Monte Carlo Simulation

Examples

SIMULATION

Simulation can be used for

- Estimating probabilities
- Optimization (find best decisions)

Example – Rolling Dice

ROLLING DICE - EXAMPLE

- Consider rolling two dice and compute their sum
- What is the probability that the sum is 9?

Answer

- Probability is the long-run relative frequency of an outcome
- Repeat the experiment many times
- The relative frequency of the outcome will converge to the probability

ROLLING DICE - EXAMPLE

```
set.seed(123)
n <- 20
n_dice <- 2
#
# create a matrix with twenty rows, two columns
matrix1 = matrix(0, nrow=n, ncol=n_dice)

# fill the matrix with the outcomes
for(i in 1:n){
  temp <- sample(1:6, n_dice, replace=TRUE)
  matrix1[i,] = temp
}

head(matrix1)
```

repeat experiment 20 times

```
##      [,1] [,2]
## [1,]    3    6
## [2,]    3    2
## [3,]    2    6
## [4,]    3    5
## [5,]    4    6
## [6,]    6    1
```

ROLLING DICE - EXAMPLE

```
set.seed(123)
n <- 20
n_dice <- 2
#
# create a matrix with twenty rows, two columns
matrix1 = matrix(0, nrow=n, ncol=n_dice)

# fill the matrix with the outcomes
for(i in 1:n){
  temp <- sample(1:6, n_dice, replace=TRUE)
  matrix1[i,] = temp
}

head(matrix1)
```

repeat experiment 20 times

```
##      [,1] [,2]
## [1,]    3    6
## [2,]    3    2
## [3,]    2    6
## [4,]    3    5
## [5,]    4    6
## [6,]    6    1
```

```
df = data.frame(matrix1)
names1 <- c("roll1","roll2")
colnames(df) = names1
head(df)
```

```
##   roll1 roll2
## 1     3     6
## 2     3     2
## 3     2     6
## 4     3     5
## 5     4     6
## 6     6     1
```

ROLLING DICE - EXAMPLE

```r
set.seed(123)
n <- 20
n_dice <- 2
#
# create a matrix with twenty rows, two columns
matrix1 = matrix(0, nrow=n, ncol=n_dice)

# fill the matrix with the outcomes
for(i in 1:n){
  temp <- sample(1:6, n_dice, replace=TRUE)
  matrix1[i,] = temp
}

head(matrix1)
```

```
##      [,1] [,2]
## [1,]    3    6
## [2,]    3    2
## [3,]    2    6
## [4,]    3    5
## [5,]    4    6
## [6,]    6    1
```

```r
df = data.frame(matrix1)
names1 <- c("roll1","roll2")
colnames(df) = names1
head(df)
```

```
##   roll1 roll2
## 1     3     6
## 2     3     2
## 3     2     6
## 4     3     5
## 5     4     6
## 6     6     1
```

```
> # add column with the sum
> #
> df$sum = df$roll1 + df$roll2
> head(df)
  roll1 roll2 sum
1     3     6   9
2     3     2   5
3     2     6   8
4     3     5   8
5     4     6  10
6     6     1   7
```

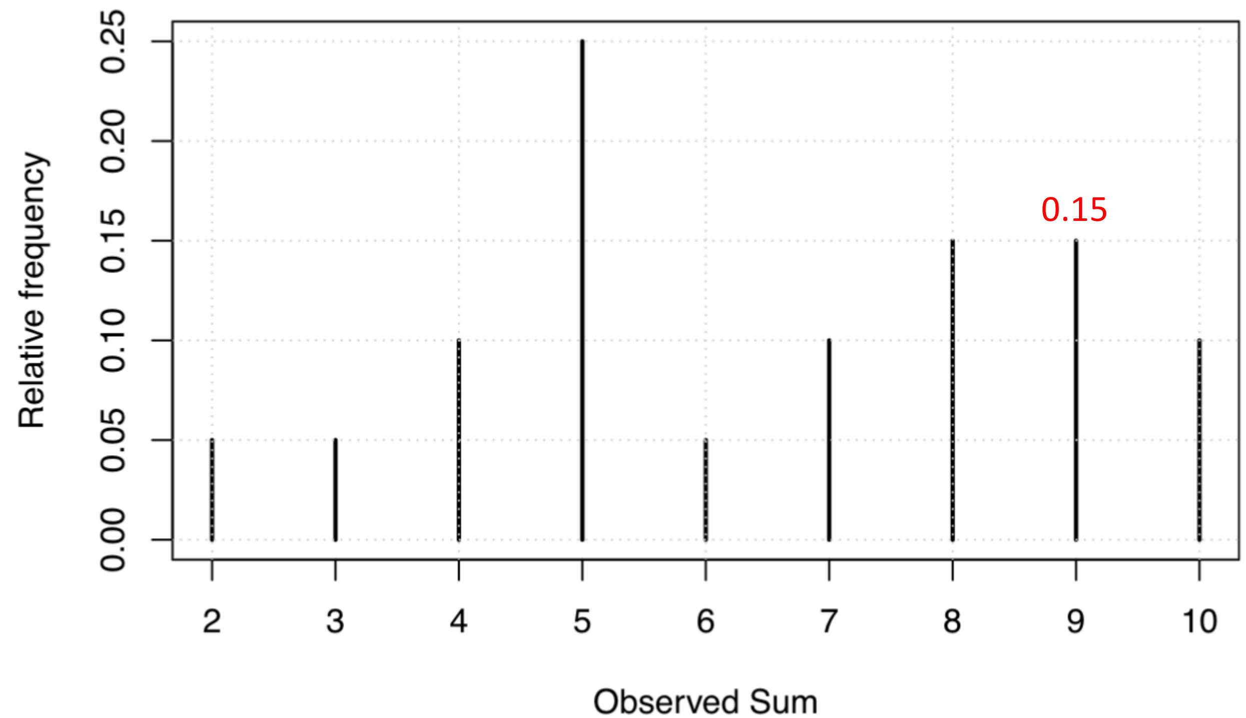
ROLLING DICE - EXAMPLE

```
> df$sum = df$roll1 + df$roll2
> head(df)
  roll1 roll2 sum
1     3     6   9
2     3     2   5
3     2     6   8
4     3     5   8
5     4     6  10
6     6     1   7
> #
> # number of times each outcome is observed
> table(df$sum)

 2  3  4  5  6  7  8  9 10 
 1  1  2  5  1  2  3  3  2 
> #
> # fraction each outcome is observed
> table(df$sum)/n

   2    3    4    5    6    7    8    9   10 
0.05 0.05 0.10 0.25 0.05 0.10 0.15 0.15 0.10 
```

ROLLING DICE - EXAMPLE

```
> df$sum = df$roll1 + df$roll2
> head(df)
  roll1 roll2 sum
1     3     6   9
2     3     2   5
3     2     6   8
4     3     5   8
5     4     6  10
6     6     1   7
> #
> # number of times each outcome is observed
> table(df$sum)

 2  3  4  5  6  7  8  9 10
 1  1  2  5  1  2  3  3  2
> #
> # fraction each outcome is observed
> table(df$sum)/n

   2    3    4    5    6    7    8    9   10
0.05 0.05 0.10 0.25 0.05 0.10 0.15 0.15 0.10
```

```
# plot relative frequencies
table1 = table(df$sum)/n
plot(table1, xlab = 'Observed Sum',
     ylab = 'Relative frequency',
     main = '20 Rolls of 2 Dice')
```



ROLLING DICE A MILLION TIMES

```
set.seed(123)
n <- 10^6
matrix1 = matrix(0, nrow=n, ncol=n_dice)
```

repeat experiment 1000000 times

```
for(i in 1:n){
  temp <- sample(1:6, n_dice, replace=TRUE)
  matrix1[i,] = temp
}

df = data.frame(matrix1)
names1 <- c("roll1","roll2")
colnames(df) = names1
```

```
df$sum = df$roll1 + df$roll2
head(df)
```

```
##   roll1 roll2 sum
## 1     3     6   9
## 2     3     2   5
## 3     2     6   8
## 4     3     5   8
## 5     4     6  10
## 6     6     1   7
```

ROLLING DICE A MILLION TIMES

```r
set.seed(123)
n <- 10^6
matrix1 = matrix(0, nrow=n, ncol=n_dice)
```

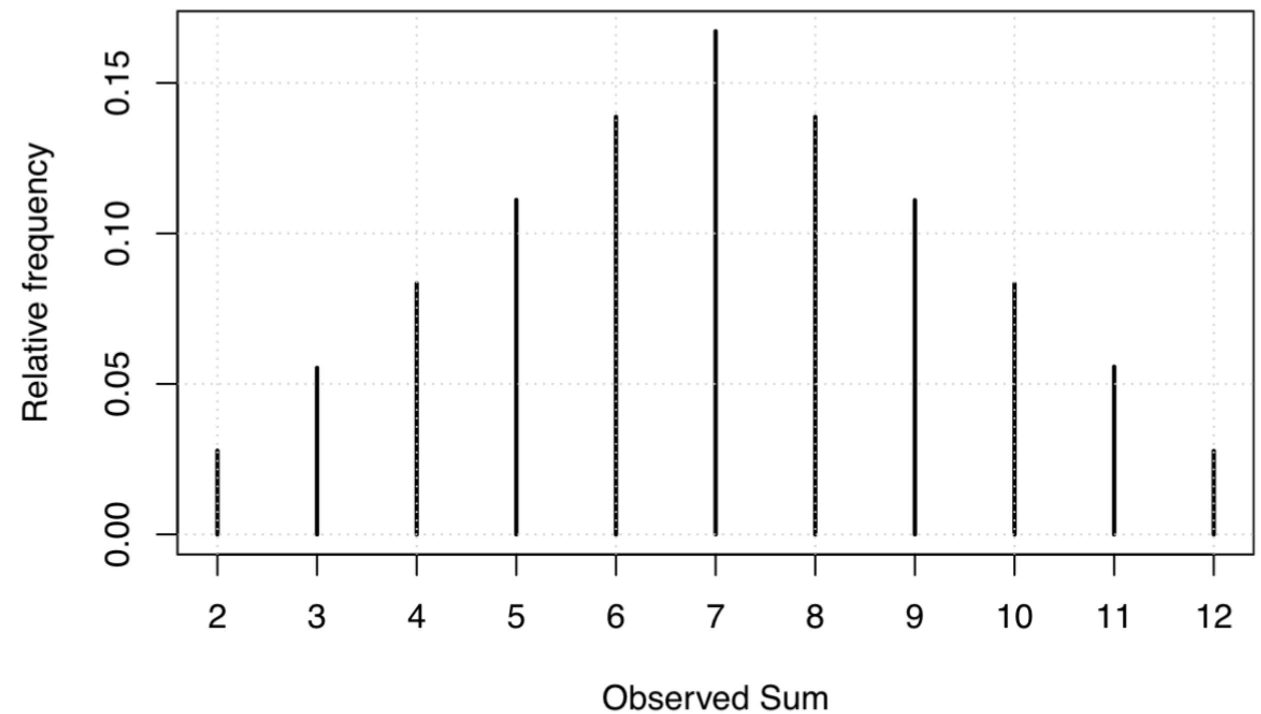
```r
for(i in 1:n){
  temp <- sample(1:6, n_dice, replace=TRUE)
  matrix1[i,] = temp
}

df = data.frame(matrix1)
names1 <- c("roll1","roll2")
colnames(df) = names1
```

```r
df$sum = df$roll1 + df$roll2
head(df)
```

```
##   roll1 roll2 sum
## 1     3     6   9
## 2     3     2   5
## 3     2     6   8
## 4     3     5   8
## 5     4     6  10
## 6     6     1   7
```

```r
table1 = table(df$sum)/n
table1
```

```
##
##        2        3        4        5        6        7        8        9
## 0.027826 0.055336 0.083405 0.111132 0.138748 0.167175 0.138725 0.111016
##       10       11       12
## 0.083221 0.055703 0.027713
```



ROLLING DICE A MILLION TIMES

```
set.seed(123)
n <- 10^6
matrix1 = matrix(0, nrow=n, ncol=n_dice)
```

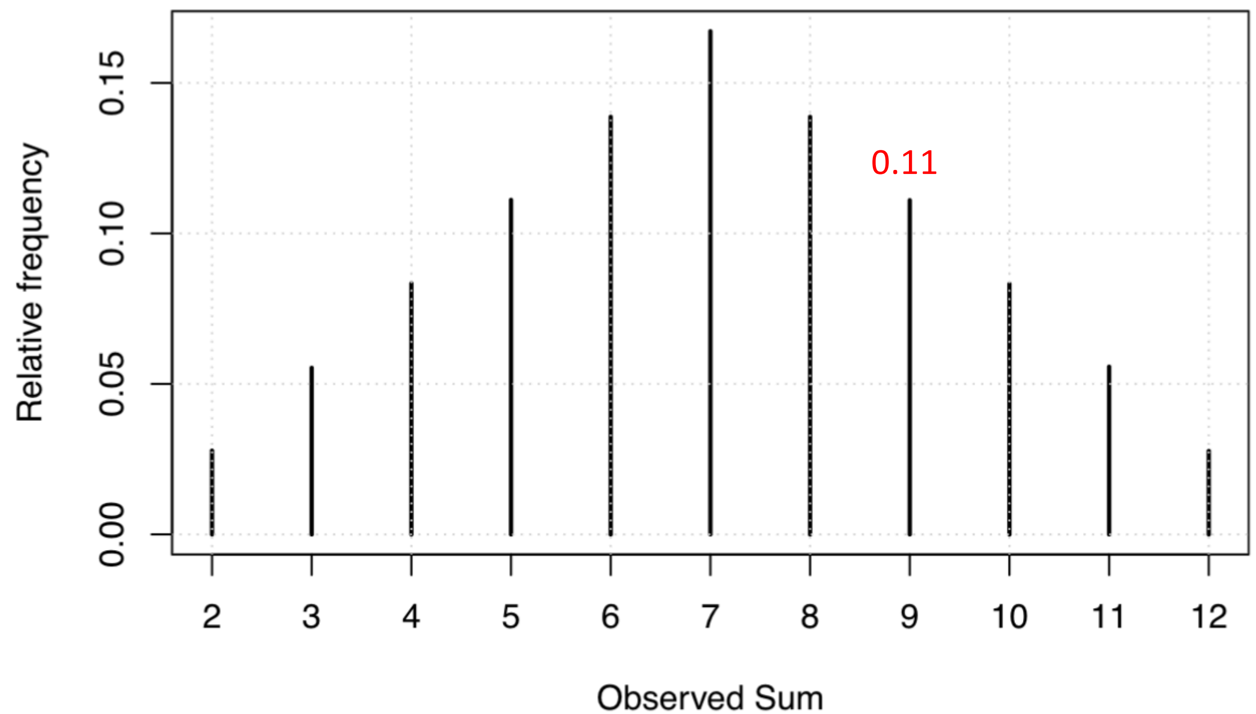
```
for(i in 1:n){
  temp <- sample(1:6, n_dice, replace=TRUE)
  matrix1[i,] = temp
}

df = data.frame(matrix1)
names1 <- c("roll1","roll2")
colnames(df) = names1
```

```
df$sum = df$roll1 + df$roll2
head(df)
```

```
##   roll1 roll2 sum
## 1     3     6   9
## 2     3     2   5
## 3     2     6   8
## 4     3     5   8
## 5     4     6  10
## 6     6     1   7
```

```
table1 = table(df$sum)/n
table1
```

```
##
##        2        3        4        5        6        7        8        9
## 0.027826 0.055336 0.083405 0.111132 0.138748 0.167175 0.138725 0.111016
##       10       11       12
## 0.083221 0.055703 0.027713
```



Example - Overbooking

OVERBOOKING - EXAMPLE

- An airline assumes that 1% of the passengers on a certain flight will not show up.
- Therefore, their policy is to sell 52 tickets for a flight that can hold up to 50 passengers (overbooking)
- What is the risk of having more passengers than seats in that flight?

OVERBOOKING - EXAMPLE

Let

X: the number of passengers that show up

$X \sim BINO(n = 52, p=0.99)$

The risk of the airline is $P[X>50]$

```
> 1-pbinom(50,52,0.99)
[1] 0.9044236
```

OVERBOOKING - EXAMPLE

```r
# probability of at most 50 passengers
pbinom(50,52,0.99)
```

```
## [1] 0.09557643
```

P[X <= 50]

```r
#
# probability of overbooking
1-pbinom(50,52,0.99)
```

```
## [1] 0.9044236
```

P[X > 50]

OVERBOOKING - EXAMPLE

```r
n_reservations <- 52
set.seed(1)
#
# n flights with 52 reservations each
n <- 10000
passengers <- rbinom(n,size=n_reservations,prob=0.99)
head(passengers,16)
```

```
##  [1] 52 52 52 50 52 51 50 51 51 52 52 52 51 52 51 52
```

```r
#
table1 = table(passengers)/n
table1
```

```
## passengers
##     47     48     49     50     51     52
## 0.0002 0.0010 0.0145 0.0838 0.3031 0.5974
```

OVERBOOKING - EXAMPLE

```r
n_reservations <- 52
set.seed(1)
#
# n flights with 52 reservations each
n <- 10000
passengers <- rbinom(n,size=n_reservations,prob=0.99)
head(passengers,16)
```

```
##  [1] 52 52 52 50 52 51 50 51 51 52 52 52 51 52 51 52
```

```r
#
table1 = table(passengers)/n
table1
```

```
## passengers
##     47     48     49     50     51     52
## 0.0002 0.0010 0.0145 0.0838 0.3031 0.5974
```

```r
df = data.frame(table1)
df
```

```
##   passengers   Freq
## 1         47 0.0002
## 2         48 0.0010
## 3         49 0.0145
## 4         50 0.0838
## 5         51 0.3031
## 6         52 0.5974
```

```r
str(df)
```

```
## 'data.frame':    6 obs. of  2 variables:
##  $ passengers: Factor w/ 6 levels "47","48","49",..: 1 2 3
##  $ Freq      : num  0.0002 0.001 0.0145 0.0838 0.3031 ...
```

```r
#
# convert factor column passengers to numeric
#
df$passengers = as.character(df$passengers)
df$passengers = as.numeric(df$passengers)
str(df)
```

```
## 'data.frame':    6 obs. of  2 variables:
##  $ passengers: num  47 48 49 50 51 52
##  $ Freq      : num  0.0002 0.001 0.0145 0.0838 0.3031 ...
```

OVERBOOKING - EXAMPLE

```
n_reservations <- 52
set.seed(1)
#
# n flights with 52 reservations each
n <- 10000
passengers <- rbinom(n,size=n_reservations,prob=0.99)
head(passengers,16)
```

```
##  [1] 52 52 52 50 52 51 50 51 51 52 52 52 51 52 51 52
```

```
#
table1 = table(passengers)/n
table1
```

```
## passengers
##     47     48     49     50     51     52
## 0.0002 0.0010 0.0145 0.0838 0.3031 0.5974
```

```
df = data.frame(table1)
df
```

```
##   passengers   Freq
## 1         47 0.0002
## 2         48 0.0010
## 3         49 0.0145
## 4         50 0.0838
## 5         51 0.3031
## 6         52 0.5974
```

```
str(df)
```

```
## 'data.frame':    6 obs. of  2 variables:
##  $ passengers: Factor w/ 6 levels "47","48","49",..: 1 2 3
##  $ Freq      : num  0.0002 0.001 0.0145 0.0838 0.3031 ...
```

```
#
# convert column passengers to numeric
#
df$passengers = as.character(df$passengers)
df$passengers = as.numeric(df$passengers)
str(df)
```

```
## 'data.frame':    6 obs. of  2 variables:
##  $ passengers: num  47 48 49 50 51 52
##  $ Freq      : num  0.0002 0.001 0.0145 0.0838 0.3031 ...
```

OVERBOOKING - EXAMPLE

```
df = data.frame(table1)
df
```

```
##   passengers   Freq
## 1         47 0.0002
## 2         48 0.0010
## 3         49 0.0145
## 4         50 0.0838
## 5         51 0.3031
## 6         52 0.5974
```

```
str(df)
```

```
## 'data.frame':    6 obs. of  2 variables:
##  $ passengers: Factor w/ 6 levels "47","48","49",..: 1 2 3
##  $ Freq      : num  0.0002 0.001 0.0145 0.0838 0.3031 ...
```

```
#
# convert column passengers to numeric
#
df$passengers = as.character(df$passengers)
df$passengers = as.numeric(df$passengers)
str(df)
```

```
## 'data.frame':    6 obs. of  2 variables:
##  $ passengers: num  47 48 49 50 51 52
##  $ Freq      : num  0.0002 0.001 0.0145 0.0838 0.3031 ...
```

```
# overbooking probability P[X > 50]
#
df2 = df[df$passengers>50,]
sum(df2$Freq)
```

```
## [1] 0.9005
```

OVERBOOKING - EXAMPLE

```r
# n = one million replicates
#
n <- 10^6
passengers <- rbinom(n,n_reservations,0.99)
table1 = table(passengers)/n
df = data.frame(table1)
df$passengers = as.character(df$passengers)
df$passengers = as.numeric(df$passengers)
df
```

```
##   passengers     Freq
## 1         46 0.000016
## 2         47 0.000164
## 3         48 0.001611
## 4         49 0.013541
## 5         50 0.079795
## 6         51 0.311894
## 7         52 0.592979
```

```r
# overbooking probability P[X > 50]
#
df2 = df[df$passengers>50,]
sum(df2$Freq)
```

```
## [1] 0.904873
```

```r
# theoretical probability of overbooking
1-pbinom(50,52,0.99)
```

```
## [1] 0.9044236
```

Example - New Product

EXAMPLE 1

PortaCom manufactures personal computers and related equipment. PortaCom's product design group developed a prototype for a new high-quality portable printer.

Preliminary marketing and financial analyses provided the following estimates for the first year

- Selling price \$249 per unit
- Administrative cost \$400,000
- Advertising cost \$600,000

The preliminary analysis assumed that direct labor costs \$45, the cost of parts is \$90, and a demand equal to 15000 units. What is the estimated profit from this preliminary analysis?

The resulting profit for PortaCom is given by

$$
\begin{aligned}
\text{Profit} &= (\text{price} - \text{labor cost} - \text{parts cost}) * \text{demand} - 1{,}000{,}000 \\
&= (249 - 45 - 90) * 15000 - 1{,}000{,}000 \\
&= 710{,}000
\end{aligned}
$$

EXAMPLE 1

More detailed analysis found that direct labor costs could vary between \$43 and \$47, the cost of parts between \$80 and \$100, and the demand between 1500 and 28500 units. What is the best and worst estimated profit from this detailed analysis?

WORST SCENARIO

Profit	=	(price	-	labor cost	-	parts cost	) *	demand	-	1,000,000
	=	(249	-	47	-	100	) *	1500	-	1,000,000
	=	-847,000								

EXAMPLE 1

More detailed analysis found that direct labor costs could vary between $43 and $47, the cost of parts between $80 and $100, and the demand between 1500 and 28500 units. What is the best and worst estimated profit from this detailed analysis?

WORST SCENARIO

Profit	=	(price	-	labor cost	-	parts cost	) *	demand	-	1,000,000
	=	(249	-	47	-	100	) *	1500	-	1,000,000
	=	-847,000								

BEST SCENARIO

Profit	=	(price	-	labor cost	-	parts cost	) *	demand	-	1,000,000
	=	(249	-	43	-	80	) *	28500	-	1,000,000
	=	2,591,000								

EXAMPLE 1

It is found that the estimated profit may be between \$ -847 and \$ 2,591, with a base value equal to \$ 710. However there are more questions.

- What other scenarios are plausible?
- What is the probability associated with these scenarios?
- What is the probability of a loss?
- If a loss occur, how large can it be?

EXAMPLE 1

To answer these questions the company assumed the following.
The cost of direct labor, the cost of parts, and the first-year demand for the printer are not known with certainty and are considered random variables.

- The direct labor cost will range as described by a discrete probability function

Labor cost per unit	Probability
43	0.1
44	0.2
45	0.4
46	0.2
47	0.1

- The parts cost will range from \$80 to \$100 per unit, and that values between \$80 and \$100 are equally likely. Thus a uniform distribution `UNIF(80,100)` will be used.
- The first-year demand is estimated to be normal with mean 15000 units and standard deviation 4500 units, `NORM(15000,4500)`

EXAMPLE 1

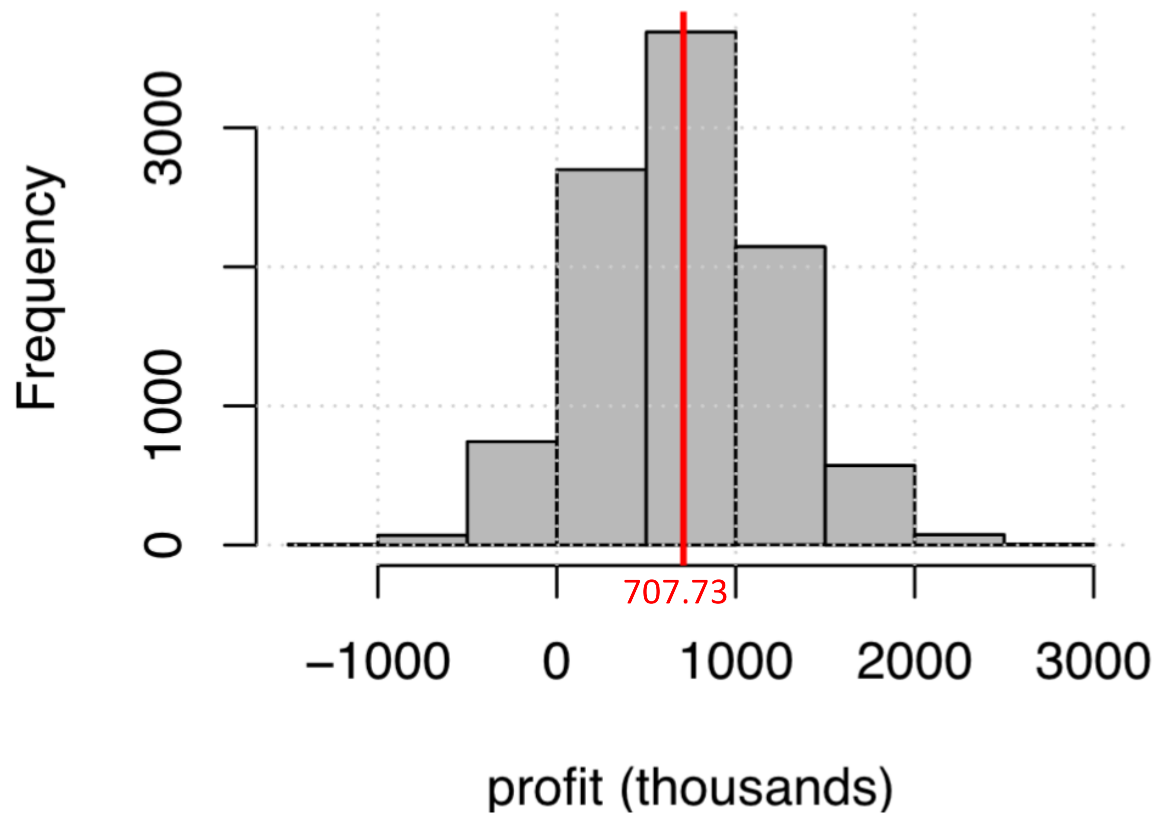
```
price = 249
fixedcost = 1000000
#
# get n=10000 observations from each random variable
#
n=10000
set.seed(1)
#
# sample from assumed distributions
#
labor = sample(c(43,44,45,46,47),
               c(0.1,0.2,0.4,0.2,0.1),
               size=n,replace=T)
#
material = runif(n,80,100)
#
demand = rnorm(n,15000,4500)
```

EXAMPLE – NEW PRODUCT

```r
# find 10000 profit values
#
profit = (price-labor-material)*demand - fixedcost
#
# scale profit in 000s
#
profit2 = profit/1000
avg2 = mean(profit2)
avg2
```

```
## [1] 707.7353
```

```r
hist(profit2,xlab="profit (thousands)",
     main="",col="gray")
abline(v=avg2,col="red",lwd=2)
```

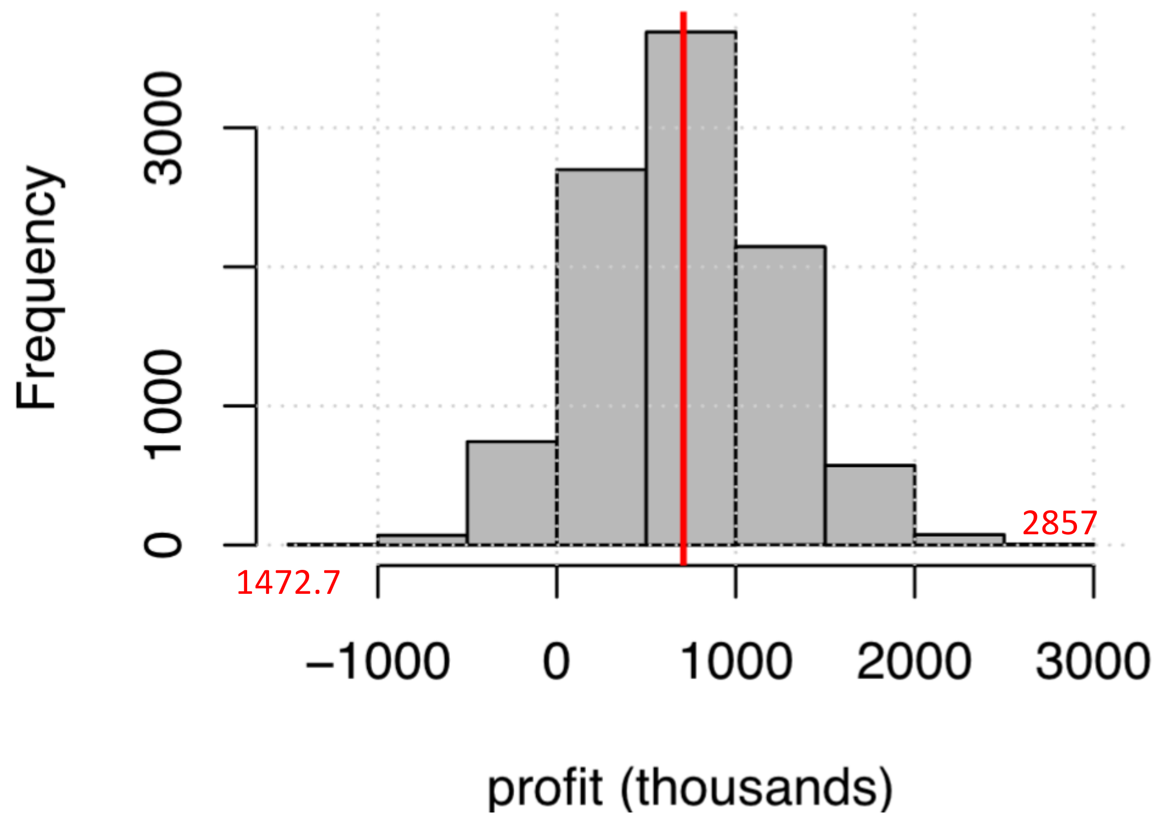


EXAMPLE – NEW PRODUCT

```r
# profit stats
#
summary(profit)
```

```
##      Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1472681   357025   695410   707735  1050616  2856900
```

```r
#
# largest loss: 1472681
# largest profit: 2856900
# most likely profit: 695410
# average profit: 707735
```



EXAMPLE – HOW RISKY IS THIS PROJECT?

```
# number of observed losses
losses = profit[profit<0]
m = length(losses)
m
```

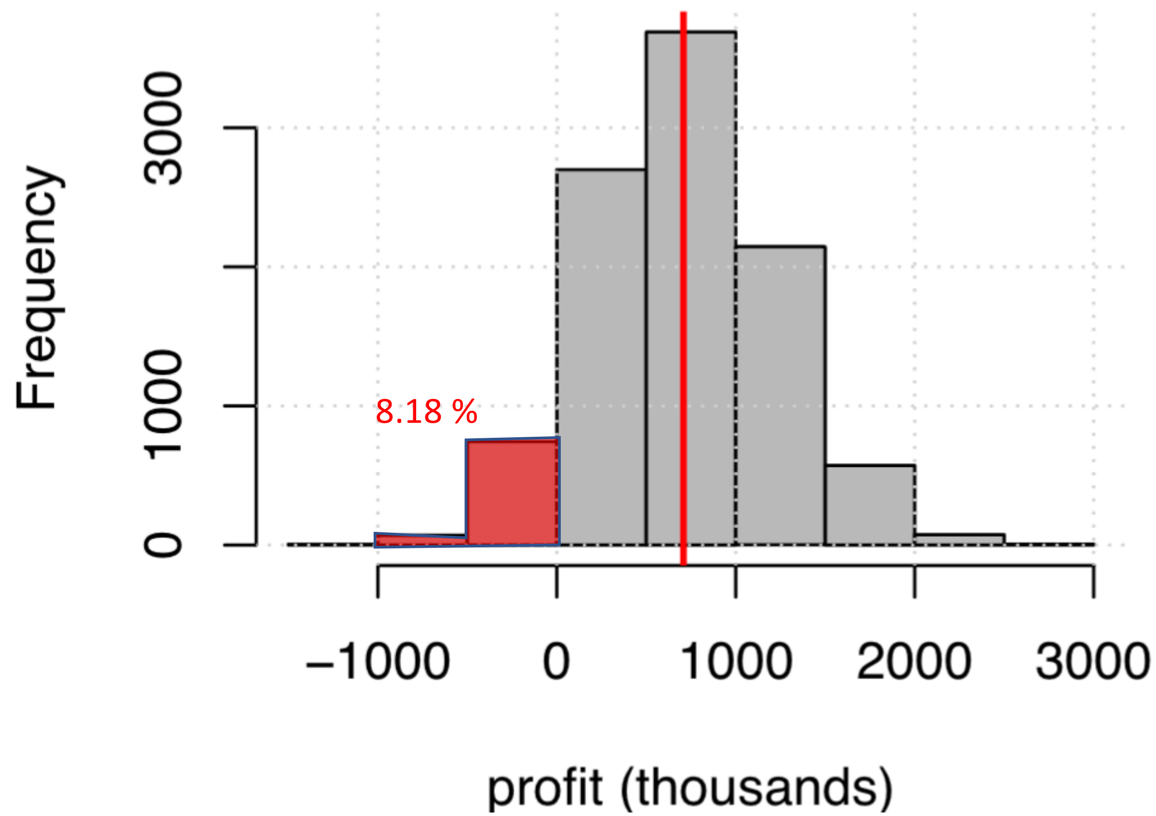
```
## [1] 818
```

```
#
# fraction of losses
obsf = m/n
obsf
```

```
## [1] 0.0818
```

```
sd(profit)
```

```
## [1] 515764.8
```



EXAMPLE – HOW RISKY IS THIS PROJECT?

```
# profit stats
#
summary(profit)
```

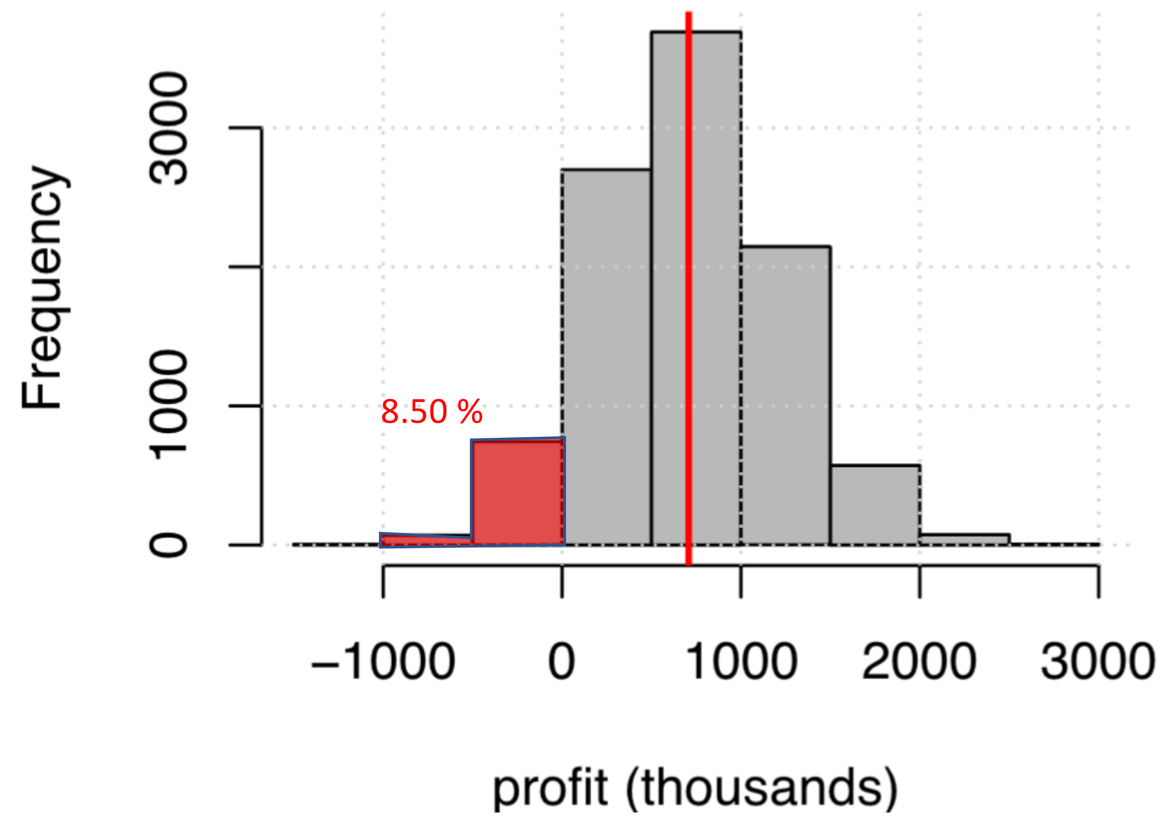
```
##      Min.  1st Qu.   Median     Mean
## -1472681   357025   695410   707735
```

```
sd(profit)
```

```
## [1] 515764.8
```

```
#
#
# Probab of a loss (if normality assumed)
pnorm(0,707735,515764.8)
```

```
## [1] 0.08499983
```



Example – Inventory

EXAMPLE 2 - INVENTORY

Consider a company that buys and sells a product. Each unit costs \$75 and sells for \$125 thus, the gross profit is \$50 for each unit sold. Monthly demand D is normal with a mean of 100 units and a standard deviation of 20 units.

The company receives monthly deliveries from its supplier and replenishes its inventory to a level of Q at the beginning of each month.

- If the monthly demand D is less than the replenishment level Q, an inventory holding cost of \$15 is charged for each unit that is not sold.
- If the monthly demand D is greater than Q, a stockout occurs and a shortage cost of \$30 is charged for each unit of demand that cannot be satisfied.

EXAMPLE 2 - INVENTORY

Management would like to use a simulation model to determine the average monthly net profit resulting from using a particular replenishment level. The decision variable is the replenishment level, Q. The random variable is the monthly demand, D.

The performance measure is the average monthly profit which is given by

$$\text{Profit} = \begin{cases} 50D - 15\,(Q - D) & \text{if} \quad D \leq Q \\ 50Q - 30\,(D - Q) & \text{if} \quad Q < D \end{cases}$$

It is of interest to answer the following questions

- What is the replenishment level Q that results in the largest profit?
- What is the largest expected profit?
- What is the probability that the monthly profit does not exceed 4000 dollars?

EXAMPLE 2 - INVENTORY

```
margin = 50
icost = 15                 inventory cost
scost = 30                 shortage cost
#
# policy
Q = 100                    replenishment level
#
# generate n=10000 random demands
#
n = 10000
set.seed(1)
demand = rnorm(n,100,20)
#
profit = rep(0,n)
for (i in 1:n)
{
   if (demand[i] <= Q)
   {
      profit[i] = margin*demand[i] -
                  icost*(Q-demand[i])
   }
   else profit[i] = margin*Q - scost*(demand[i]-Q)
}
```

$$\text{if } D \leq Q \quad \text{Profit} = 50D - 15\,(Q - D)$$

$$\text{if } Q < D \quad \text{Profit} = 50Q - 30\,(D - Q)$$

EXAMPLE 2 - INVENTORY

```
margin = 50
icost = 15                    inventory cost
scost = 30                    shortage cost
#
# policy
Q = 100                       replenishment level
#
# generate n=10000 random demands
#
n = 10000
set.seed(1)
demand = rnorm(n,100,20)
#
profit = rep(0,n)
for (i in 1:n)
{
   if (demand[i] <= Q)
   {
      profit[i] = margin*demand[i] -
                  icost*(Q-demand[i])
   }
   else profit[i] = margin*Q - scost*(demand[i]-Q)
}
```

```
summary(profit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   227.3  3949.4  4420.5  4231.8  4733.3  5000.0
```

```
sd(profit)
```

```
## [1] 685.2462
```

```
#
# Estimate the probab of a profit < 4000
# the number of times it was observed
#
losses = profit[profit<4000]
m = length(losses)
m
```

```
## [1] 2698
```

```
obsf = m/n
obsf
```

```
## [1] 0.2698
```

```
#
# Probab of a profit < 4000 is 0.27
```

EXAMPLE 2 - INVENTORY

main loop

```
set.seed(1)
n = 100000
Q = seq(100,200,1)      try replenishment levels
mprofit = NULL
for(q in Q)
{
   demand = rnorm(n,100,20)
   profit = rep(0,n)
   for (i in 1:n)
   {
      if (demand[i] <= q)
      {
         profit[i] = margin*demand[i] -
            icost*(q-demand[i])
      }
      else profit[i] = margin*q -
            scost*(demand[i]-q)
   }
   mprofit = c(mprofit,mean(profit))
}
```

```
# dataframe with the profit for each Q
#
df = data.frame(Q=Q,mprofit=mprofit)
head(df)
```

```
##       Q  mprofit
## 1 100 4238.844
## 2 101 4273.124
## 3 102 4305.241
## 4 103 4330.663
## 5 104 4357.123
## 6 105 4379.505
```

```
# find Q giving max of all mean profits
#
idx <- which.max(df$mprofit)
idx
```

```
## [1] 19
```

```
df[19,]
```

```
##       Q  mprofit
## 19 118 4543.432
```

```
bestQ <- Q[19]
bestQ
```

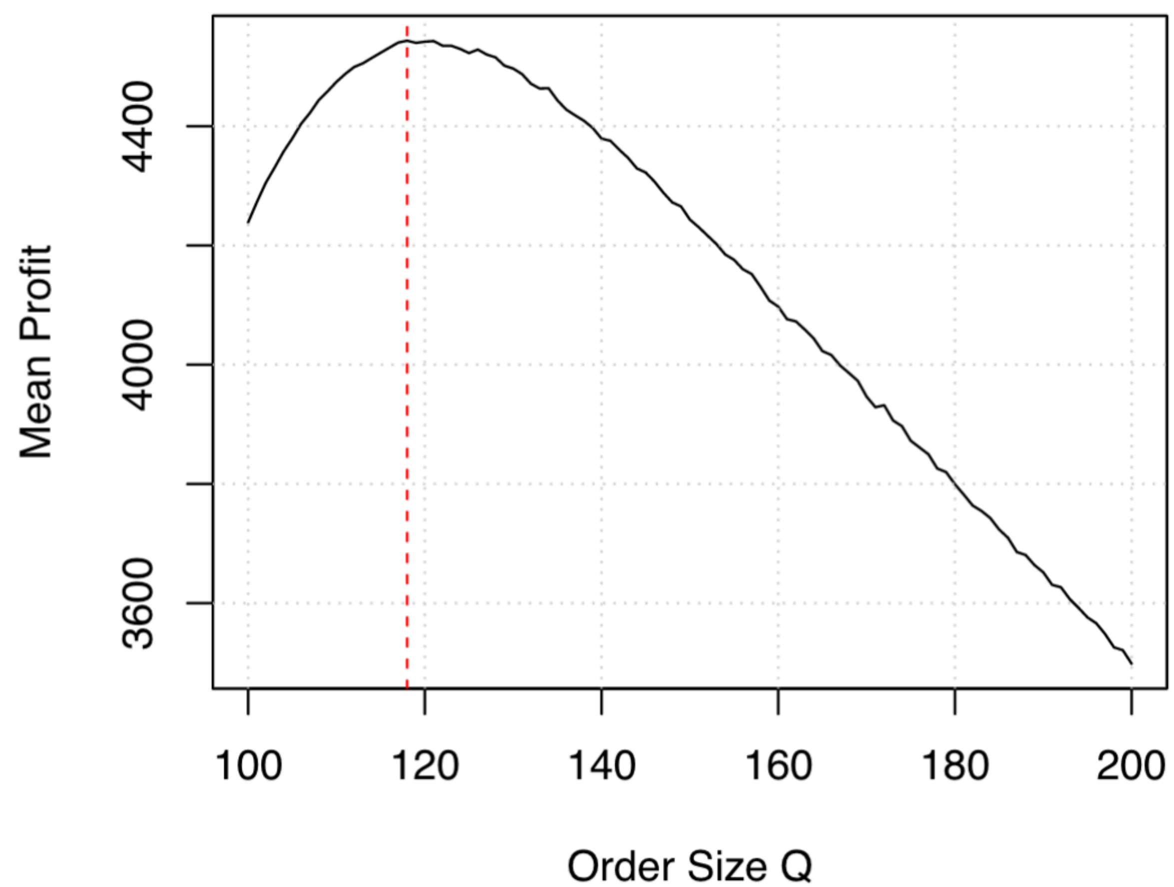
```
## [1] 118
```

EXAMPLE 2 - INVENTORY

```
bestQ <- Q[19]
bestQ
```

```
## [1] 118
```

```
plot(mprofit~Q,type="l",
     xlab="Order Size Q",ylab="Mean Profit")
abline(v=bestQ,col="red",lty=2)
```



Example – Auto Insurance

EXAMPLE 2 – AUTO INSURANCE

Suppose you consider purchasing auto insurance. The insurance has a \$1000 deductible. That is, if you have an accident and the damage is less than \$1000, you pay for it. However, if the damage is greater than \$1000, you pay the first \$1000 and the insurance pays the remaining amount. Assume that for each year the is a probability 0.025 that you will have an accident. Also, assume that you would not have more than one accident per year. If you have an accident, the damage amount is normally distributed with mean \$3000 and standard deviation \$1200.

EXAMPLE 2 – AUTO INSURANCE

1. The average amount you pay for damages to your car (some years there is no accident and the amount to pay is zero).
2. What is the average amount you have to pay for damages to your car (if you have an accident).
3. What is the probability that you have to pay less than \$500?
4. What is the probability that you have to pay more than \$600?
5. What is the probability that you have to pay the deductible?
6. Find a 95% confidence interval for the average amount you pay out of pocket.
7. Find a 95% confidence interval for the average amount have to you pay (if you have an accident).
8. Compare the average amount you pay and the 95% confidence interval for the average amount you pay when the deductible vary from \$500 to \$4000 in multiples of \$500. Including \$0 when there is no accident.

EXAMPLE 2 – AUTO INSURANCE

```r
# simulate 10000 years of car usage
set.seed(1)
n=10000
# probability of a car accident
p = 0.025
deductible = 1000
#
# vector of 0-1 indicating a year with a car accident
accident = rbinom(n,1,0.025)
table(accident)
```

```
## accident
##    0    1
## 9734  266
```

EXAMPLE 2 – AUTO INSURANCE

```r
# for each year find out_of_pocket
#
damage_cost = rep(0,n)
out_of_pocket = rep(0,n)
for(i in 1:n)
{
  if (accident[i] == 1)
    {
      damage_cost[i] = rnorm(1,3000,1200)
      if (damage_cost[i] >= 1000) out_of_pocket[i] = 1000
      else if (damage_cost[i] > 0 & damage_cost[i] <1000) out_of_pocket[i] = damage_cost[i]
  }
}
```

EXAMPLE 2 – AUTO INSURANCE

```r
accident = as.factor(accident)
df = data.frame(accident,damage_cost,out_of_pocket)
summary(df)
```

```
##  accident  damage_cost       out_of_pocket
##  0:9734   Min.   :   0.0   Min.   :   0.00
##  1: 266   1st Qu.:   0.0   1st Qu.:   0.00
##           Median :   0.0   Median :   0.00
##           Mean   :  78.6   Mean   :  26.36
##           3rd Qu.:   0.0   3rd Qu.:   0.00
##           Max.   :5741.6   Max.   :1000.00
```

```r
# dataframe excluding years with no accident
#
df2 = df[df$accident==1,]
head(df2)
```

```
##     accident damage_cost out_of_pocket
## 18         1   1180.3520     1000.0000
## 104        1   3754.9694     1000.0000
## 111        1    986.1672      986.1672
## 121        1   4415.7374     1000.0000
## 139        1   4341.1854     1000.0000
## 293        1   1514.7169     1000.0000
```

EXAMPLE 2 – AUTO INSURANCE

```r
accident = as.factor(accident)
df = data.frame(accident,damage_cost,out_of_pocket)
summary(df)
```

```
##  accident  damage_cost      out_of_pocket
##  0:9734   Min.   :   0.0   Min.   :   0.00
##  1: 266   1st Qu.:   0.0   1st Qu.:   0.00
##           Median :   0.0   Median :   0.00
##           Mean   :  78.6   Mean   :  26.36
##           3rd Qu.:   0.0   3rd Qu.:   0.00
##           Max.   :5741.6   Max.   :1000.00
```

```r
# dataframe excluding years with no accident
#
df2 = df[df$accident==1,]
summary(df2)
```

```
##  accident  damage_cost      out_of_pocket
##  0:  0    Min.   :  58.95   Min.   :  58.95
##  1:266    1st Qu.:2132.47   1st Qu.:1000.00
##           Median :2900.43   Median :1000.00
##           Mean   :2954.76   Mean   : 990.98
##           3rd Qu.:3752.94   3rd Qu.:1000.00
##           Max.   :5741.65   Max.   :1000.00
```

EXAMPLE 2 – AUTO INSURANCE

```r
# 1. Average amount to pay (include $0 when there is no accident)
avg_out_of_pocket = mean(df$out_of_pocket)
avg_out_of_pocket
```

```
## [1] 26.35994
```

```r
sd_out_of_pocket = sd(df$out_of_pocket)
sd_out_of_pocket
```

```
## [1] 159.8831
```

```r
# 2. Average amount to pay (if accident occurs)
avg_out_of_pocket2 = mean(df2$out_of_pocket)
avg_out_of_pocket2
```

```
## [1] 990.9751
```

```r
sd_out_of_pocket2 = sd(df2$out_of_pocket)
sd_out_of_pocket2
```

```
## [1] 70.80511
```

EXAMPLE 2 – AUTO INSURANCE

```
# frequency table for questions 3 and 4
#
h1 = hist(df2$out_of_pocket)
str(h1)
```

```
## List of 6
##  $ breaks  : num [1:11] 0 100 200 300 
##  $ counts  : int [1:10] 1 0 0 0 0 1 2 
##  $ density : num [1:10] 3.76e-05 0.00 
##  $ mids    : num [1:10] 50 150 250 350
```

```
breaks = h1$breaks
m = length(breaks)
rbounds = breaks[-1]
lbounds = breaks[-m]
df4 = data.frame(from=lbounds,to=rbounds,
                 midpoint = h1$mids,
                 abs.frequency = h1$counts)
```

```
##    from   to midpoint abs.frequency
## 1     0  100       50             1
## 2   100  200      150             0
## 3   200  300      250             0
## 4   300  400      350             0
## 5   400  500      450             0
## 6   500  600      550             1
## 7   600  700      650             2
## 8   700  800      750             1
## 9   800  900      850             1
## 10  900 1000      950           260
```

EXAMPLE 2 – AUTO INSURANCE

```r
# 3. probability that driver has to pay less than $500 (if accident occurs)
1/266
```

```
## [1] 0.003759398
```

```r
#
# 4. probability that driver has to pay more than $600 (if accident occurs)
264/266
```

```
## [1] 0.9924812
```

```
##      from   to midpoint abs.frequency
## 1       0  100       50             1
## 2     100  200      150             0
## 3     200  300      250             0
## 4     300  400      350             0
## 5     400  500      450             0
## 6     500  600      550             1
## 7     600  700      650             2
## 8     700  800      750             1
## 9     800  900      850             1
## 10    900 1000      950           260
```

EXAMPLE 2 – AUTO INSURANCE

```
# 5. probability that driver has to pay the deductible (if accident occurs)
pay_deductible = df2$out_of_pocket[df2$out_of_pocket==1000]
length(pay_deductible)
```

```
## [1] 257
```

```
length(pay_deductible)/266
```

```
## [1] 0.9661654
```

```
##    from   to midpoint abs.frequency
## 1     0  100       50             1
## 2   100  200      150             0
## 3   200  300      250             0
## 4   300  400      350             0
## 5   400  500      450             0
## 6   500  600      550             1
## 7   600  700      650             2
## 8   700  800      750             1
## 9   800  900      850             1
## 10  900 1000      950           260
```

EXAMPLE 2 – AUTO INSURANCE

```
# 6. 95% CI on the average amount to pay (include $0 when there is no accident)
lb = avg_out_of_pocket - 1.96*sd_out_of_pocket/sqrt(n)
ub = avg_out_of_pocket + 1.96*sd_out_of_pocket/sqrt(n)
c(lb,ub)
```

```
## [1] 23.22623 29.49365
```

```
#
# 7. 95% CI on the average amount to pay (if accident occurs)
lb2 = avg_out_of_pocket2 - 1.96*sd_out_of_pocket2/sqrt(n)
ub2 = avg_out_of_pocket2 + 1.96*sd_out_of_pocket2/sqrt(n)
c(lb2,ub2)
```

```
## [1] 989.5873 992.3628
```

EXAMPLE 2 – AUTO INSURANCE

```r
# 8. DEDUCTIBLE = 500, 1000,...,4000
#
deductible_vals = seq(500,4000,500)
deductible_vals
```

```
## [1]  500 1000 1500 2000 2500 3000
```

```r
n=10000
#
set.seed(1)
accident = rbinom(n,1,0.025)
# create empty matrix
matrix1 = NULL
j = 1
```

```r
for(deductible in deductible_vals)
{
  damage_cost = rep(0,n)
  out_of_pocket = rep(0,n)
  for(i in 1:n)
  {
    if (accident[i] == 1)
    {
      damage_cost[i] = rnorm(1,3000,1200)
      if (damage_cost[i] > deductible) out_of_pocket[i] = deductible
      else if (damage_cost[i] > 0 & damage_cost[i] < deductible)
              out_of_pocket[i] = damage_cost[i]
    }
  }

}
```

EXAMPLE 2 – AUTO INSURANCE

```r
# 8. DEDUCTIBLE = 500, 1000,...,4000
#
deductible_vals = seq(500,4000,500)
deductible_vals
```

```
## [1]  500 1000 1500 2000 2500 3000
```

```r
n=10000
#
set.seed(1)
accident = rbinom(n,1,0.025)
# create empty matrix
matrix1 = NULL
j = 1
```

```r
for(deductible in deductible_vals)
{
  damage_cost = rep(0,n)
  out_of_pocket = rep(0,n)
  for(i in 1:n)
  {
    if (accident[i] == 1)
    {
      damage_cost[i] = rnorm(1,3000,1200)
      if (damage_cost[i] > deductible) out_of_pocket[i] = deductible
      else if (damage_cost[i] > 0 & damage_cost[i] < deductible)
              out_of_pocket[i] = damage_cost[i]
    }
  }
  avg_out_of_pocket = mean(out_of_pocket)
  sd_out_of_pocket = sd(out_of_pocket)
  # z = qt(0.975,n-1)
  lb = avg_out_of_pocket - 1.96*sd_out_of_pocket/sqrt(n)
  ub = avg_out_of_pocket + 1.96*sd_out_of_pocket/sqrt(n)
  column_values = c(avg_out_of_pocket,sd_out_of_pocket,lb,ub)
  matrix1 = cbind(matrix1,column_values)
  colnames(matrix1)[j] = deductible
  j = j + 1
}
```

EXAMPLE 2 – AUTO INSURANCE

```
matrix1
```

```
##           500      1000      1500      2000      2500      3000      3500
## [1,] 13.25590  26.09909  38.19261  49.81629  57.98257  66.29556  73.43757
## [2,] 80.31360 159.05547 234.27351 307.30282 362.66445 416.12407 467.05523
## [3,] 11.68175  22.98161  33.60084  43.79316  50.87435  58.13953  64.28329
## [4,] 14.83004  29.21658  42.78437  55.83943  65.09079  74.45159  82.59185
```

```
df1 = data.frame(matrix1)
df1
```

```
##       X500     X1000     X1500     X2000     X2500     X3000     X3500
## 1 13.25590  26.09909  38.19261  49.81629  57.98257  66.29556  73.43757
## 2 80.31360 159.05547 234.27351 307.30282 362.66445 416.12407 467.05523
## 3 11.68175  22.98161  33.60084  43.79316  50.87435  58.13953  64.28329
## 4 14.83004  29.21658  42.78437  55.83943  65.09079  74.45159  82.59185
```

```
colnames(df1) = deductible_vals
df1
```

```
##        500      1000      1500      2000      2500      3000      3500
## 1 13.25590  26.09909  38.19261  49.81629  57.98257  66.29556  73.43757
## 2 80.31360 159.05547 234.27351 307.30282 362.66445 416.12407 467.05523
## 3 11.68175  22.98161  33.60084  43.79316  50.87435  58.13953  64.28329
## 4 14.83004  29.21658  42.78437  55.83943  65.09079  74.45159  82.59185
```

EXAMPLE 2 – AUTO INSURANCE

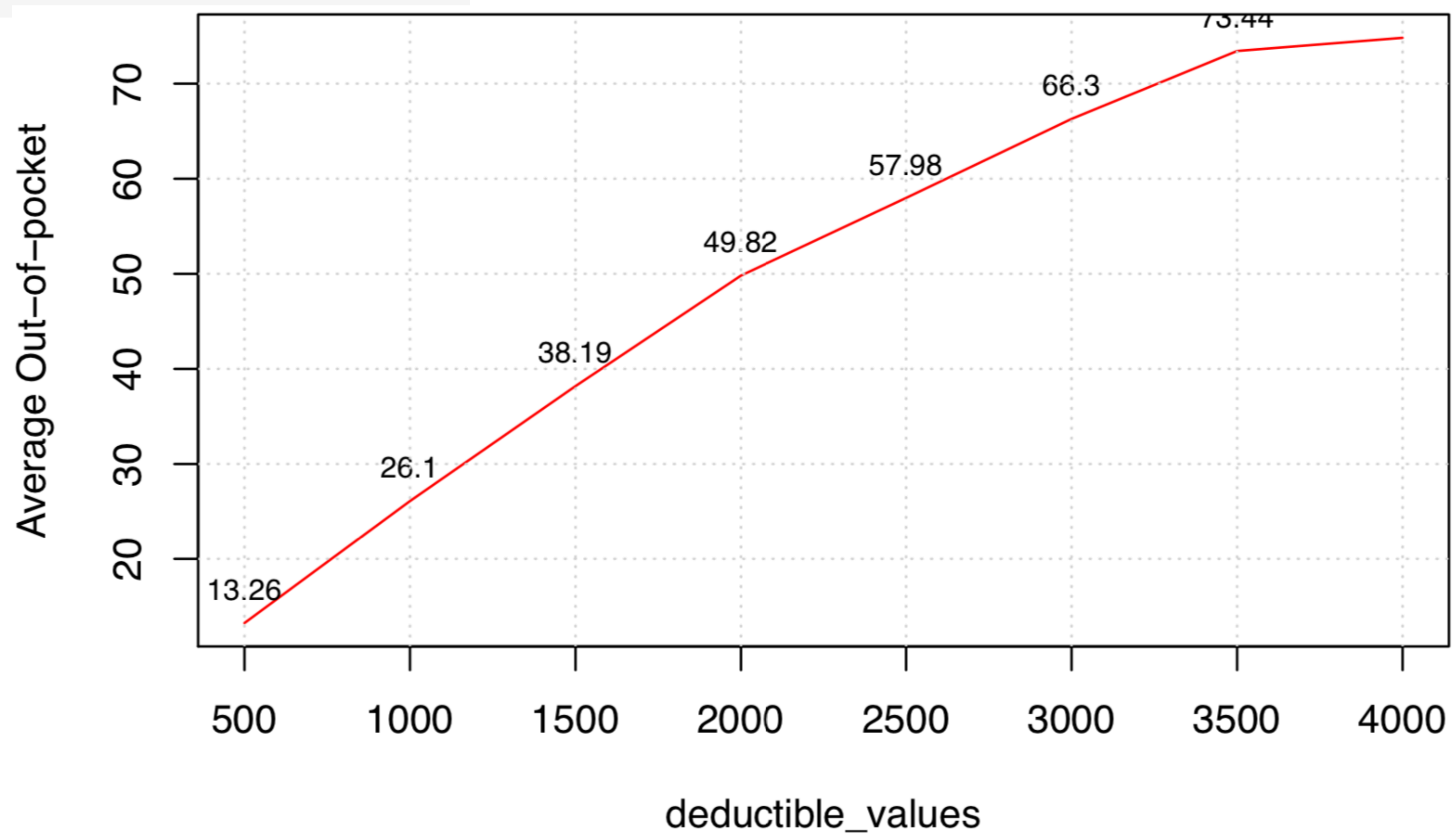
```
rownames(df1) = c("mean","std.deviation","lower_limit","upper_limit")
print(round(df1,2))
```

```
##                  500   1000   1500   2000   2500   3000   3500   4000
## mean           13.26  26.10  38.19  49.82  57.98  66.30  73.44  74.82
## std.deviation  80.31 159.06 234.27 307.30 362.66 416.12 467.06 484.94
## lower_limit    11.68  22.98  33.60  43.79  50.87  58.14  64.28  65.32
## upper_limit    14.83  29.22  42.78  55.84  65.09  74.45  82.59  84.33
```

```
# collect rows
#
avg_out_of_pocket = df1[1,]
lower_limit = df1[3,]
upper_limit = df1[4,]
#
# convert them to vectors
#
avg_out_of_pocket = as.numeric(avg_out_of_pocket)
lower_limit = as.numeric(lower_limit)
upper_limit = as.numeric(upper_limit)
```

EXAMPLE 2 – AUTO INSURANCE

```
# Plot average amount to pay for different deductibles
#
plot(avg_out_of_pocket~deductible_vals,#ylim=c(0,60),
     type="l",col="red",
     xlab="deductible_values",ylab="Average Out-of-pocket")
text(deductible_vals,avg_out_of_pocket,labels=round(avg_out_of_pocket,2),
     col="black",pos=3,off=0.5,cex=0.75)
```



EXAMPLE 2 – AUTO INSURANCE

```
plot(upper_limit~deductible_vals,
      type="l",xlab="deductible_values",ylab="Average Out-of-pocket")
lines(avg_out_of_pocket~deductible_vals,
     type="l",col="red")
     #xlab="deductible_values",ylab="Average Out-of-pocket")
lines(lower_limit~deductible_vals,
     type="l")
text(deductible_vals,avg_out_of_pocket,labels=round(avg_out_of_pocket,2),
     col="black",pos=3,off=0.5,cex=1.1)
```