INFX 573: Problem Set 5 - Statistical Theory

Rajendran Seetharaman

Due: Thursday, November 9, 2017

Problem Set 5

Collaborators:

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(1) # 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.

```
N <- 1000
```

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?

Ans. The sample mean is -0.00609382446085629. The values in the sample greater than 1.959964 or less than -1.959964 are statistically significant.

The statistically significant values are listed below.

```r
#generate sample
randnorm <- rnorm(N,mean=0,sd=1)
#compute sample mean
print(paste("Mean:",mean(randnorm)))
```

```
## [1] "Mean: -0.0116481419383404"
```

```r
#compute 95% confidence interval
upper <- qnorm(0.975)
lower <- qnorm(0.025)
count <-0
print("Statistically Significant values:")
```

```
## [1] "Statistically Significant values:"
```

```r
#check which values are outside the confidence interval
for(val in randnorm){
  if(val>upper || val<lower)
{
  print(val)
  count <- count+1
  }
}
```

```
## [1] -2.2147
## [1] -1.989352
## [1] 1.9804
## [1] 2.401618
## [1] 2.172612
## [1] 2.087167
## [1] 2.206102
## [1] 2.307978
## [1] 2.075245
## [1] -2.285236
## [1] 2.497662
## [1] -2.888921
## [1] -2.000165
## [1] -2.403096
## [1] 2.649167
## [1] -2.289124
## [1] 1.971337
## [1] -2.264889
## [1] -2.592328
## [1] -2.129361
## [1] -2.033286
## [1] -3.008049
## [1] -2.342723
## [1] -2.097883
## [1] 2.165369
## [1] 2.350554
## [1] 2.446531
## [1] -1.972935
## [1] 2.284659
## [1] 3.810277
## [1] -2.528501
```

```
## [1] -2.201782
## [1] -2.331712
## [1] 2.024842
## [1] -1.962353
## [1] -2.114335
## [1] 2.001719
## [1] 2.675741
## [1] -2.43264
## [1] -2.939774
## [1] 2.189752
## [1] -1.967195
## [1] -2.442311
## [1] 1.971572
## [1] -2.41245
## [1] -2.245153
## [1] -2.035004
## [1] 2.021347
## [1] -2.596111
## [1] -2.424317
## [1] 1.979633
## [1] 2.349493
## [1] 2.236323
## [1] 2.005719
## [1] -2.070571
## [1] 3.055742
## [1] -2.371023
## [1] -2.08834
## [1] 2.321334
## [1] -2.402231
## [1] 2.169116
## [1] 2.251883
## [1] -2.239783
## [1] -2.514425
## [1] 2.210952
## [1] -2.189876
## [1] -2.996949
## [1] 2.027056
```

```
print("Significant Values count:")
```

```
## [1] "Significant Values count:"
```

```
print(count)
```

```
## [1] 68
```

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.

Ans. In my case, 50 means fall out of the theoretical 95% quartiles range.

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%`.

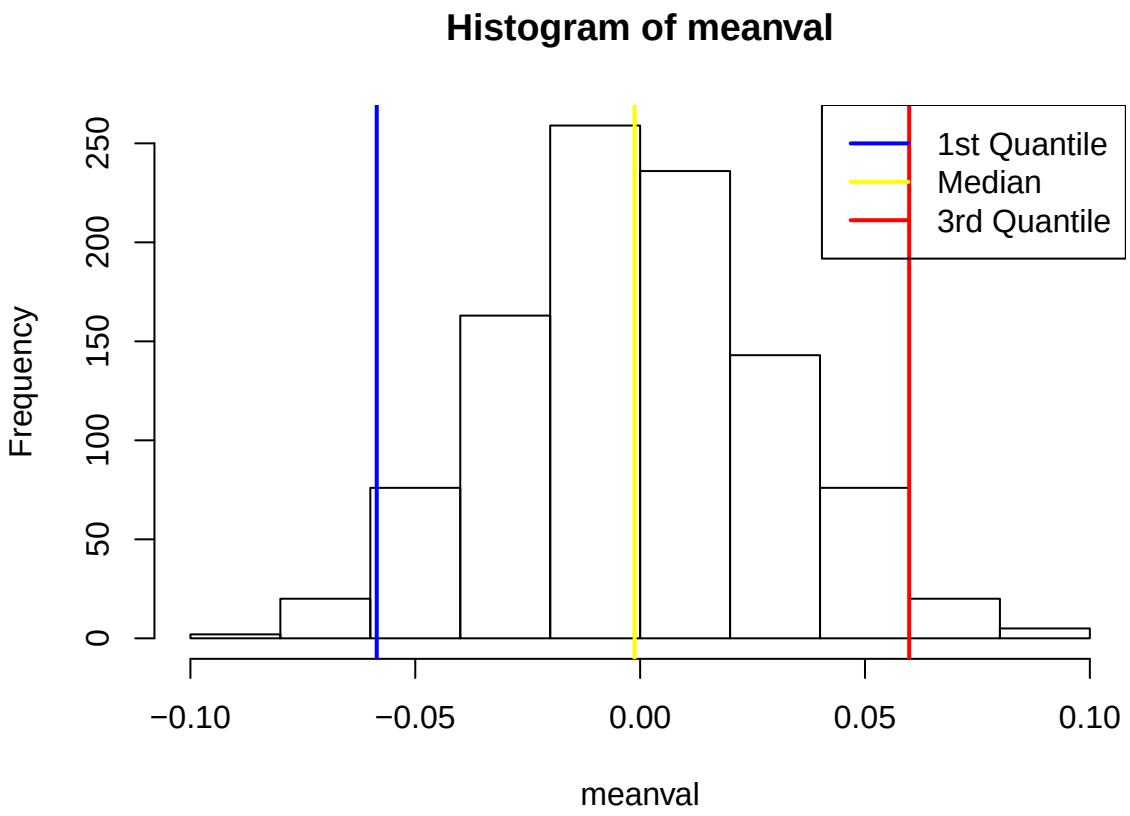
```
registerDoParallel(cores=20)
#set number of experiments
R <- 1000
#repeat sampling R times and compute mean of each sample
meanval <- foreach(i <- 1:R, .combine=append)  %dopar% {mean(rnorm(N,mean=0,sd=1))}
#compute the 2.5%, 50%, and 97.5% quantiles for meanval
quants <- quantile(meanval,probs=c(0.025,0.50,0.975))
quants
```

```
##         2.5%          50%        97.5%
## -0.058569375 -0.001222561  0.059830411
```

```
mean_outliers <-0
#count the ouliers
for(x in meanval)
{
  if(x<quants[1] || x>quants[3])
  {
    mean_outliers<- mean_outliers+1
  }
}
print(paste("Count of mean outliers:",mean_outliers))
```

```
## [1] "Count of mean outliers: 50"
```

```
#plot the histogram of computed means
plot.new()
hist(meanval)
abline(v=quants[1],col="blue",lwd=2)
abline(v=quants[2],col="yellow",lwd=2)
abline(v=quants[3],col="red",lwd=2)
legend(x = "topright",
 c("1st Quantile", "Median", "3rd Quantile"),
 col = c("blue", "yellow", "red"),
 lwd = c(2, 2, 2))
```



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.

Ans. The distribution looks normal as seen in the qqnorm plot.

2. what is the 95% confidence interval of the distribution?

Ans. the 95% confidence interval of the distribution is (-0.6257261,0.6243399).

3. what were the 95% theoretical confidence intervals in case of no clustering (or alternatively, if $c_i = 0 \; \forall i$)?

Ans. The 95% confidence intervals in the case of no clustering, as seen from the calculation below are-(-0.01917353,0.0205135)

4. Why is your confidence interval in case of clustering so much larger than for no clustering?

Ans. In case of clustering, each of the cluster centers are generated randomly from N(0,1). The cluster members in each of these clusters are distributed around those centers. Due to this, there is a high probability of cluster members being highly spread out and far away from the mean of N(0,1) which is 0. This results in a higher 95% confidence interval as a greater number of points would be far away from 0. In case of no clustering, ie. i.e 0 center clusters, all points would be normally distributed with center 0, resulting in lesser points being far away from 0, have a mean close to 0, and a much smaller 95% confidence interval.

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!

Ans. It is necessary to regenerate the cluster centers as having the same cluster centers for each cluster would reduce the randomness of the experiment.

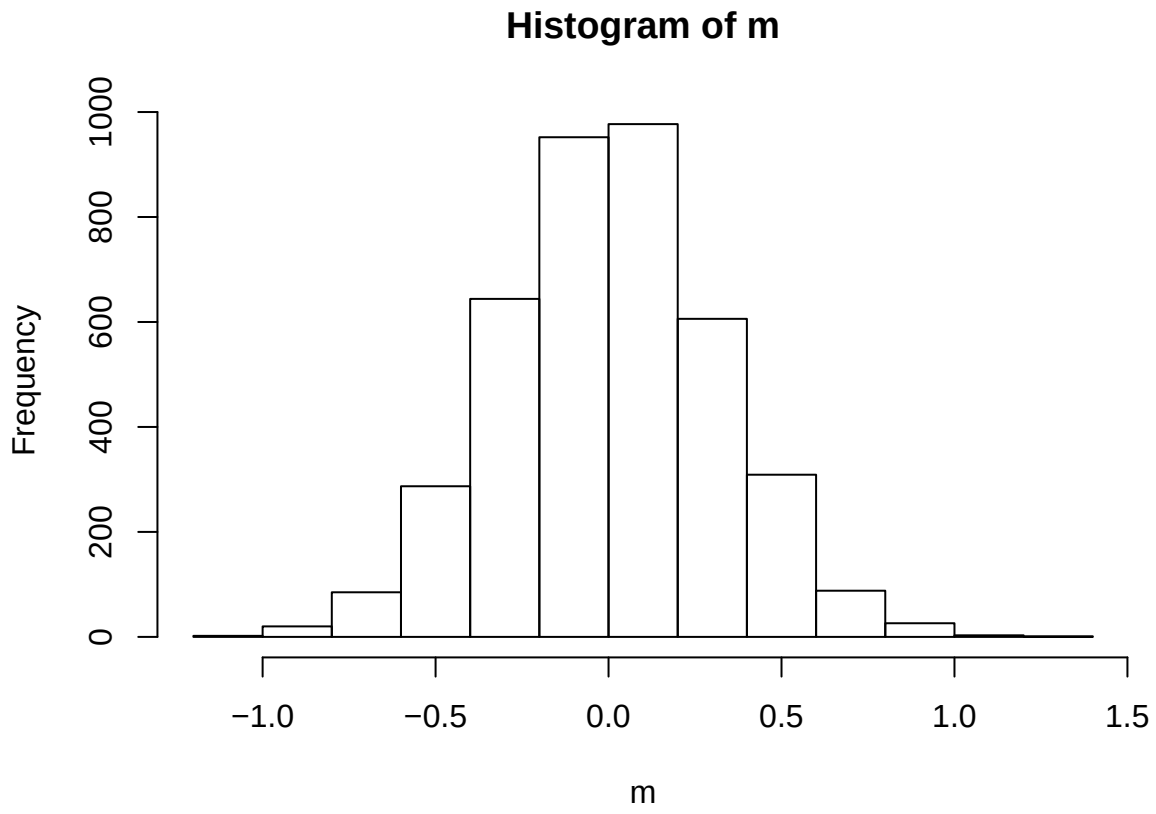
```
#function to compute means of cluster with random cluster centre
cluster_mean_comp <- function(s)
{
  #set number of clusters
  C <- 10
  all_cluster_members=c()
  #for each cluster do the following
for(i in 1:C)
{
  #set the cluster centre
  mu_c <- rnorm(1)
  #compute the individual cluster members
  cluster_members <-rnorm(N)+mu_c
  #append all members to a list
  all_cluster_members <- append(all_cluster_members,cluster_members)
}
  return(mean(all_cluster_members))
}

#function to compute means of cluster with cluster centre as 0
no_cluster_mean_comp <- function(s)
{
  #set number of clusters
  C <- 10
  all_cluster_members=c()
for(i in 1:C)
{
  #set cluster centre as 0
  mu_c <- 0
  cluster_members <-rnorm(N,0,1)+mu_c
  all_cluster_members <- append(all_cluster_members,cluster_members)
}
  return(mean(all_cluster_members))
}

print('With randomly generated cluster centres')
```

```
## [1] "With randomly generated cluster centres"
```

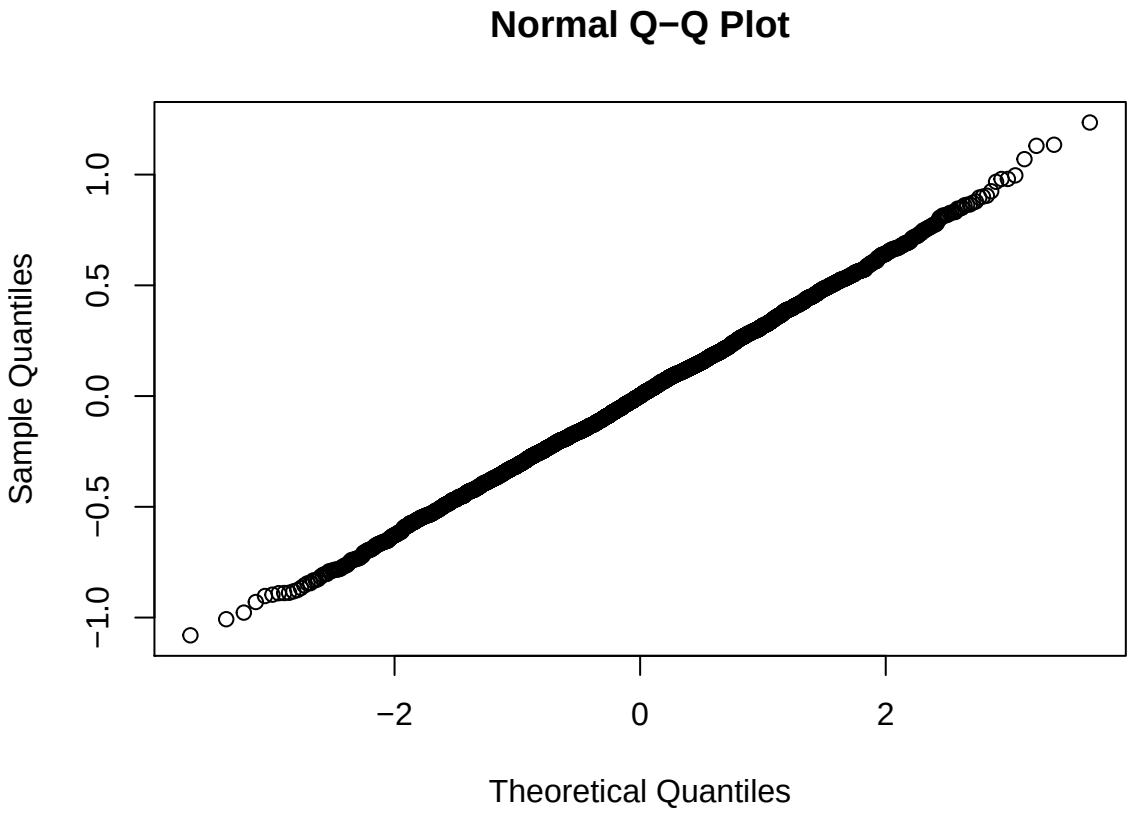
```
R <- 1000
S <- 1:(4*R)
m <- c()
#run cluster function 4R times
m <- append(m,lapply(S,cluster_mean_comp))
m <- unlist(m)
  hist(m)
```




```
  print(paste("Mean: ",mean(m)))
```

```
## [1] "Mean:  0.00245653591584931"
```

```
qqnorm(m)
```



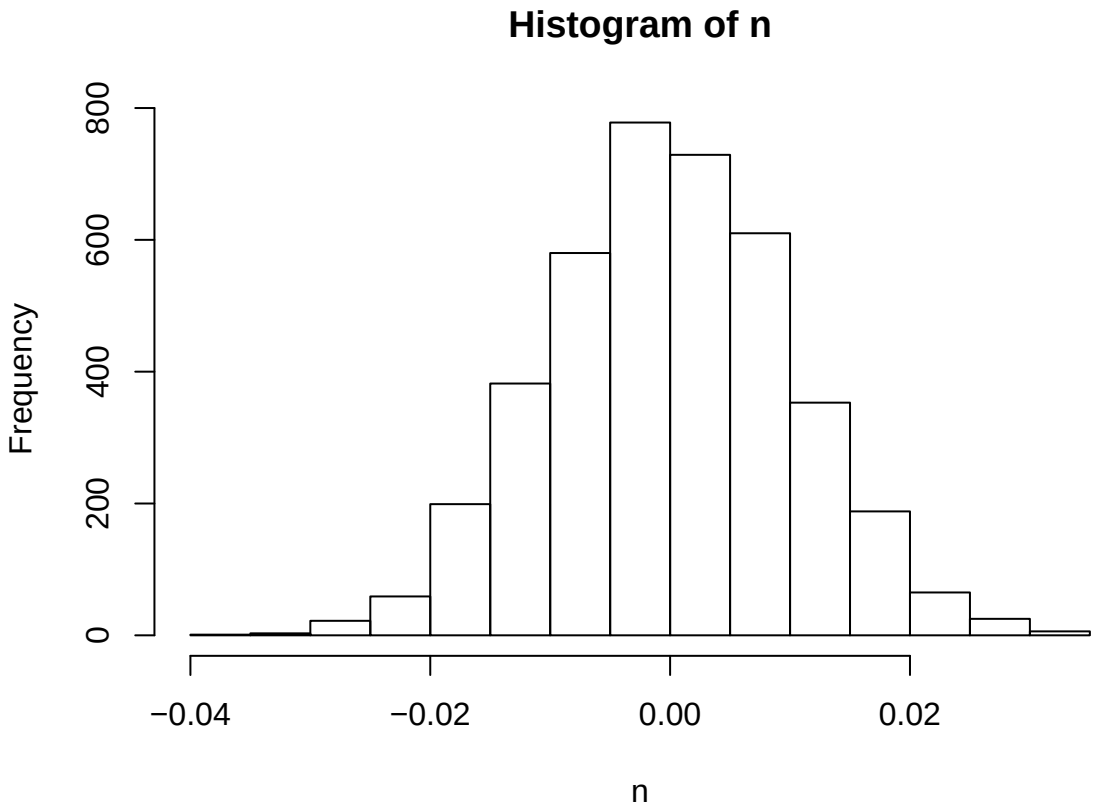

```
quantile(m,probs = c(0.025,0.975))
```

```
##       2.5%      97.5%
## -0.6153992  0.6328212
```

```
print('With 0 cluster centres')
```

```
## [1] "With 0 cluster centres"
```

```
n <- c()
#run cluster function 0 center 4R times
n <- append(n,lapply(S,no_cluster_mean_comp))
n <- unlist(n)
hist(n)
```




```
print(paste("Mean: ",mean(n)))
```

```
## [1] "Mean:  -3.3699856410541e-05"
```

```
quantile(n,probs = c(0.025,0.975))
```

```
##        2.5%       97.5%
## -0.01933405  0.01982784
```

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?

Ans. I think the distribution of a researcher's influence would be highly unequally distributed as one researcher might have a much higher or much lesser influence than another researcher. A particular well known researcher could have much higher citations than a researcher who is not that well known. Its mean might be highly random.

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.

Ans. The expression for its PDF is - (a * (xm)^a) / (x)^a+1 for x >=xm where xm is the minimum value of X, and a is the shape parameter.

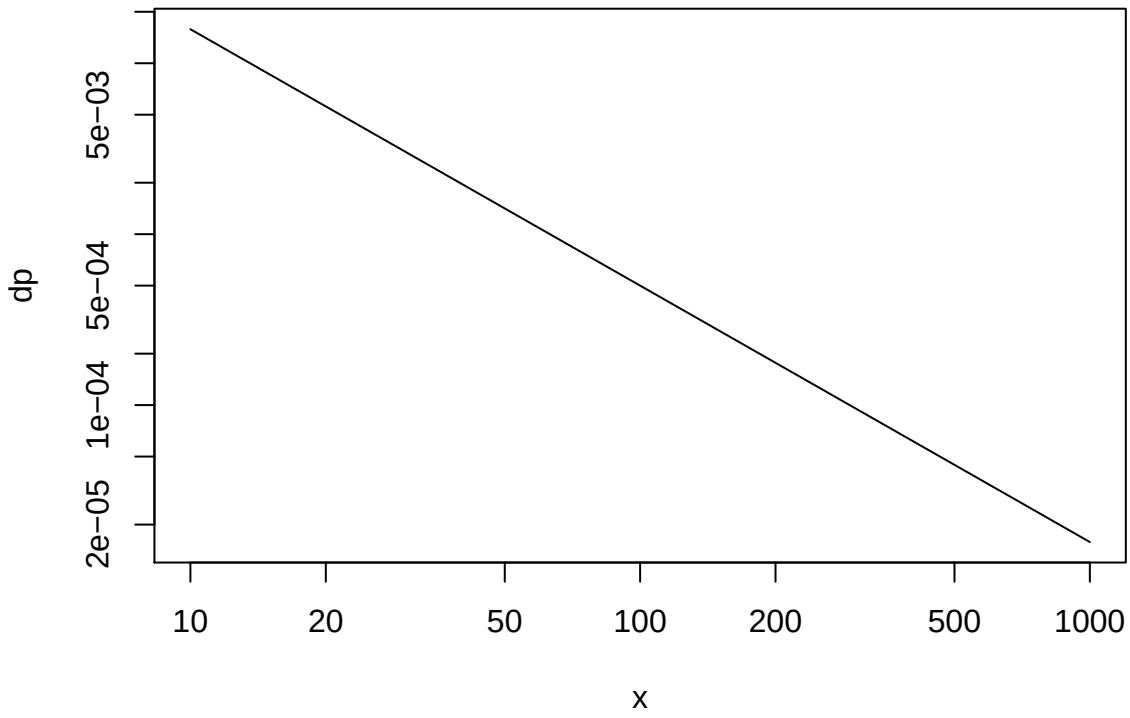
2. make a graph of it's pdf using log-log scale.

```
library(VGAM)
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```
dp <- function(x){dpareto(x,1,0.5)}
plot(dp,10,1000,log='xy')
```



3. what is it's expected value? What are the conditions?

Ans. Is expected value is infinity for a<=1 and (a*xm)/(a-1) for a>1.

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.

Answer the similar questions as above:

1. does the distribution of m look normal?

Ans. The distribution of m seems normal as seen from qqnorm.

2. what is the 95% confidence interval of this distribution?

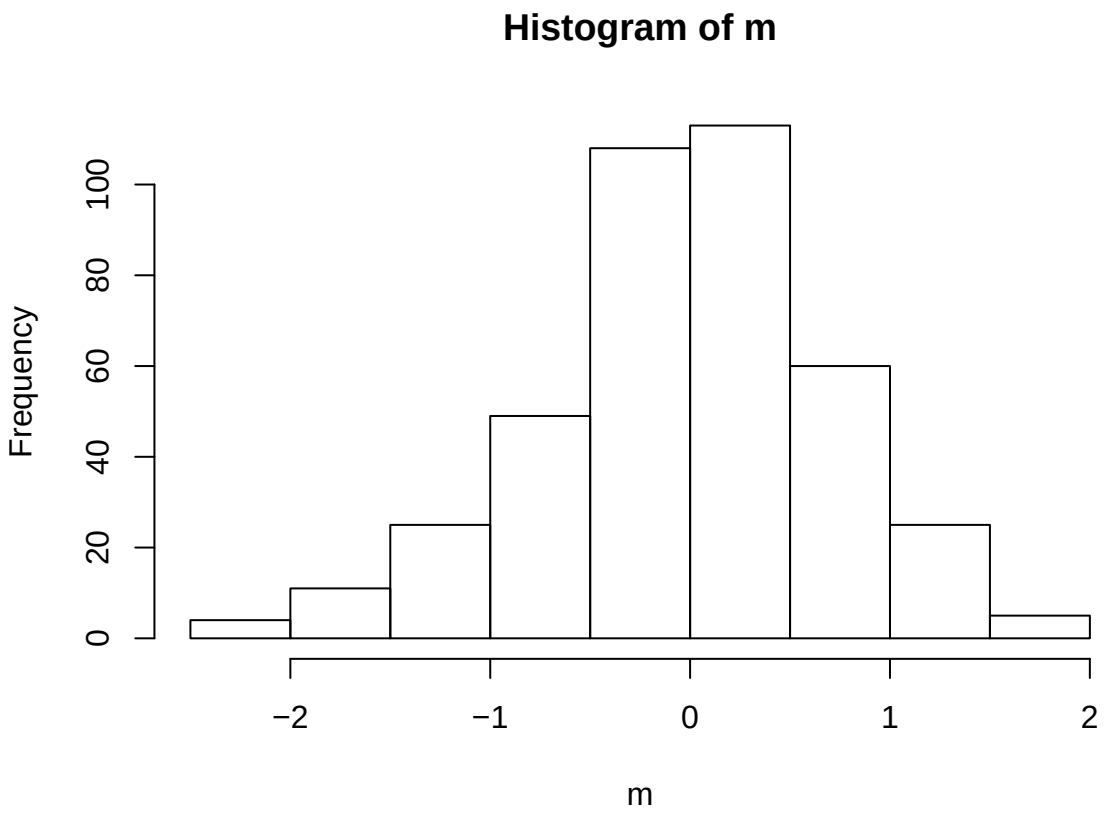
Ans. The 95% confidence interval for this distribution is (-1.466905,1.647726)

3. compare the outcome with the one above and explain the differences.

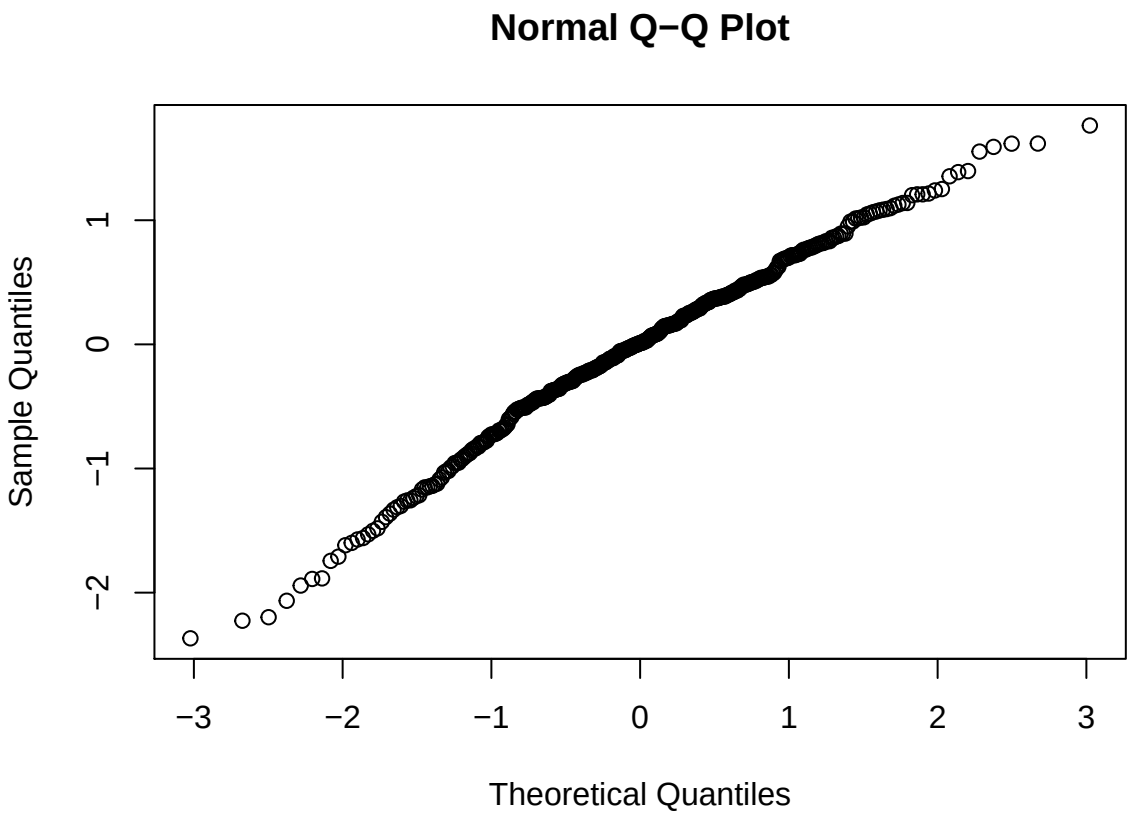
Ans. The difference in this case is that the 95% confidence interval is much higher than the equally sized clusters. The unequal size of the clusters results in a much higher variability in the distribution of cluster members around 0 (Theoretical mean of N(0,1)). A larger size of a cluster results the probability of a much greater spread.

```
library(VGAM)
#function to compute cluster mean of all cluster members
cluster_mean_comp <- function(s)
{
  C <- 10
  all_cluster_members=c()
  #compute cluster members for each cluster using randomly generated cluster size from pareto distribut
for(i in 1:C)
{
  mu_c <- rnorm(1)
  cluster_members <-rnorm(as.integer(rpareto(1,shape=0.5)))+mu_c
  all_cluster_members <- append(all_cluster_members,cluster_members)
}
  return(mean(all_cluster_members))
}


#compute cluster mean function 4R times
R <- 100
S <- 1:(4*R)
m <- c()
m <- append(m,lapply(S,cluster_mean_comp))
m <- unlist(m)
#plot histogram of m and compute quantiles
hist(m)
```




```
qqnorm(m)
```




```
quantile(m,probs = c(0.025,0.975))
```

```
##      2.5%     97.5%
## -1.599622  1.215665
```

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?

Ans. The appropriate probability distribution for analyzing the number of falls during a one hour race, given the average number of falls is the Poisson distribution. Using this ditribution, we can determine the probability of say an x number of falls (p(x)), given that a runner falls once during a one hour race. Each fall of the runner is an independent event, and the rate at which events occur is constant.

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

Ans. Both the expected value and variance of the number of times the athelete falls is lambda , whcih is the average number of times a runner falls during a one hour race, which is 1.

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

Ans. The pmf for poisson distribution is given by -> P(K=4) = (e ^ -lambda)*(lambda ^ k)/(k!) For 4 falls, P(4)=(e^-1)*(1^4)/(4!), which is 0.0153. So the probability of 4 falls within a one hour race would be 0.0153 or 1.53% which is rare, so it would be exceptional if an experienced runner falls 4 times.

```
dpois(4,1,log=FALSE)
```

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

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

Ans. According to the above pmf for a poisson distribution, P(k<=2)=P(k=0)+P(k=1)+P(k=2). This comes out to 0.919. So the probability that the experienced runner will not fall more than twice is a one hour race is 0.919 which is extremely high.

```
dpois(0,1,log=FALSE)+dpois(1,1,log=FALSE)+dpois(2,1,log=FALSE)
```

```
## [1] 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?

Ans. The distribution that I would use to describe the actual number of show-ups for the flight would be the binomial distribution. This is because for each ticket sold, a passenger might show up or not show up. So the probability that x people show up for the n tickets sold can easily be determined using the binomial distribution. Each event here of a passenger showing up is independent.

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

Ans. The expected revenue in the case that the airline never overbooks is the cost of a ticket * the number of tickets sold. The number of tickets sold here is equal to the number of seats here, so no passenger will be refused boarding, so there would be never any penalties. So the revenue here would be $40,000. My assumption here is that the passenger pays for the ticket at the time of booking. The expected revenue can also be computed as summation n times p(n), which also comes out to be 40000 dollars.

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

Ans. The probability that all 201 passengers will show up is (0.99)^201, which is 0.1326.

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

Ans. The expected revenue in this case would be 40,200 dollars for the 201 seats sold. The expected losses for the one passenger who was refused would be 1,000 dollars. So the expected profits would be 39,200 dollars. I would recommend overbooking as it could increase the expected revenue for the airline.

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

Ans. The probability that all 202 passengers will show up for 202 tickets sold is (0.99)^202, which is 0.1313.

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

Ans. Using the formula for binomial distribution, the probability that 201 out of 202 tickets sold shows up is 0.2679.

```
dbinom(201,size = 202,prob=0.99)
```

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

7. Would it be advisable to sell 202 tickets?

Ans. It would not be advisable to sell 202 tickets a sthe expected revenue in this case would be about 39869.44 dollars which would be lesser than the 40000 dollars that the airline would earn normally by selling 200 tickets. The assumption here is that each passenger pays for tickets before showing up for the flight.

```
exp_revenue <- 0
#calculating expected revenue
for(i in 0:202)
{
  if(i<=200)
  {
  #probability of i passengers showing up * revenue if i pass. show up where i <=200
  exp_revenue <- exp_revenue+ dbinom(i,size = 202,prob=0.99)*200*202
  }
  else
  {
    #probability of i passengers showing up * revenue if i pass. show up where i > 200
    exp_revenue <- exp_revenue+ dbinom(i,size = 202,prob=0.99)*(200*202 - (i-200)*1000)
  }
}
print(exp_revenue)
```

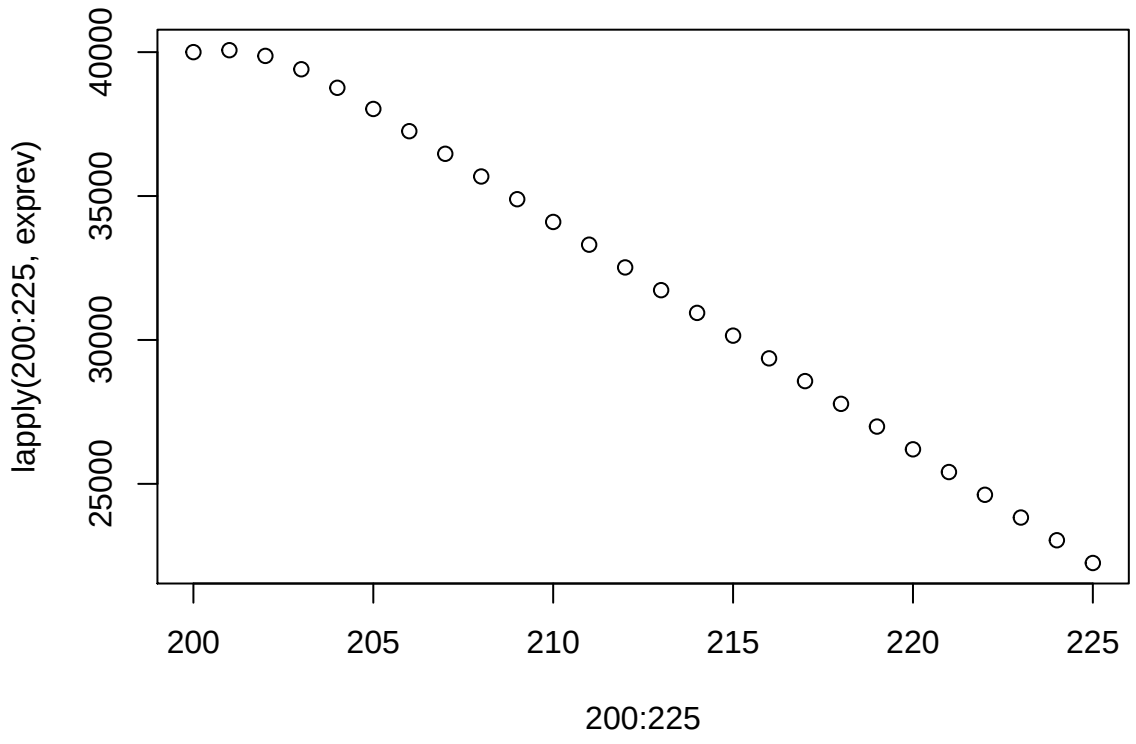
```
## [1] 39869.44
```

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

Ans. According to my analysis, 201 seats is the optimum number of seats that the airline should sell as it maximizes the expected revenue. The expected profits (revenue-penalties) in this case would be approximately 40067.36 dollars. If the airline attempts to sell any more tickets, the expected profits keep declining as in the graph below.

```
#function to compute expected revenue
   exprev <- function(x)
   {
     exp_revenue <- 0
for(i in 0:x)
{
  if(i<=200)
  {
    #exp. revenue = probability that i passengers show up * revenue when i pass. show up, i<=200
  exp_revenue <- exp_revenue+ dbinom(i,size = x,prob=0.99)*200*x
  }
  else
  {
     #exp. revenue = probability that i passengers show up * revenue when i pass. show up, i>200
    exp_revenue <- exp_revenue+ dbinom(i,size = x,prob=0.99)*(200*x - (i-200)*1000)
  }
}
return(exp_revenue)
   }
#Calculate number of seats at which expected revenue is maximized
maxrev <-0
k <-0
for(j in 200:300)
{
  expected_revenue=exprev(j)
```

```
  if(expected_revenue>maxrev)
  {
    maxrev=expected_revenue
    k=j
  }
}
#plot graph of expected revenue
plot(200:225,lapply(200:225,exprev))
```




```
print(maxrev)
```

```
## [1] 40067.36
```

```
print(k)
```

```
## [1] 201
```

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.