INFX 573: Problem Set 5 - Statistical Theory

Charudatta Deshpande

Due: Thursday, November 14, 2017

Problem Set 5

Collaborators: Manjiri Kharkar, Ram Ganesan, Robert Hindhaw, Charles Hemstreet

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio.

1. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
2. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text.
3. Collaboration on problem sets is fun and encouraged, but turn in your individual write-up in your own words. List the names of all collaborators. Do not copy-and-paste from other students' responses or code.
4. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the R Markdown file to `YourLastName_YourFirstName_ps5.Rmd`, knit a PDF and submit the PDF file on Canvas.
5. This problem set involves a lot of experiments with random numbers. Ensure your results can be repeated by selecting a fixed seed

```
set.seed(100) # or pick whatever number you like ;-)
```

1. How often do we get big outliers?

The task in this problem is to conduct a series of MC simulations and see how often do we get outliers of given size. How often do we get "statistically significant" results even if there is nothing significant in our model

1.1 The easy: just normal distribution

Pick your sample size N. 100 or 1000 are good choices.

Now generate a sample of N independent standard normal random variables, and find its mean. It's almost never exactly 0. How big it is in your case? Which values would you consider statistically significant at 95% confidence level?

```
#create a a sample of 1000 independent standard normal random variables.
N <- 1000
x <- rnorm(N)
# m is the mean of the sample.
```

```r
m <- mean(x)
m
```

```
## [1] 0.01680509
```

```r
#cm is the mean of the values falling in 95% confidence interval.
cm <- mean(x < -1.96 | x > 1.96)
cm
```

```
## [1] 0.059
```

```r
# 95% confidence interval values are indicated by this quantile function.
quantile(x, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## -2.042356  2.064136
```

```r
#
```

Answer -
The value of mean is 0.01680509 in this case.

Lower bound of the 95% confidence interval at 2.5% is -2.042356.
Upper bound of the 95% confidence interval at 97.5% is 2.064136.

The values that are between above limits are considered statistically significant at 95% confidence level. It means that if we take samples from this population many times, the mean of the population will be contained within each sample confidence interval 95% of the time.

1.2 Get serious (at least a little).

Select a big R (1000 or more is a good choice) and run the previous experiment R times. Save these results, and based on these calculate the 95% critical quantiles. I.e. compute the values that contain 95% of the means you received in the experiment. (Check out the function `quantile()`). How many means fall out of the theoretical range? Make a histogram of your computed means, and mark the quantiles and the median on it.

Extra challenge: if this seems easy for you, check out *doParallel* package and run it in a `foreach()` loop (package *foreach*) in parallel with `%dopar%`.

```r
#################### METHOD 1 ######################
library(foreach)
set.seed(50)
R <- 1000
# calculate mean of the sample 1000 times
m <- sapply(1:R, function(i) mean(rnorm(100)))
m1 <- mean(m)
m1
```

```
## [1] 7.420987e-05
```

```r
median1 <- median(m)
median1
```

```
## [1] -0.0009874921
```

```r
sd1 <- sd(m)
sd1
```

```
## [1] 0.1003288
```

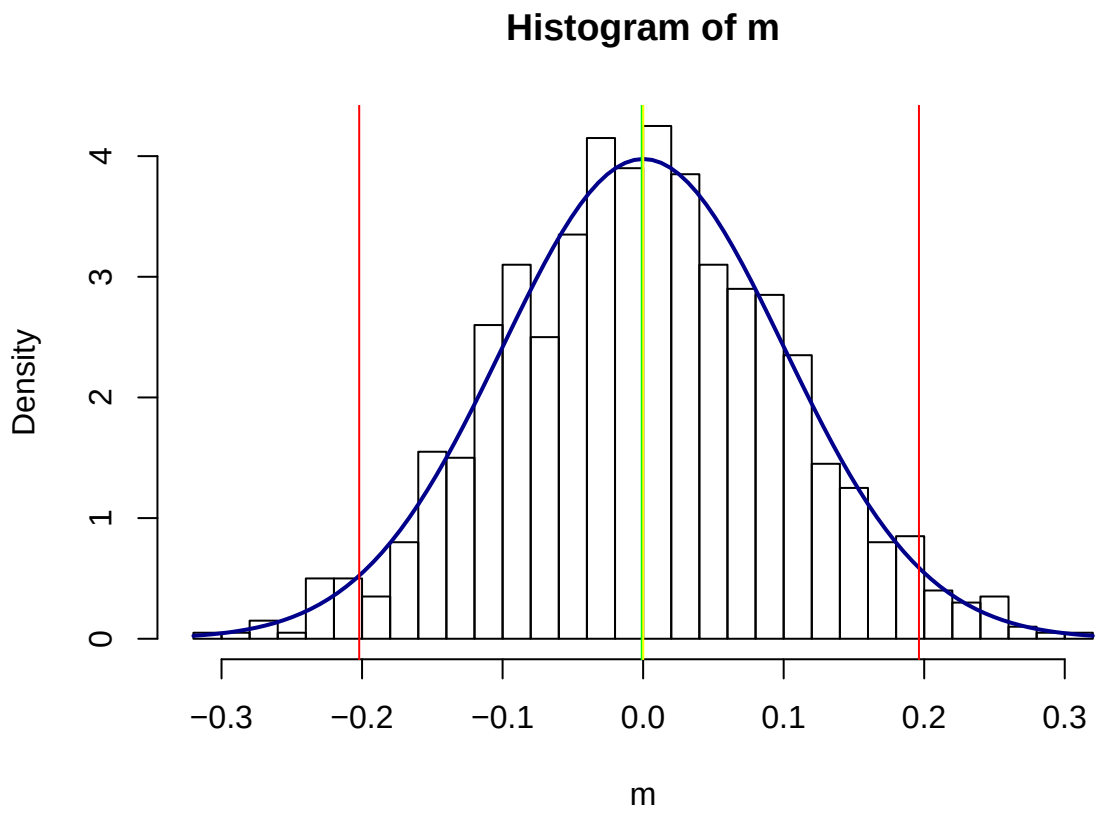
```r
q1 <- quantile(m, c(0.025, 0.975))
q1
```

```
##       2.5%      97.5%
## -0.2020255  0.1962620
```

```r
#Calculate number of means falling outside the confidence interval.
sum(m < -0.2020255 | m > 0.1962620)
```

```
## [1] 50
```

```r
# create a histogram for m
hist(m,breaks=sqrt(R), prob=T)
# apply the best fitting distribution curve to the histogram
curve(dnorm(x, mean=m1, sd=sd1), col="darkblue", lwd=2, add=TRUE)
# place the quantile lines on the histogram
abline(v=quantile(m, c(0.025, 0.975)), col="red")
# place the median on the histogram
abline(v=median1, col="green")
# place the mean on the histogram
abline(v=m1, col="yellow")
```




```r
#
###################   METHOD 2 with %dopar% and foreach()   #############
set.seed(50)
library(doParallel)
```

```
registerDoParallel()
m <- foreach(i=1:R, .combine=c) %dopar% {
mean(rnorm(100))
}
#This would give you the same 'm' as in method 1, may be with slightly different values.
#Then we can follow the same process for histogram and other analysis.
```

Answer -

a. Extra challenge question - Done.

b. 95% CI values are -0.2020255 and 0.1962620.

c. Number of means falling outside of 95% CI values is 50, or exactly 5% of the sample size in this case. However it is not always so exact, but the expected value is always very close to 5%.

d. Histogram of your computed means, and mark the quantiles and the median - Done.
The yellow mean and green median are almost identical values, but can be identified as separate line if zoomed sufficiently.

1.3 Clustered data: get even more serious

So far we looked at samples that contained homogeneous identical members. Everything was sampled from $N(0,1)$. Now let's introduce some heterogeneity (clusters) into the sample. Imagine we are analyzing students from different schools. First we (randomly) pick a number of schools, and thereafter we randomly pick a number of students from each of these schools.

Your Data Generating Process (DGP) shold look as follows:

1. pick number of clusters C (10 is a good choice)
2. create cluster centers μ_c for each cluster $c \in \{1, ..., C\}$ by sampling from $N(0,1)$.
3. create N cluster members for each cluster. The value for cluster member should be shifted by the cluster center: $x_{ci} = \mu_c + \epsilon_i$ where $\epsilon \sim N(0,1)$.
4. compute the total mean of all members m.

Repeat the process 2-4 R times (you may pick another R if you wish).

Answer the similar questions as above:

1. does the distribution of m look normal? You may want to use `qqnorm()` function to show it.
2. what is the 95% confidence interval of the distribution?
3. what were the 95% theoretical confidence intervals in case of no clustering (or alternatively, if $c_i = 0 \; \forall i$)?
4. Why is your confidence interval in case of clustering so much larger than for no clustering?
5. in the simulation: why should you re-generate the cluster centers c? What would happen if you just repeat the steps 3 and 4? Try it out if you cannot find the theoretical explanation!

```
#create 10 cluster centers, create 10 clusters around the centers
# and repeat the process 5000 times.
R <- 5000
C <- 10
y <- list()
u <- list()
m <- vector()
for (a in 1:R) {
c <- rnorm(C)
for (i in 1:C) {
  y[[i]] <- rnorm(100) + c[i]
```

```
}
u[[a]] <- unlist(y)
m[a] <- mean(u[[a]])
}
#The resulting vector 'm' has 5000 unlisted means from the above
#experiment. We will use this to answer the questions.
m2 <- mean(m)
m2
```

```
## [1] 0.001595317
```

```
median2 <- median(m)
median2
```

```
## [1] -0.003546918
```

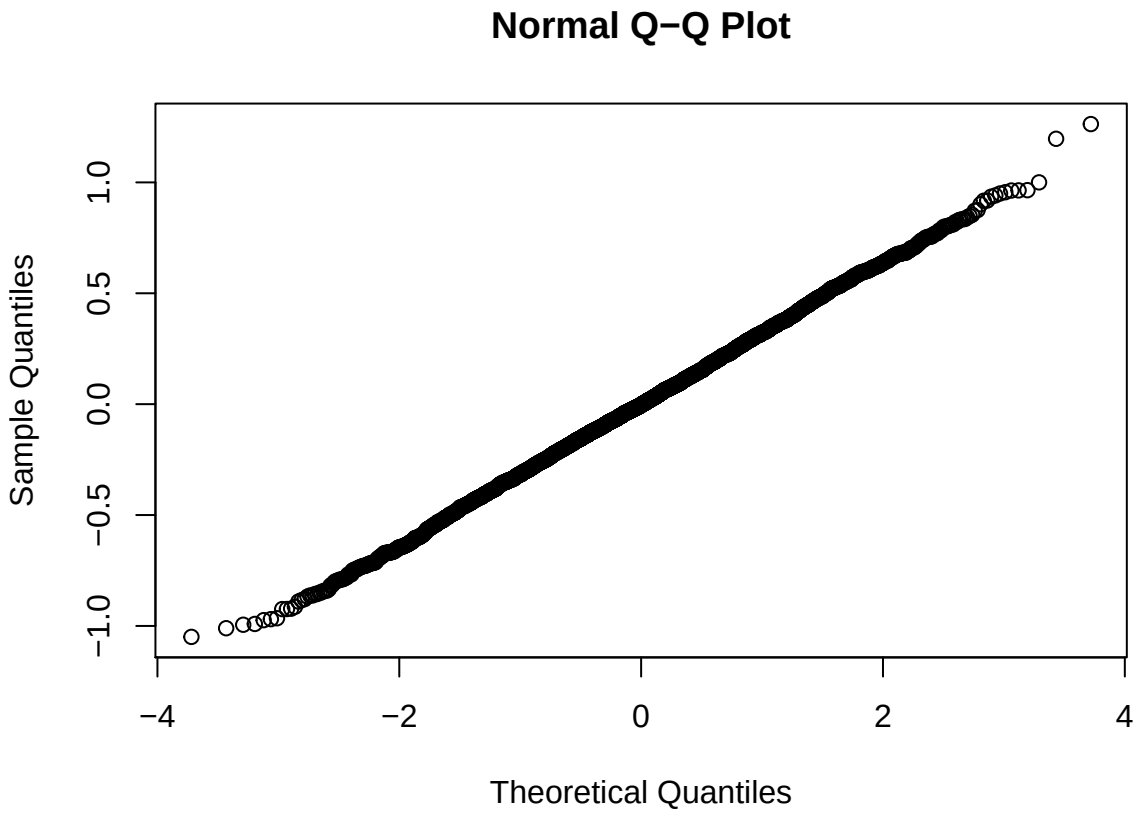
```
sd2 <- sd(m)
sd2
```

```
## [1] 0.3186894
```

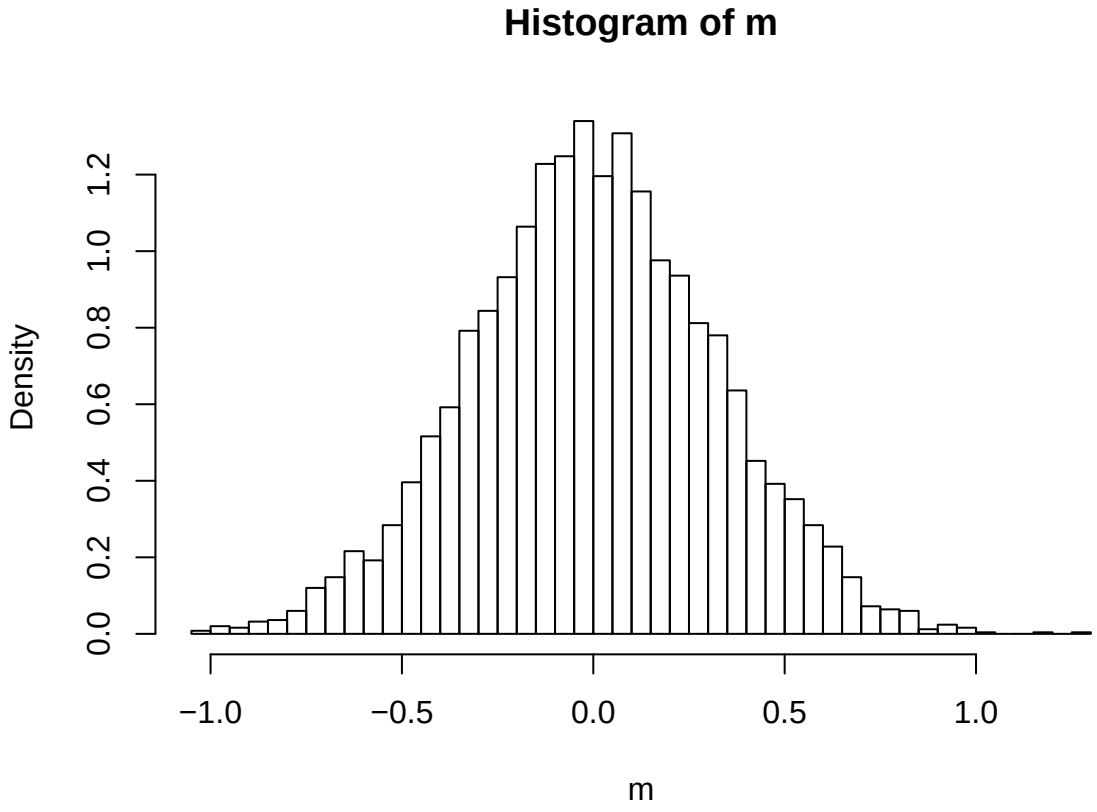
```
q2 <- quantile(m, c(0.025, 0.975))
q2
```

```
##       2.5%      97.5%
## -0.6389836  0.6252287
```

```
qqnorm(m)
```



```
hist(m,breaks=sqrt(R), prob=T)
```




```
#Calculate m without clustering
m <- sapply(1:R, function(i) mean(rnorm(100)))
q3 <- quantile(m, c(0.025, 0.975))
q3
```

```
##       2.5%      97.5%
## -0.1917157  0.1955008
```

Answers -

1. The distribution of m is normal around zero. This indicates that means of clusters created using normally distributed centers with uniform cluster sizes are normally distributed too.

2. 95% CI values are stored in q2 (these will keep changing since we do not have a seed set.). For exact values in this PDF refer to printed values of q2 above.

3. 95% CI values without clustering are stored in q3 (these will keep changing since we do not have a seed set.). For exact values in this PDF refer to printed values of q3 above.

4. Clustering 10 times and repeating the process 5000 times means that the measures will be spread across a larger distribution than a single sample created 5000 times. That is why the CI size with clustering is much larger than without clustering.

5. Cluster centers need to be regenerated to be able to truly create 10 random clusters 5000 times. If centers are not regenerated, we will end up adding more values to same 10 clusters 5000 times. So instead of getting 10 random clusters 5000 times, we will get 10 very large clusters. That is not the

purpose of our analysis. We need to create different clusters each time to truly analyze effects of clustering on mean distribution.

1.4 It gets worse: unequal cluster size

Earlier our clusters were of similar size. However, there are many distributions that are highly inequal.

1. Before reading any further, what do you think, how does distribution of researchers' influence (say, number of citations) look like? What might be it's mean?

Answer to above question -

The distribution of researchers' influence will be highly unequal, comparable to clusters of unequal sizes. More famous researchers, or those working on popular topics will get more citations, and will have more general influence. Those working on less popular topics, or those less famous researchers will have less influence. To analyze such phenomenon, we would need a statistical method that would account for such unequal distribution.

Pareto distribution is a popular distribution to describe such highly unequal distributions, such as sizes of cities, forest fires, internet traffic through servers, income, influence of humans, etc. Analyze Pareto distribution:

1. what is the analytic expression for it's pdf? Explain the parameters.
2. make a graph of it's pdf using log-log scale.
3. what is it's expected value? What are the conditions?

Answers to above questions -

1. The expression of Pareto Distribution PDF is as given below -

$f(x) = ab^a/x^{a+1}$, x >= b

a is the 'shape parameter' while b is 'scale parameter'. A shape parameter, as the name suggests, affects the general shape of the Pareto distribution. Scale parameter is a measure of variance of the distribution. It usually stretches or squeezes a graph depending on the variance.

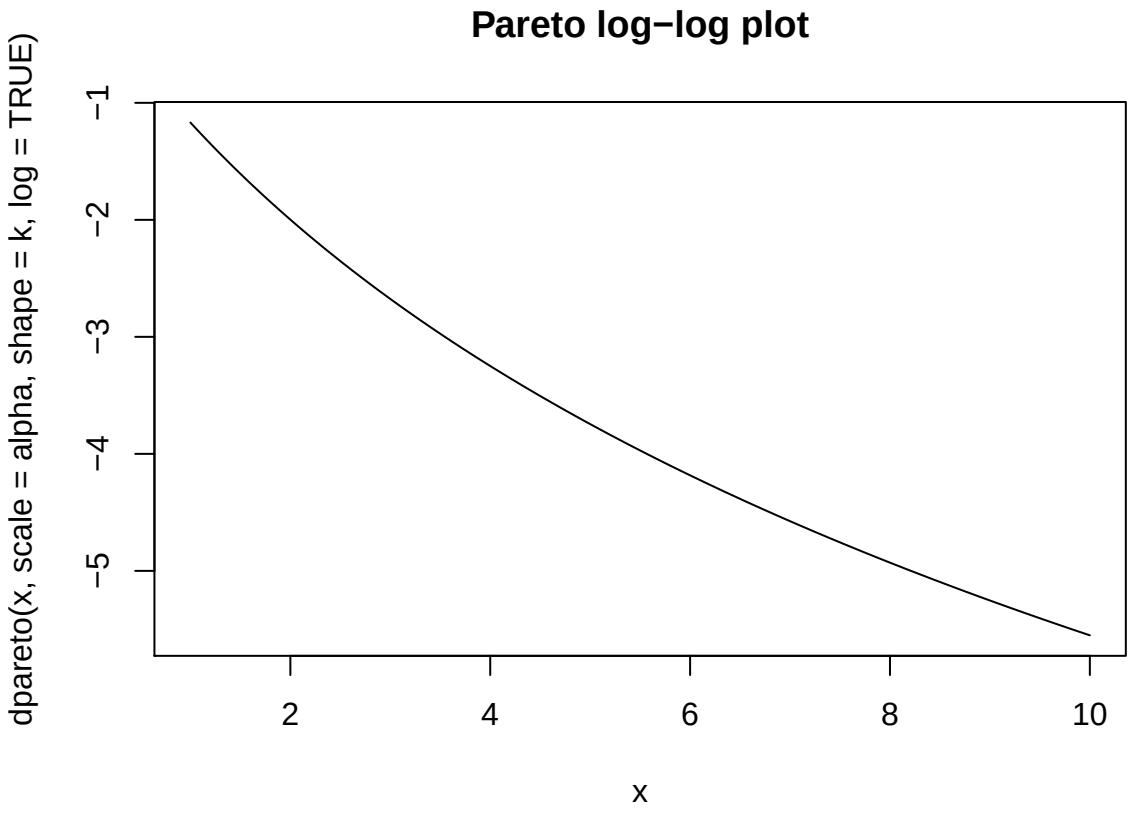
2. Following code will create a log-log plot for Pareto distribution.

```
library(actuar)
```

```
##
## Attaching package: 'actuar'
```

```
## The following object is masked from 'package:grDevices':
##
##     cm
```

```
alpha <- 3; k <- exp(1); x <- seq(1, 10, len = 300)
plot(x, dpareto(x, scale = alpha, shape = k, log=TRUE), type = "l",
     main = "Pareto log-log plot")
```



3. The expected value of a measure in the distribution is -

E(X) = ab/(b-1) for b > 1. If b is 1, the expected value is infinite.

Now your task is to conduct a similar experinent using unequally sized clusters.

Your Data Generating Process (DGP) shold look as follows:

1. pick number of clusters C (10 is a good choice)
2. create cluster sizes N_c using Pareto distribution. Pick a highly unequal version using the shape parameter ≤ 1. You can set the minimum size to 1.
3. create cluster centers μ_c for each cluster $c \in \{1, ..., C\}$ by sampling from $N(0, 1)$.
4. create N_c cluster members for each cluster c. The value for cluster member should be shifted by the cluster center: $x_{ci} = \mu_c + \epsilon_i$ where $\epsilon \sim N(0, 1)$.
5. compute the total mean of all members m.
6. compute the total number of observations $N = \sum_c N_c$.

Repeat the steps 2-6 R times.

```
R <- 5000
C <- 10
y <- list()
u <- list()
m <- vector()
scale <- 1
shape <- 0.77656  # Randomly chosen shape parameter
lower_bound <- 1 # lower bound of cluster size, set to 1 as suggested in assignment
upper_bound <- 1000 # upper bound of cluster size
```

```
#
for (a in 1:R) {
#use pareto distribution to calculate quantiles to be used for creation of cluster sizes
#Repeat this process 5000 times
quantiles <- ppareto(c(lower_bound, upper_bound), scale, shape)
uniform_random_numbers <- runif(C, quantiles[1], quantiles[2])
#unequal cluster sizes for 10 clusters
cluster_sizes <- qpareto(uniform_random_numbers, scale, shape)
#create cluster centers 5000 times
c <- runif(C, 0, 1)
for (i in 1:C) {
  y[[i]] <- runif(cluster_sizes[i], 0, 1) + c[i]
}
u[[a]] <- unlist(y)
m[a] <- mean(u[[a]])
}
# Calculate total number of observations
z <- sum(sapply(u,length))
z
```

```
## [1] 486759
```

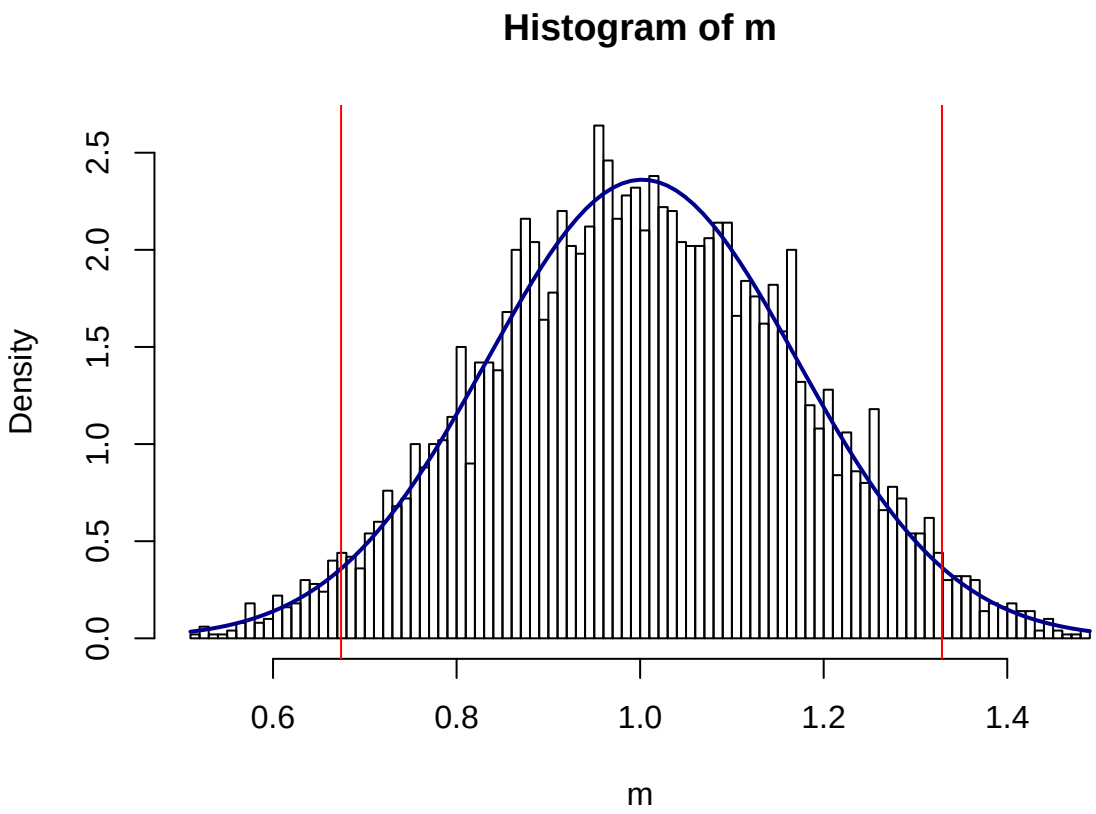
```
# Calculate mean of all means in 'm' vector
grand.mean <- mean(m)
grand.mean
```

```
## [1] 1.002232
```

```
#calculate 95% CI
q4 <- quantile(m, c(0.025, 0.975))
q4
```

```
##      2.5%     97.5%
## 0.6741549 1.3288884
```

```
#histogram of m with best fitting curve and quantile lines
hist(m,breaks=sqrt(R), prob=T)
curve(dnorm(x, mean=mean(m), sd=sd(m)), col="darkblue", lwd=2, add=TRUE)
abline(v=quantile(m, c(0.025, 0.975)), col="red")
```



Answer the similar questions as above:

1. does the distribution of m look normal?
2. what is the 95% confidence interval of this distribution?
3. compare the outcome with the one above and explain the differences.

Answers -
Total number of observations - refer to printed value of z above. The number is above 400,000 observations.

Mean of all means - refer to printed value of grand.mean above. It is very close to 1.

1. The distribution of 'm' is not normal around 0. It is centered around very close to 1. However around 1, the distribution of m can be considered normal.

2. 95% CI - refer to printed values of q4 above.

3. When cluster sizes are uniform and cluster centers are normally distributed, the distribution of cluster means is normal around zero. The mean of cluster members is very close to zero.
   However when clusters of unequal sizes are created with Pareto distribution, the distibution of cluster means is not normal around zero. It is normalized around very close to 1 in this experiment. However it could be normal about some other number depending on shape and scale chosen for Pareto distribution.

Note - this distribution is normal about 1 because cluster centers were chosen with runif. If cluster centers were chosen with Pareto distribution, the distribution of m would be highly unpredictable.

2. Find the right distribution

Off-trail running, such as orienteering involves crossing uneven terrain at speed. An experienced runner falls approximately once during an one-hour race in average.

1. What is an appropriate probability distribution for analyzing the number of falls?

Answer -

The appropriate probability distribution for analyzing the number of falls is **Poisson Distribution**. It describes the number of successes in a series of independent Yes/No experiments with different success probabilities. This applies to our qyestion at hand.

2. What is the expected value and variance of the number of time the athlete falls?

Answer -

The expected value of number of time an athlete falls = **1**.

Mean and variance are indicated by same variable $\lambda = \mathbf{1}$.

3. Would it be exceptional if the runner falls 4 times?

Answer -

The word 'exceptional' is not categorically defined, hence only an estimate can be made if a probability is indeed exceptional. The probability that a runner falls 4 times is calculated by following equation -

```
dpois(4, 1)
```

```
## [1] 0.01532831
```

The probability that a runner will fall 4 times is 0.01532831, which can be considered exceptional.

4. What is the probability that the runner will fall no more than twice during a given (1hr) race.

Answer -

The probability that the runner will fall no more than twice is calculated by following code.

```
ppois(2, 1)
```

```
## [1] 0.9196986
```

The probability is 0.9196986.

3. Overbooking Flights

You are hired by *Air Nowhere* to recommend the optimal overbooking rate.

The airline uses a 200-seat plane and tickets cost $200. So a fully booked plane generates $40,000 revenue. The sales team found that the probability that passengers who have paid their fare actually show up is 99%, and individual show-ups can be considered independent. The additional costs, associated with finding an alternative solutions for passengers who are refused boarding are $1000 per person.

1. Which distribution would you use to describe the actual number of show-ups for the flight?

Answer -

The correct distribution here is **Binomial Distribution**. A passanger showing up for the flight, or not showing up for the flight are mutually exclusive events. They have same probabilities for success/failure for all events, and these type of experiments are analyzed using Binomial Distribution.

2. Assume the airline never overbooks. What is it's expected revenue?

Answer -

Assuming all 200 seats are sold, the expected revenue is $40,000. Otherwise it is simply number of tickets sold * $200.

3. Now assume the airline sells 201 tickets for 200 seats. What is the probability that all 201 passengers will show up?

Answer -

```
dbinom(201, 201, 0.99)
```

```
## [1] 0.1326399
```

The probability that all 201 passangers will show up for flight is 0.1326399.

4. What are the expected profits (= revenue − expected losses) in this case? Would you recommend overbooking over booking the just right amount?

Answer -

Expected profits if all 201 passangers show up = (200*201) - $1000 = $39,200. This is less profit than not overbooking, however, the probability of this happening is only about 13%. There is 87% chance that 201 passangers will not show up, so overbooking by 201 tickets is recommended.

5. Now assume the airline sells 202 tickets. What is the probability that all 202 passengers show up?

Answer -

```
dbinom(202, 202, 0.99)
```

```
## [1] 0.1313135
```

The probability that all 202 passangers will show up for flight is 0.1313135.

6. What is the probability that 201 passengers – still one too many – will show up?

Answer -

```
dbinom(201, 202, 0.99)
```

```
## [1] 0.2679326
```

The probability that 201 out of 202 passangers will show up for flight is 0.2679326.

The probability that 201 or 202 out of 202 passangers will show up for flight is dbinom(201, 202, 0.99) + dbinom(202, 202, 0.99) which is = 0.399246.

7. Would it be advisable to sell 202 tickets?

Answer -

The probability that 201 or 202 out of 202 will show up is 0.399246. This means there is 0.600754 probability that airline can sell 202 seats and still only 200 people show up.

But the important factor to consider here is the rebooking cost. Even if airline profits 60% of the time, it will end up paying rebooking fees 40% of the time for 1 or 2 passengers. Since rebookign fees are 5 times that of the profit from a single seat, in the long run airline is going to lose revenue by selling 202 tickets.

For these reasons I do NOT recommend selling 202 tickets.

8. What is the optimal number of seats to sell for the airline? How big are the expected profits?

Answer -

As determined above, selling 202 seats is not advisable. The maximum number of tickets that the airline can sell without loss is 201. When 201 seats are booked, 87% of the time airline is going to profit. IIn those 87% of times, the airline will get 200 * 201 = $40,200 revenue.

In 13% of the time, the airline will need to rebook one customer. So the expected revenue in these 13% of times is 40,200 - 1000 = $39,200.

Hint: some of the expressions may be hard to write analytically. You may use R functions to do the actual calculations instead, but then explain in the text how do you proceed.