (steps == 0) {
  return(0)
 }
}
```

```
> rwalk(3)
random numbers are  -1 -1 1
[1]  0 -1 -2 -1
```

```
> rwalk(0)
[1] 0
```

Question 4b

```
rwalk2 = function(steps) {
  if (steps > 0) {
    # to account for origin 0
    total_distance = numeric(steps+1)
    # generates random number between 1 and -1 with biased probability
    random = sample(c(-1,1), steps, prob = c(0.45, 0.55), replace = TRUE)
    cat("random numbers are ", random, "\n")
    x_n = cumsum(random)
    total_distance[2:length(total_distance)] = x_n
    cat("distances are ", total_distance, "\n")
    rhs = sum(ifelse(x_n > 0 | total_distance[-steps] > 0, 1, 0))
    return(rhs)
```

```
    }

  else if (steps == 0) {
    return(0)
  }
}
```

```
> rwalk2(3)
random numbers are  1 1 -1
distances are  0 1 2 1
[1] 3
```

```
> rwalk2(0)
[1] 0
```