HW1

Shijie Gao, USC ID:6037-6293-25

2023-01-11

```
#1 X~Bino(n = 12, p = 0,25)
#(a) P(X <= 2)
Prob = pbinom(2, size = 12, prob = 0.25)
Prob
```

```
## [1] 0.390675
```

```
#(b) P(0.7 < X <= 3.1)
Prob = pbinom(3.1, size = 12, prob = 0.25) - pbinom(0.7, size = 12, prob = 0.25)
Prob
```

```
## [1] 0.6171023
```

```
#2 X~N(1.9, 0.36)
#(a) P(X > 2)
Prob = 1- pnorm(2, mean = 1.9, sd = 0.6)
Prob
```

```
## [1] 0.4338162
```

```
#(b) P(0.7 < X < 3.1)
Prob = pnorm(3.1, mean = 1.9, sd = 0.6) - pnorm(0.7, mean = 1.9, sd = 0.6)
Prob
```

```
## [1] 0.9544997
```

```
#(c) the 95th percentile of X
x = qnorm(0.95, mean = 1.9, sd = 0.6)
x
```

```
## [1] 2.886912
```

```
#3 X~Exp(0.2)
#(a) P(X>5)
Prob = 1 - pexp(5, 0.2)
Prob
```

```
## [1] 0.3678794
```

```
#(b) P(1.4 <= X <= 4.2)
Prob = pexp(4.2, 0.2) - pexp(1.4, 0.2)
Prob
```

```
## [1] 0.3240732
```

```
#(c) P(1.4 < X < 4.2)
Prob = pexp(4.2, 0.2) - pexp(1.4, 0.2)
Prob
```

```
## [1] 0.3240732
```

```
#4
unit_price = 10
unit_cost = 7.5
unit_refund = 2.5

orders = seq(0, 250, 5)
m = length(orders)

set.seed(1)
n = 100000
avg_profits = rep(0, m)
loss_prob = rep(0, m)
for (j in 1:m)
{
    order_size = orders[j]
    total_unit_cost = unit_cost * order_size
    demand = rnorm(n, 200, 40)
    unit_sold = pmin(order_size, demand)
    revenue = unit_price * unit_sold

    returns = rep(0 ,n)
    for (i in 1:n)
    {
        if(demand[i] < order_size)
            returns[i] = order_size - demand[i]
    }
    total_refund = unit_refund * returns

    profit = revenue - total_unit_cost + total_refund
    avg_profits[j] = mean(profit)
    losses = profit[profit<0]
    loss_prob[j] = length(losses)/n
}
#df = data.frame(Order_size = orders, Average_Profit = avg_profits)
df = data.frame(Order_size = orders, Loss_prob = loss_prob)
df
```

```
##    Order_size Loss_prob
## 1           0   0.00000
## 2           5   0.00000
## 3          10   0.00001
## 4          15   0.00000
## 5          20   0.00001
## 6          25   0.00000
## 7          30   0.00000
## 8          35   0.00002
## 9          40   0.00002
## 10         45   0.00001
## 11         50   0.00000
## 12         55   0.00003
## 13         60   0.00004
## 14         65   0.00002
## 15         70   0.00005
```

```
## 16         75   0.00015
## 17         80   0.00005
## 18         85   0.00020
## 19         90   0.00023
## 20         95   0.00026
## 21        100   0.00054
## 22        105   0.00056
## 23        110   0.00076
## 24        115   0.00101
## 25        120   0.00135
## 26        125   0.00179
## 27        130   0.00238
## 28        135   0.00301
## 29        140   0.00388
## 30        145   0.00515
## 31        150   0.00634
## 32        155   0.00811
## 33        160   0.01006
## 34        165   0.01183
## 35        170   0.01471
## 36        175   0.01885
## 37        180   0.02280
## 38        185   0.02785
## 39        190   0.03335
## 40        195   0.03989
## 41        200   0.04819
## 42        205   0.05732
## 43        210   0.06681
## 44        215   0.07835
## 45        220   0.09101
## 46        225   0.10508
## 47        230   0.12079
## 48        235   0.13859
## 49        240   0.16097
## 50        245   0.17789
## 51        250   0.20174
```

```
#So the order_size = 0 leads to the smallest probability of a loss.
#The reason is that if you not order anything, then you will never get a loss
```