R Notebook

Code ▾

Run Cmd+Shift+Enter. Insert Chunk Cmd+Option+I. Preview Cmd+Shift+K

1. Probability that a recombination event occurs between two adjacent base pairs is small, on the order on 10^-8…also approximately 3 billion base pairs where recombination can occur

A. Assume there are 3B positions where recombination can occurs and the probability for it is 10^-8… recombination events are independent of each other…total # of recombination events during once instance of meiosis is binomially distributed

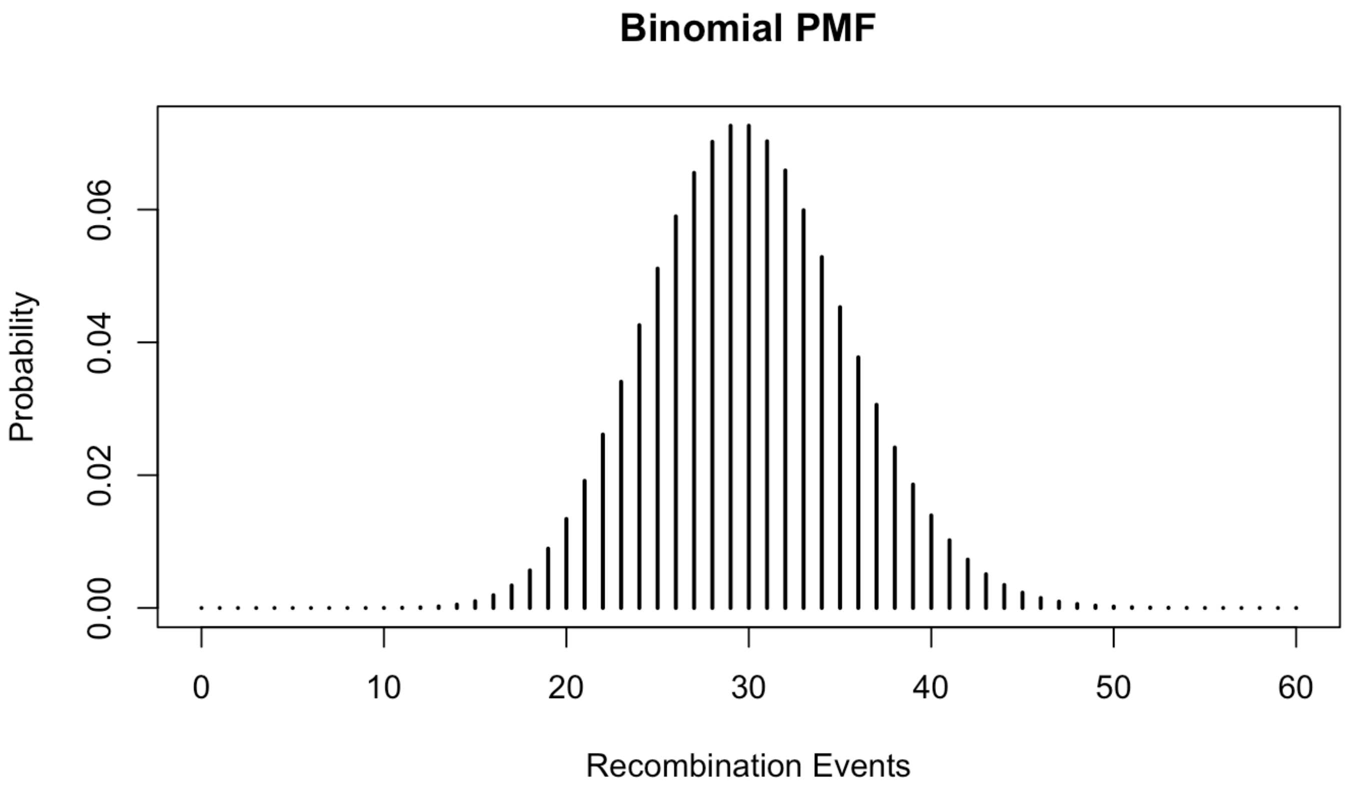
```
i. Use dbinom with parameters size = 3000000000, prob = 0.00000001 and plot the PMF for
the number of recombination rates per meiosis
```

Hide

```
size <- 3000000000
prob <- 0.00000001
lambda <- 30

x <- seq(0, 60, by=1)
y <- dbinom(x, size, prob)

plot(x, y, type="h", lwd=2, main="Binomial PMF", xlab="Recombination Events", ylab="Prob
ability")
```



```
ii. Use pbinom with the same parameters to calculate the probability that the number of recombination events per meiosis is 20 or less
```

Hide

```
pbinom(20, size, prob)
```

```
[1] 0.03528462
```

```
iii. Use pbinom with the same parameters to calculate the probability that the number of recombination events per meiosis is 40 or more
```

Hide

```
1 - pbinom(39, size, prob)
```

```
[1] 0.04625304
```

```
iv. Use qbinom with the same parameters to calculate the 1st percentile
```

Hide

```
qbinom(0.01, size, prob)
```

```
[1] 18
```

```
v. Use qbinom with the same parameters to calculate the 99th percentile
```

Hide

```
qbinom(0.99, size, prob)
```

```
[1] 43
```

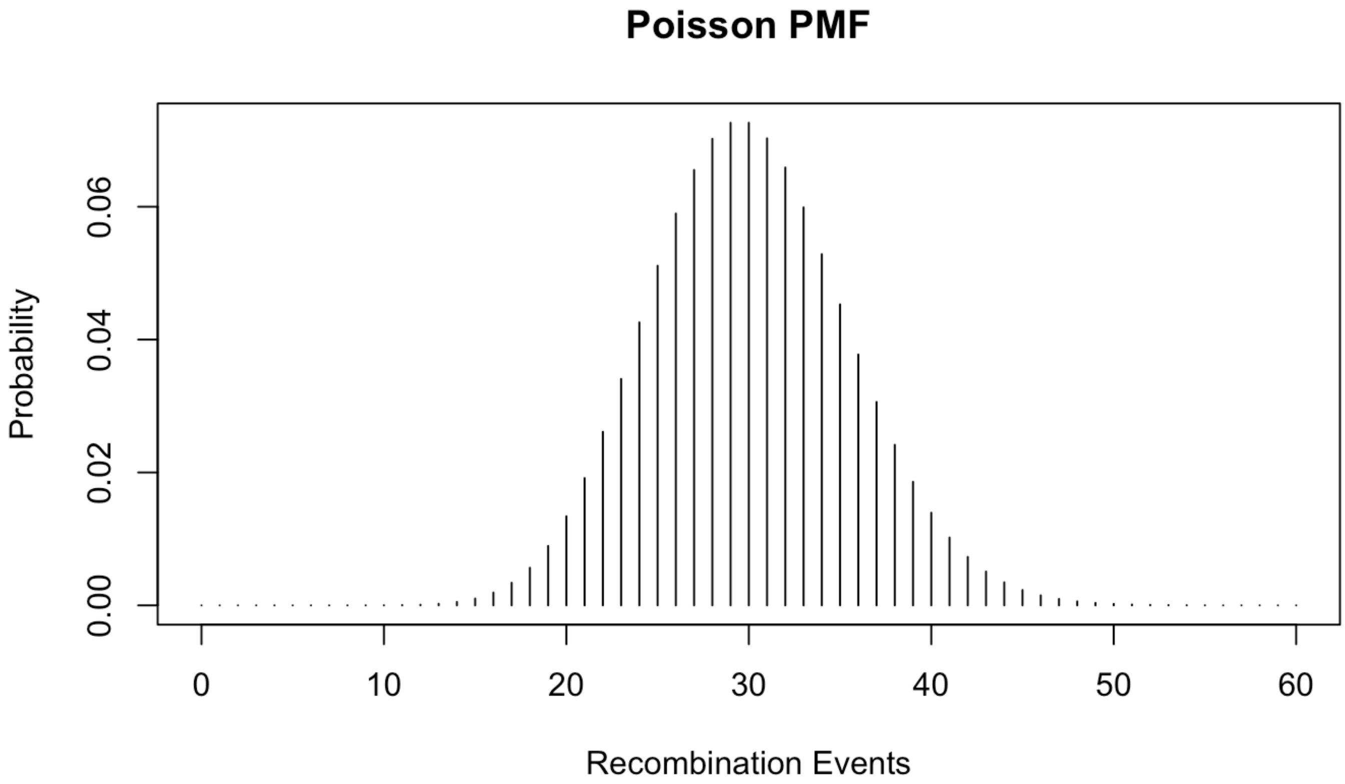
B. Approximate the binomial in part A with the Poisson with parameter λ= 30…30=3 billion x 10^-8

```
i. Use dpois with parameter lamba = 30 and plot the PMF for the number of recombination events per meiosis
```

Hide

```
y <- dpois(x, lambda)

plot(x, y, type="h", main="Poisson PMF", xlab="Recombination Events", ylab="Probability")
```



```
ii. Use ppois with the same parameters to calculate the probability that the number of r
ecombination events per meiosis is 20 or less
```

Hide

```
ppois(20, lambda)
```

```
[1] 0.03528462
```

```
iii. Use ppois with the same parameters to calculate the probability that the number of
recombination events per meiosis is 40 or more
```

Hide

```
1 - ppois(39, lambda)
```

```
[1] 0.04625304
```

```
iv. Use qpois with the same parameters to calculate the 1st percentile
```

Hide

```
qpois(0.01, lambda)
```

```
[1] 18
```

```
v. Use qpois with the same parameters to calculate the 99th percentile
```

Hide

```
qpois(0.99, lambda)
```

```
[1] 43
```

```
vi. Does Poisson appear to be a good approximation to the binomial for this problem?
  Yes, the results are identical so the Poisson approximation is valid
```

2. The exponential density function is (see pdf)…expected value (same as random variable mean) is 1/lambda. Use the dexp function with parameter rate = 1 (corresponds to lambda = 1) to plot the density function. Let n = 100

A. i. Simulate n exponential random variables (use rexp with rate = 1) and take their sample mean…repeat this 1,000 times

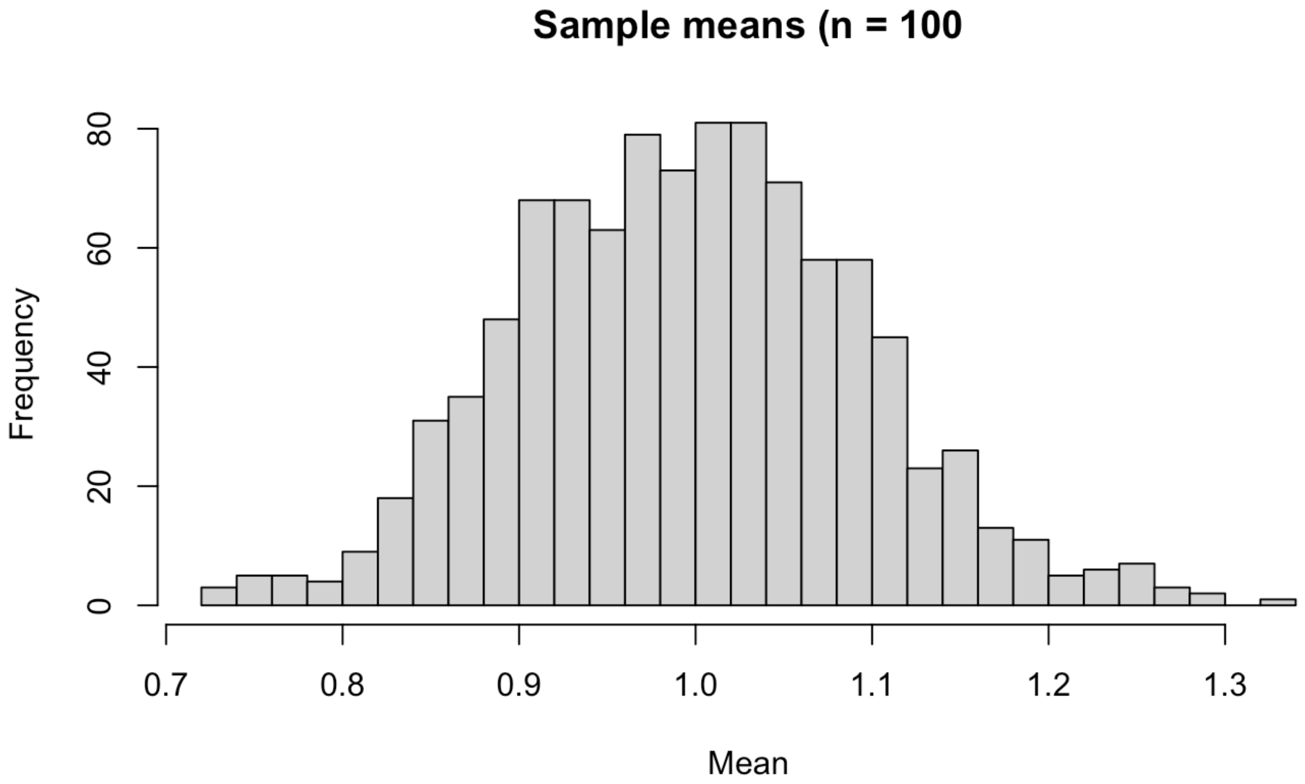
Hide

```
set.seed(123) # reproducibility
n <- 100
sample_means_100 <- replicate(1000, mean(rexp(n, rate = 1)))
```

```
ii. Plot the histogram of the 1,000 sample means from i
```

Hide

```
hist(sample_means_100, breaks = 30, main="Sample means (n = 100", xlab="Mean")
```



```
iii. Calculate the min, mean, an the max of 1,000 sample means from i
```

Hide

```
summary(sample_means_100)
```

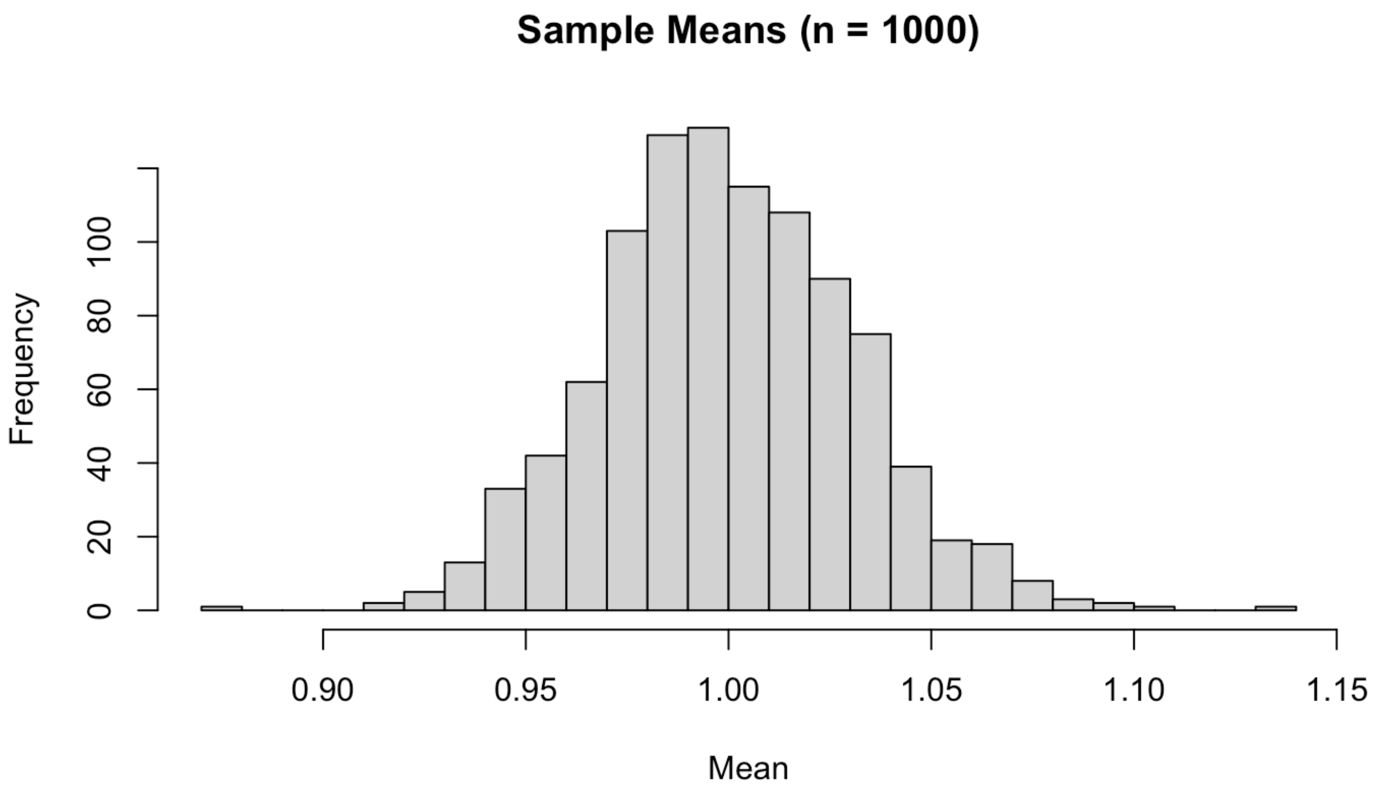
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.7228  0.9271  0.9984  0.9975  1.0617  1.3202 
```

```
iv. Repeat steps i - iii for n = 1000 and n = 10000. Does the sample mean appear to conv
erge?
```

Hide

```
n <- 1000
sample_means_1000 <- replicate(1000, mean(rexp(n, rate = 1)))

hist(sample_means_1000, breaks = 30, main="Sample Means (n = 1000)", xlab="Mean")
```



Hide

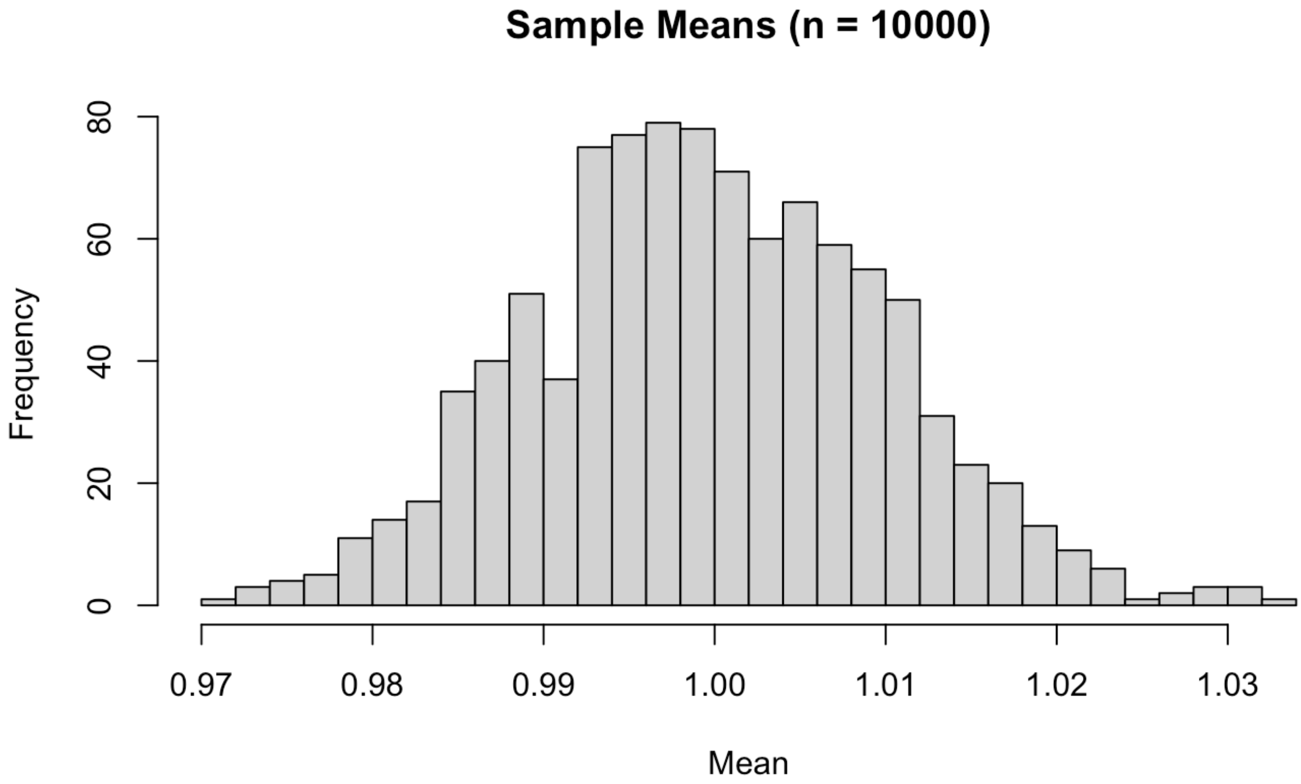
```
summary(sample_means_1000)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.8733  0.9791  0.9980  0.9997  1.0210  1.1317 
```

Hide

```
n <- 10000
sample_means_10000 <- replicate(1000, mean(rexp(n, rate = 1)))

hist(sample_means_10000, breaks = 30, main="Sample Means (n = 10000)", xlab="Mean")
```



Hide

```
summary(sample_means_10000)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.9719  0.9930  0.9993  0.9997  1.0071  1.0335 
```

```
  The sample mean stabilizes around 0.97 - 1.0 as n increases and the spread of the hist
ogram decreases.
```

B. The pareto distribution is (see pdf)…isn’t the standard in R…need to install.packages(“EnvStats)…when a = 1 the expected value is undefined (or infinity)…recall the Law of Large Numbers and the Central Limit Theorem

```
i. Simulate n Pareto random variables (use rpareto with location = 1, shape = 1) and tak
e their sample mean. Repeat this 1000 times
```

Hide

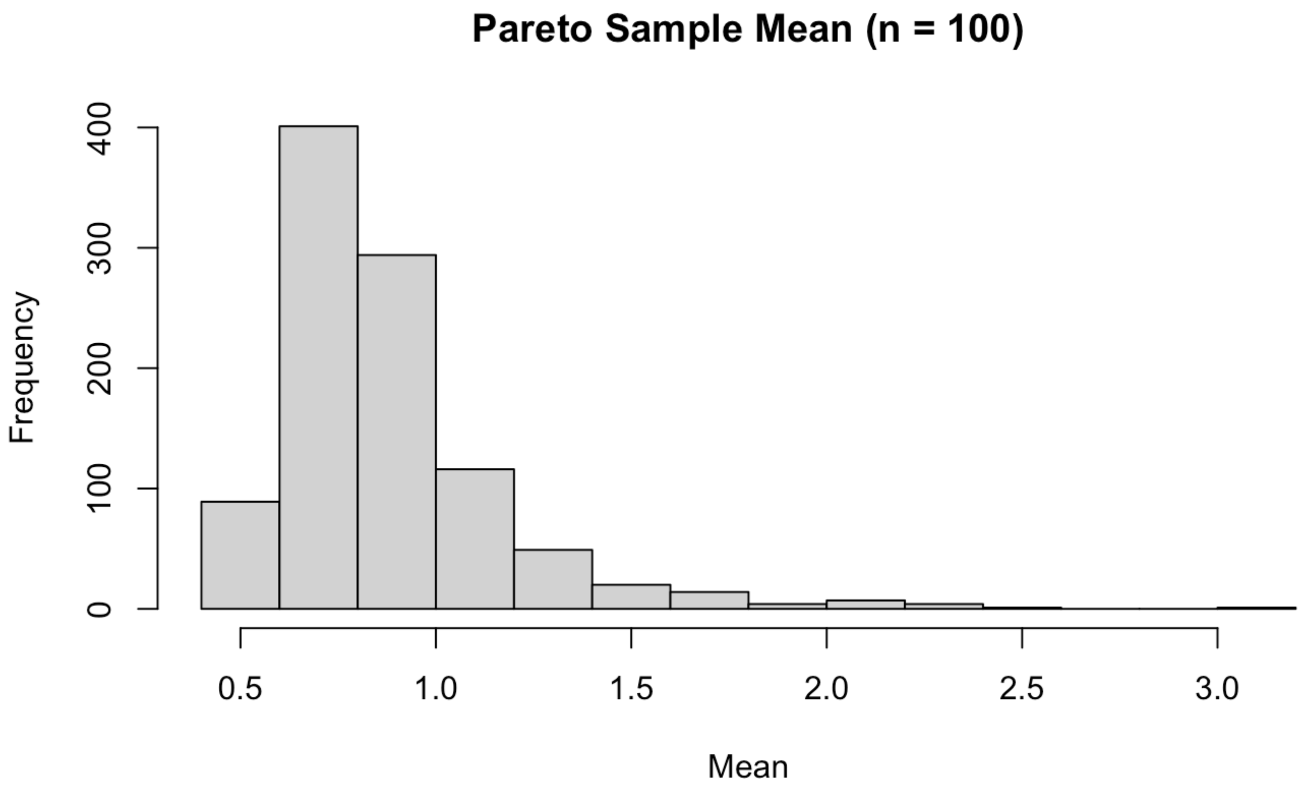
```
#install.packages("EnvStats") # hash after running once
library(EnvStats)

set.seed(123) # for reproducibility
n <- 100
sm_pareto_100 <- replicate(1000, mean(rpareto(n, location = 1, shape = 1)))
```

```
ii. Plot the histogram of the 1000 sample means from i
```

Hide

```
hist(log10(sm_pareto_100), breaks=10, main="Pareto Sample Mean (n = 100)", xlab="Mean")
```



```
iii. Calculate the min, mean, and the max of 1,000 sample means from i
```

Hide

```
summary(sm_pareto_100)
```

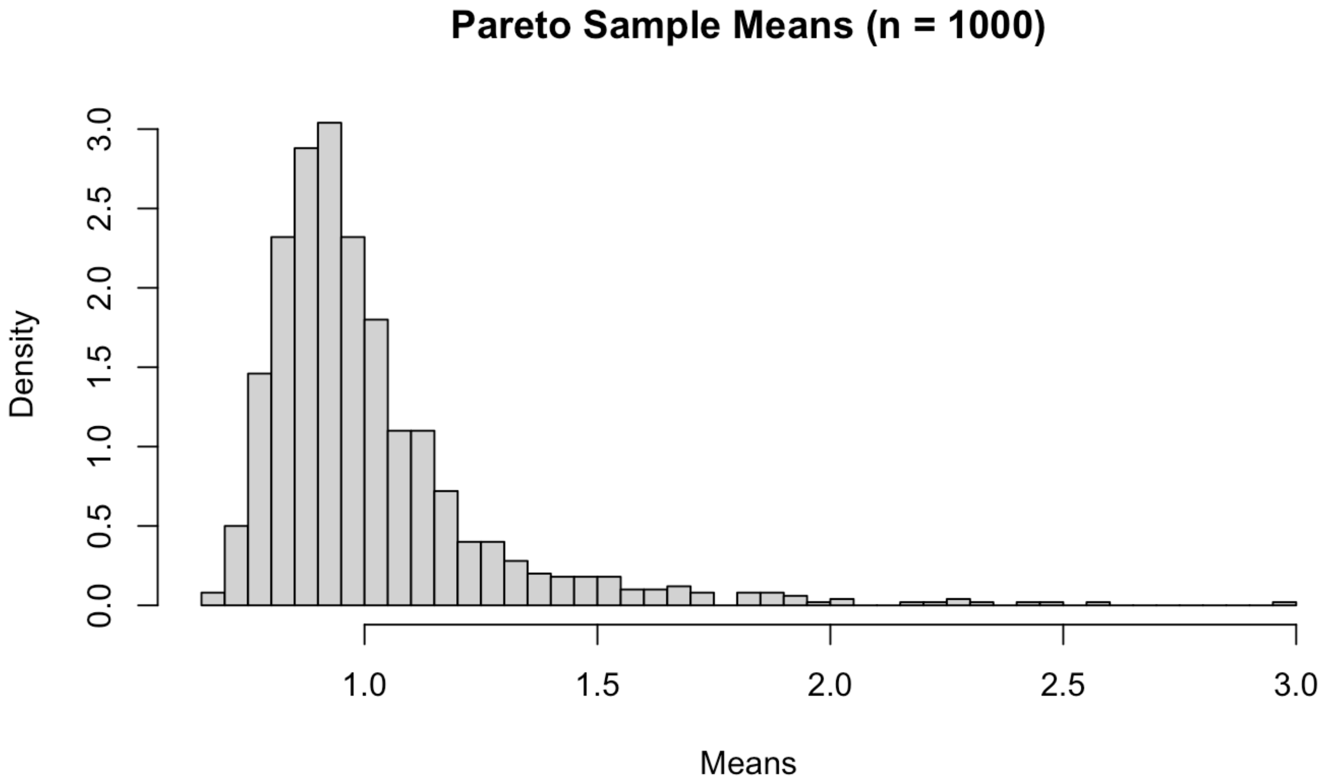
```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
   3.005    4.886    6.391   11.980    9.274 1303.628 
```

```
iv. Repeat steps i - iii for n = 1000 and n = 10000. Does the sample mean appear to conv
erge?
```

Hide

```
n <- 1000
sm_pareto_1000 <- replicate(1000, mean(rpareto(n, location = 1, shape = 1)))

hist(log10(sm_pareto_1000), breaks = 50, main="Pareto Sample Means (n = 1000)", xlab="Me
ans", probability=TRUE)
```



Hide

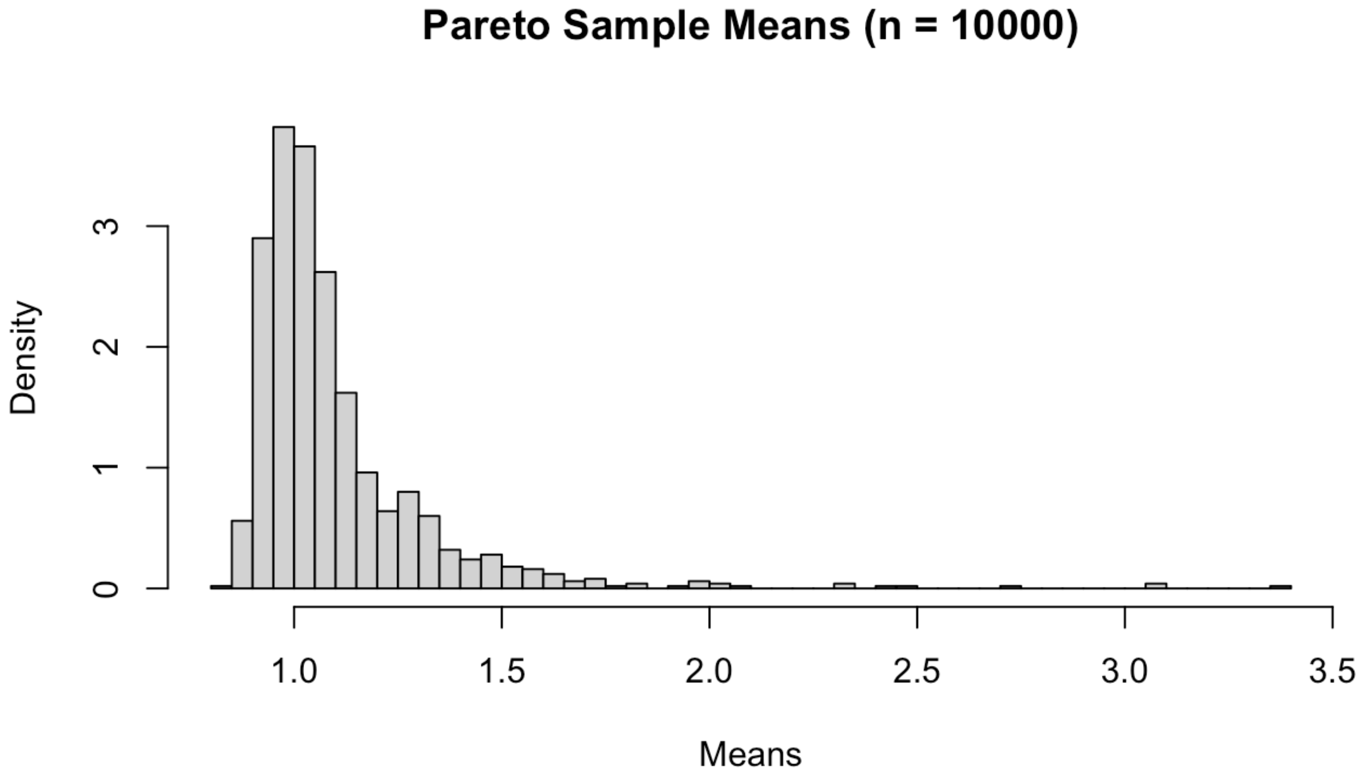
```
summary(sm_pareto_1000)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  4.560   7.263   8.816  14.507  11.947 900.052 
```

Hide

```
n <- 10000
sm_pareto_10000 <- replicate(1000, mean(rpareto(n, location = 1, shape = 1)))

hist(log10(sm_pareto_10000), breaks = 50, main="Pareto Sample Means (n = 10000)", xlab
="Means", probability=TRUE)
```



Hide

```
summary(sm_pareto_10000)
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
   6.885    9.296   10.862   19.833   13.900 2358.245 
```

3. Consider the simple model for father and son heights. Let X be a normal random variable with mean 70 and sdev o, let e1 be a normal random variable with mean 0 and sdev t, and let e2 be a normal random variable with mean 0 and sdev t. Further assume X, e1, and e2 are independent. Define F = X + e1 and S = X + e2… idea is that if F and S are father and son heights, respectively, they both have the same X due to genetics and independent 'disturbance' terms e1 and e2

A. Calculate the covariance of F and S

Hide

```
set.seed(123)
n <- 10000

sigma <- 3 #sdev of genetic component
tau <- 2 #sdev of environmental variation

X <- rnorm(n, mean = 70, sd = sigma) # Genetic
epsilon_F <- rnorm(n, mean = 0, sd = tau) # Random effect father
F <- X + epsilon_F # father's height
epsilon_S <- rnorm(n, mean = 0, sd = tau) # Random effect son
S <- X + epsilon_S # son's height

# Covariance calculation
cov(F, S) # should be close to sigma^2
```

```
[1] 9.119239
```

B. Calculate the correlation of F and S

Hide

```
cor(F, S) # Should be <1 due to regression
```

```
[1] 0.6940373
```

Hide

```
expected_corr <- sigma^2 / (sigma^2 + tau^2)
expected_corr
```

```
[1] 0.6923077
```