Lecture B2. Newsvendor 2

Sim, Min Kyu, Ph.D., mksim@seoultech.ac.kr

서울과학기술대학교 데이터사이언스학과

Discrete distribution - try out all alternatives

- In the previous newsvendor lecture, following information was given.

1. *(demand)* How many customer? *11, 12, 13, 14, or 15, equally likely*
2. *(retail price)* How much do you sell a copy at? *$2 per copy*
3. *(material cost)* How much do you pay to the wholesaler? *$1 per copy*
4. *(salvage value)* How much do you sell an unsold copy back to the wholesaler? *$0.5 per copy*

Implementation

- Following code tries the stock level $X \in \{11, 12, 13, 14, 15\}$.

```r
for (X in 11:15){
  MC_N <- 10000
  D <- sample(11:15, MC_N, replace = T) # random discrete uniform
  sales_rev <- 2*pmin(D,X) # vector level minimum
  salvage_rev <- 0.5*pmax(X-D,0) # vector level maximum
  material_cost <- 1*X
  profit <- sales_rev + salvage_rev - material_cost
  print(paste0("X: ", X, ", expected profit: ", mean(profit)))
}
```

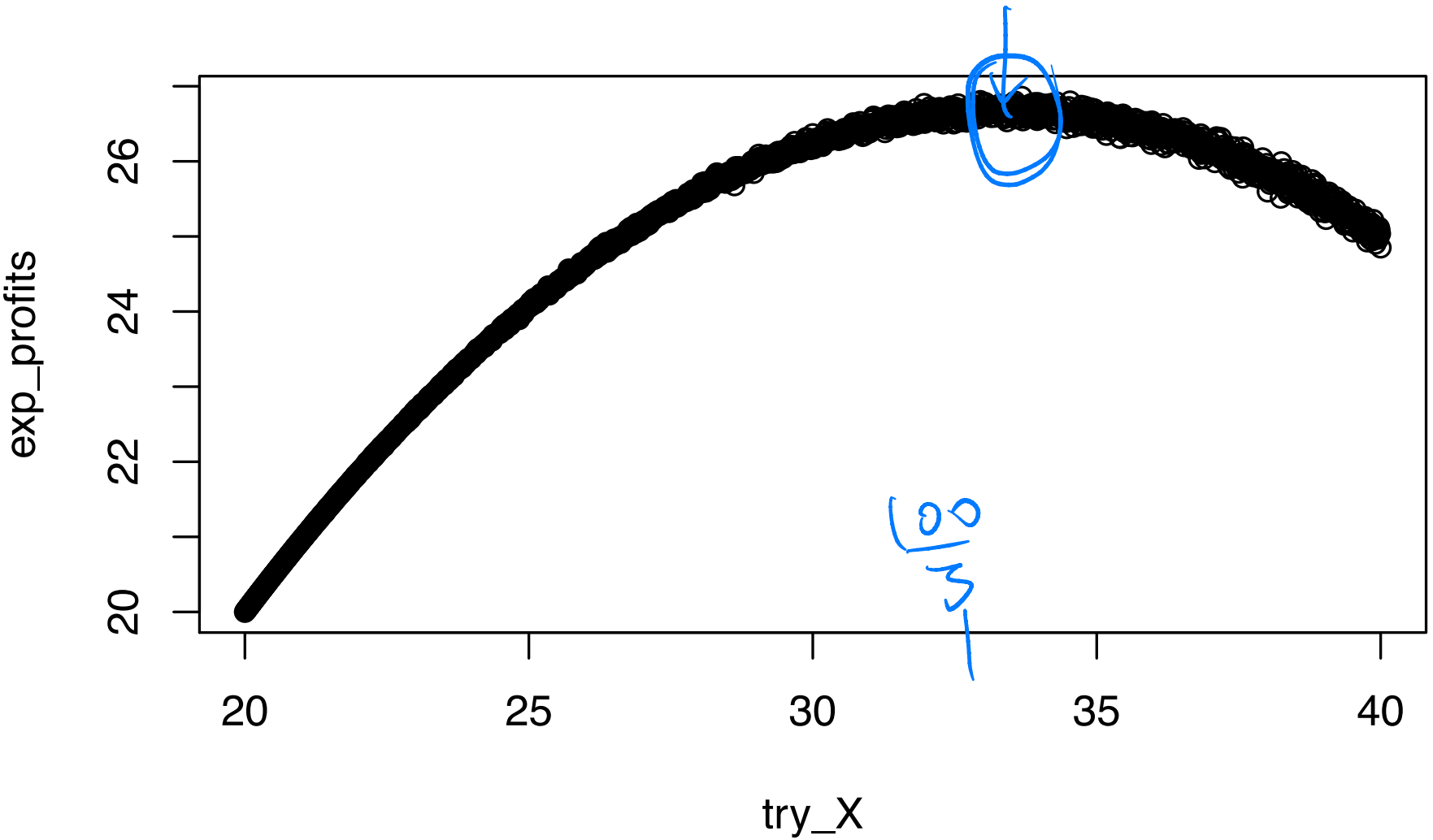
```
## [1] "X: 11, expected profit: 11"
## [1] "X: 12, expected profit: 11.6961"
## [1] "X: 13, expected profit: 12.11545"
## [1] "X: 14, expected profit: 12.20885"
## [1] "X: 15, expected profit: 11.99085"
```

Continuous distribution - grid search approach

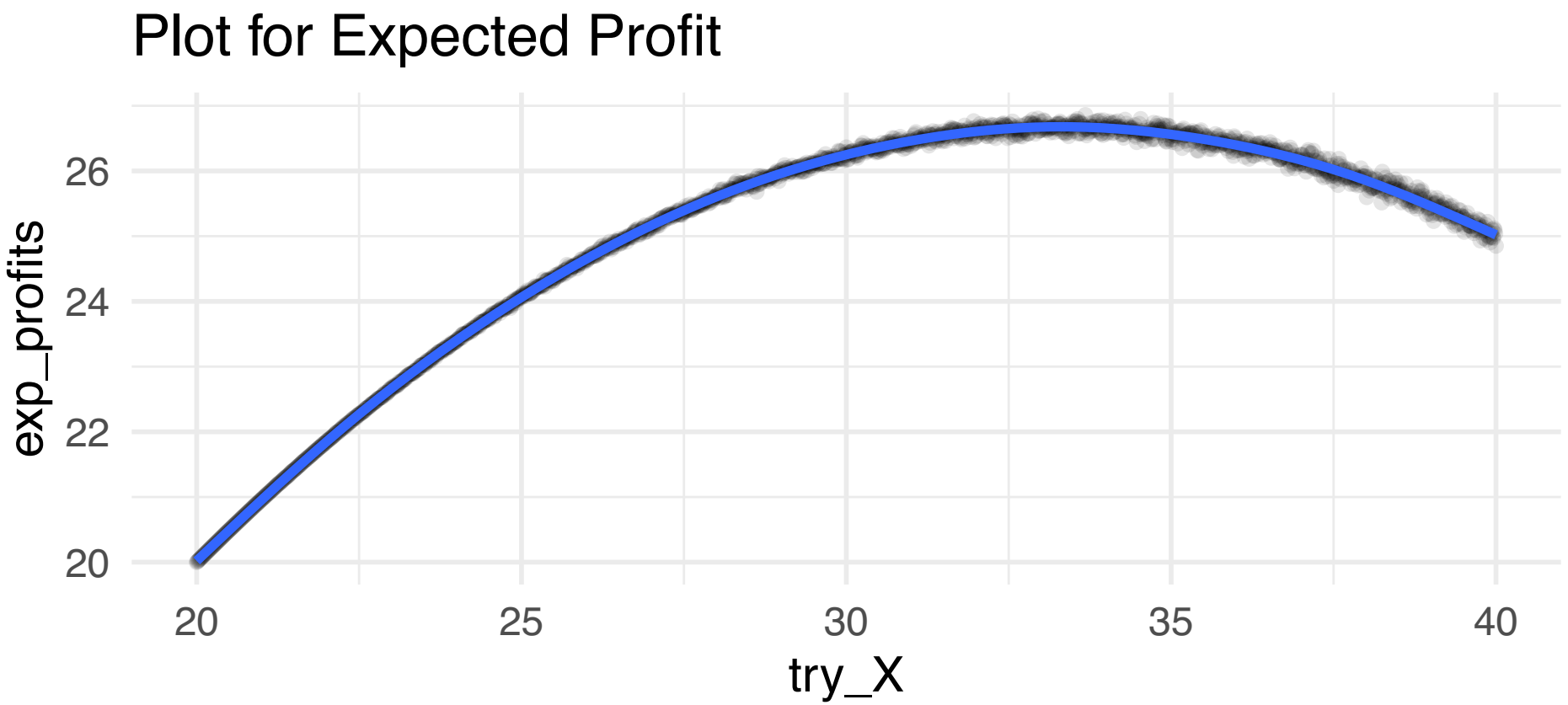
- Your brother is now selling milk. The customer demands follow $U(20, 40)$ gallons. Retail price is \$2 per gallon, material cost is \$1 per gallon, and salvage cost is \$0.5 per gallon. Find optimal stock level and expected profit.
- Notice that there are the marginal difference compared to the previous page's code.

```
try_X <- seq(from = 20, to = 40, by = 0.01)
exp_profits <- NULL
for (X in try_X){
  MC_N <- 10000
  D <- runif(MC_N, min = 20, max = 40)
  sales_rev <- 2*pmin(D,X) # vector level minimum
  salvage_rev <- 0.5*pmax(X-D,0) # vector level maximum
  material_cost <- 1*X
  exp_profit <- mean(sales_rev + salvage_rev - material_cost)
  exp_profits <- c(exp_profits, exp_profit)
}
results <- data.frame(try_X, exp_profits)
```

```r
plot(try_X, exp_profits)
```



```
library(tidyverse)
ggplot(results, aes(x=try_X, y=exp_profits)) +
  geom_point(alpha = 0.1, size = 1) + geom_smooth(size = 1) +
  labs(title="Plot for Expected Profit") +
  theme(element_text(size = 25)) +
  theme_minimal()
```



```r
idx <- which(exp_profits==max(exp_profits)) # index for maximum profit
try_X[idx] # this is optimal quantity
```

```
## [1] 33.68
```

```r
exp_profits[idx] # this is expected optimal profit
```

```
## [1] 26.85996
```

If I only had an hour to chop down a tree, I would spend the first 45 minutes sharpening my axe. - A. Lincoln