INFX 573: Problem Set 5 - Statistical Theory

TAPASVI BANSAL

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(5) # 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?

```
N
```

```
## [1] 1000
```

```r
M =array()
M <- rnorm(N)      #commented to keep the value of M same for the calculations
mean(M)
```

```
## [1] 0.01739946
```

```r
sum(M < -1.96 | M > 1.96)
```

```
## [1] 60
```

```r
quantile(M,probs=c(.025,.975))
```

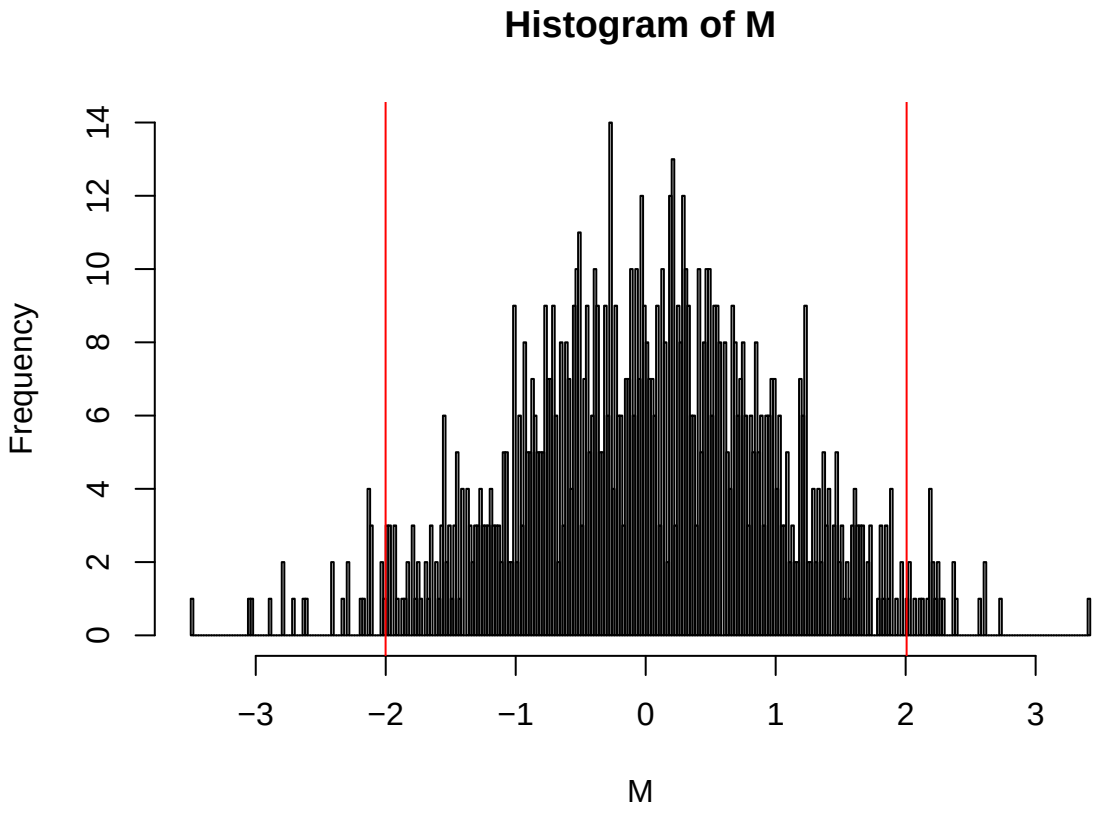
```
##      2.5%     97.5%
## -2.000981  2.007319
```

```r
# For finding the values that fall out of the range
for(i in 1:N)
  {if (M[i] < -1.96 | M[i] > 1.96) { print(M[i])}}
```

```
## [1] -2.183967
## [1] 2.215461
## [1] -2.000473
## [1] -2.102329
## [1] 2.387233
## [1] -1.995387
## [1] -1.965653
## [1] 2.600142
## [1] -1.978528
## [1] 2.180236
## [1] -2.621345
## [1] 2.246255
## [1] 2.181647
## [1] 2.025197
## [1] -3.498059
## [1] 2.197429
## [1] 1.962499
## [1] -2.884941
## [1] -2.334692
## [1] -2.110555
## [1] 2.029597
## [1] 2.116047
## [1] -2.707878
## [1] 2.006861
## [1] -1.980786
## [1] 2.245491
## [1] -3.034946
## [1] -2.13841
## [1] -2.288009
## [1] 3.401872
## [1] 1.966845
## [1] 2.367204
## [1] -2.79496
## [1] -2.119884
## [1] 2.562357
## [1] 2.281291
## [1] -2.403686
```

```
## [1] 2.268594
## [1] 2.137292
## [1] -3.04211
## [1] 2.724207
## [1] -2.792593
## [1] 2.609547
## [1] -2.122147
## [1] 2.364868
## [1] 2.214272
## [1] -2.02081
## [1] -1.984728
## [1] -1.963281
## [1] 2.074379
## [1] -2.604796
## [1] -2.13141
## [1] -2.178007
## [1] 2.197315
## [1] -2.406618
## [1] 2.235763
## [1] -2.280683
## [1] -2.125499
## [1] 2.174464
## [1] -2.029448
```

```r
# preliminary for plotting the graph
qts <- quantile(M,probs=c(.025,.975))

# Plotting the graph
hist(M, breaks = 300) %>%
abline(v=qts[1],col="red") %>%
abline(v=qts[2],col="red")
```



The mean value is 0.03980. The range -2.003975 to 2.005563 is 95% confidence interval. Values -2.003975 and 2.005563 are considered statiscally significant at 95% confidence level

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%`.

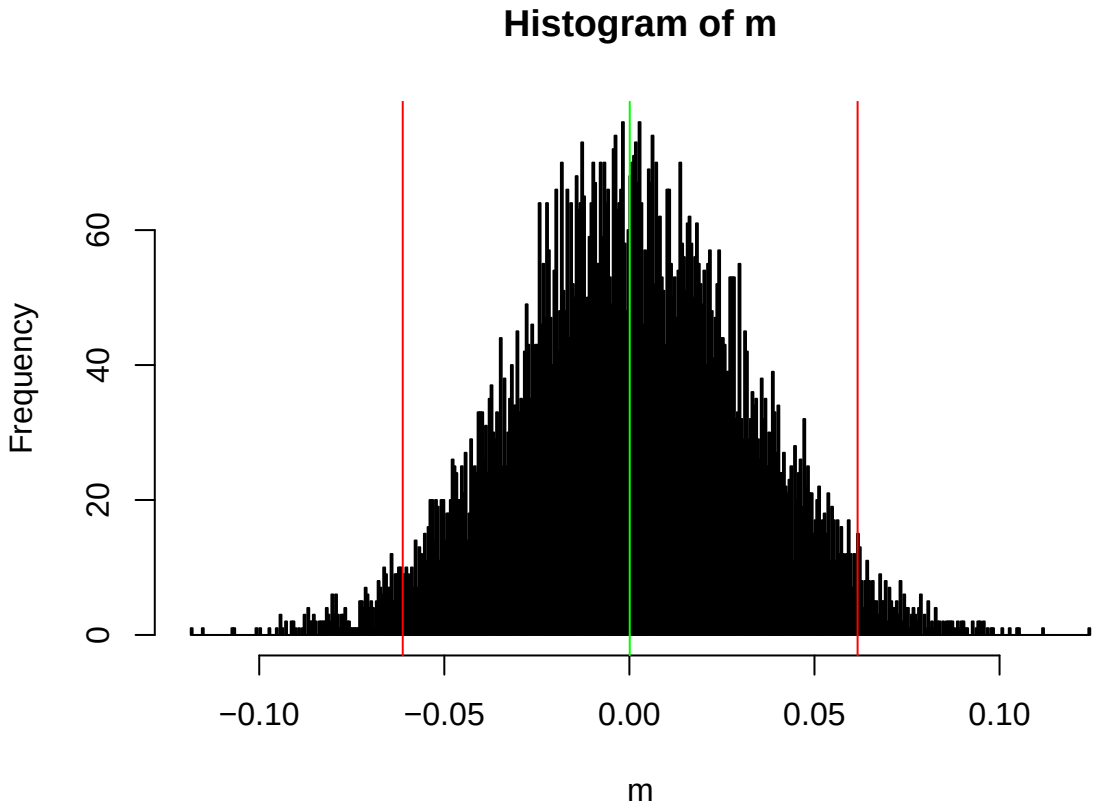
```
R <- 10000
m = array()

# M from the previous question, the values of M remains the same
for(i in 1:R)
  m[i]=mean(rnorm(M))


# calculating the total number of means falling out of the theoritical range
qts <- quantile(m,probs=c(.025,.975))
qts
```

```
##        2.5%       97.5%
## -0.06123226  0.06165493
```

```r
sum(m < qts[1]  | m > qts[2])
```

```
## [1] 500
```

```r
# For finding the values that fall out of the range
 # for(i in 1:R)
 #   {if (m[i] < -0.06121177 | m[i] > 0.06164436) { print(m[i])}}


# preliminary for plotting the graph
med <- median(m)

# Plotting the graph
hist(m, breaks = 500) %>%
abline(v=qts[1],col="red") %>%
abline(v=qts[2],col="red") %>%
abline(v=med, col="green")
```



500 means fall out of the theoretical range.

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)

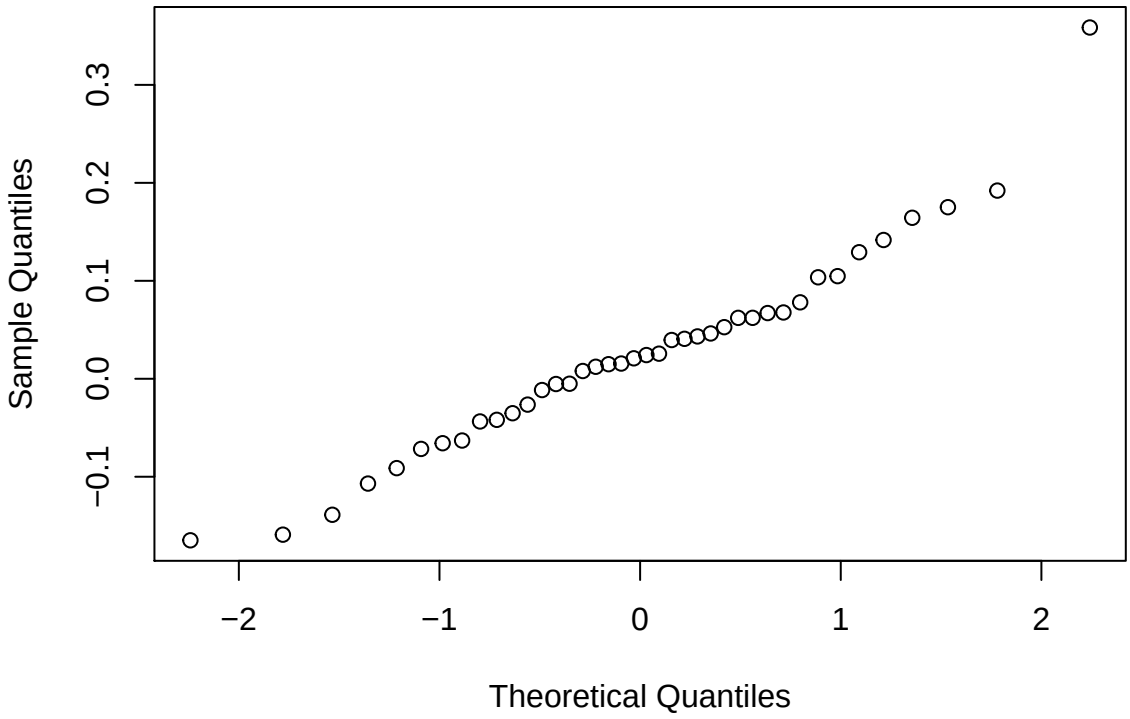
2. create cluster centers μ_c for each cluster $c \in \{1, ..., C\}$ by sampling from $N(0, 1)$.
3. create N cluster members for each cluster. The value for cluster member should be shifted by the cluster center: $x_{ci} = \mu_c + \epsilon_i$ where $\epsilon \sim N(0, 1)$.
4. compute the total mean of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

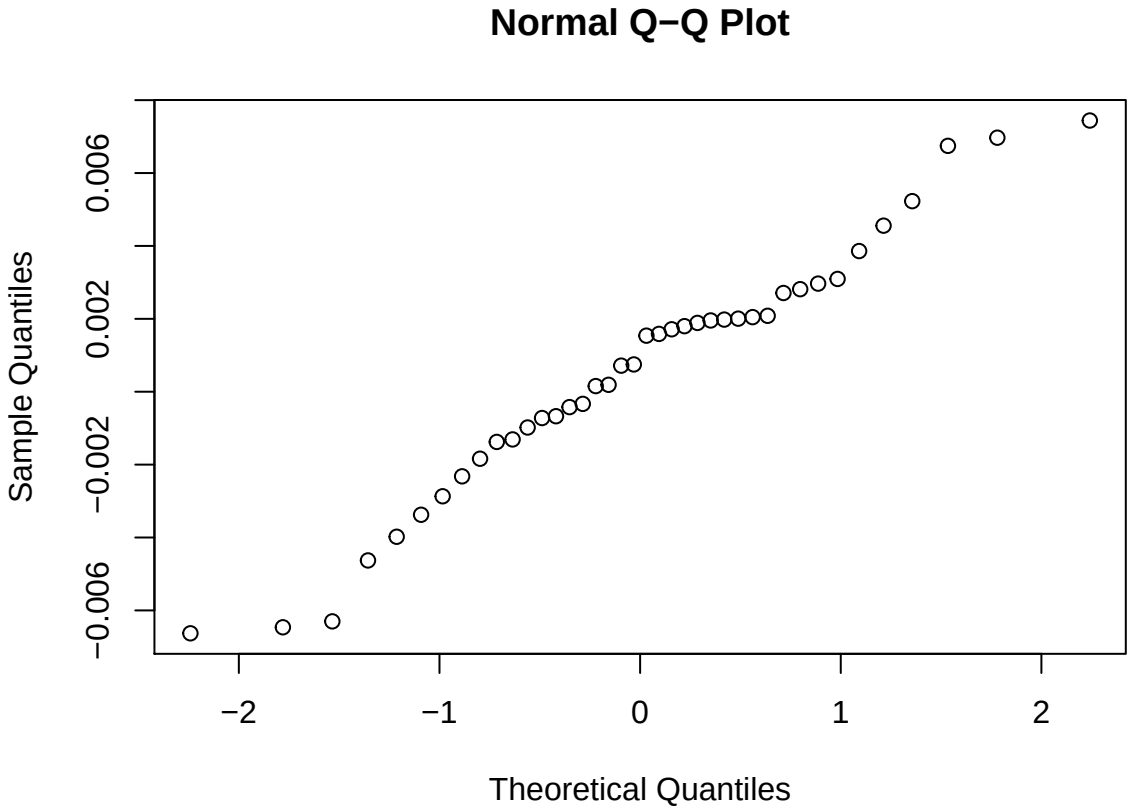
```
# Data Generating Process (DGP)
C <- 100
R <- 40
x <- list()
y <- list()
m <- list()
uc <- list()

for (j in 1:R)
  {
    uc <- rnorm(C)                    # Remove in case 5
    for (i in 1:C)
    {
      x[[i]] <- uc[i] + rnorm(1000)
    }
      y[[j]] <- unlist(x)
      m[[j]] <- mean(y[[j]])
  }
cm <- unlist (m)
qqnorm(cm)
```




```
quantile(cm,probs=c(.025,.975))
```

```
##       2.5%      97.5%
## -0.1592621  0.1962667
```

```
#hist(cm, breaks = 100)

# FOR NO CLUSTERING
C <- 100
R <- 40
x <- list()
y <- list()
nm <- list()
uc <- list()

for (j in 1:R)
  {
    uc <- rnorm(C)                      # Remove in case 5
    for (i in 1:C)
    {
      x[[i]] <- rnorm(1000)
    }
      y[[j]] <- unlist(x)
      nm[[j]] <- mean(y[[j]])
  }

ncm <- unlist (nm)
qqnorm(ncm)
```




```
quantile(ncm,probs=c(.025,.975))
```

```
##          2.5%        97.5%
## -0.006467431  0.006982142
```

```
#hist(ncm, breaks = 100)
```

1. If repeated 1000 times i.e R = 1000, there's straight line in the Normal Q-Q plot, the distribution of multiple m looks normal. If repeated 20 times i.e R = 20, then the points appear to fall in almost a straight line
2. The 95% confidence interval of the distribution (R = 40, C = 100 and cluster members = 1000) appears to be : -0.2026188 0.1517951
3. The 95% theoretical confidence intervals in case of no clustering (R = 40, rnorm = 1000) appears to be: -0.006991480 0.007433121
4. The confidence interval in case of clustering so much larger than for no clustering is because the data is scatterred more. The clusters are formed normally around different cluster centers hence the distribution of data is more. In other words, in case of no clustering the distribution is more condensed.
5. In case of not re-generating the cluster centers, we observe that the distribution remains "almost" normal with a little variation. But in case of no clustering there seems to be generation of identical values.

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?

Distribution of researchers' influence (say, number of citations) can be said to vary with the subject of the researcher's field. For example if the researcher's research topic is the current trend of the industry then there might be high number of people in academia citing the researcher's work. So in summary, the distribution of researchers' influence can vary alot.

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?

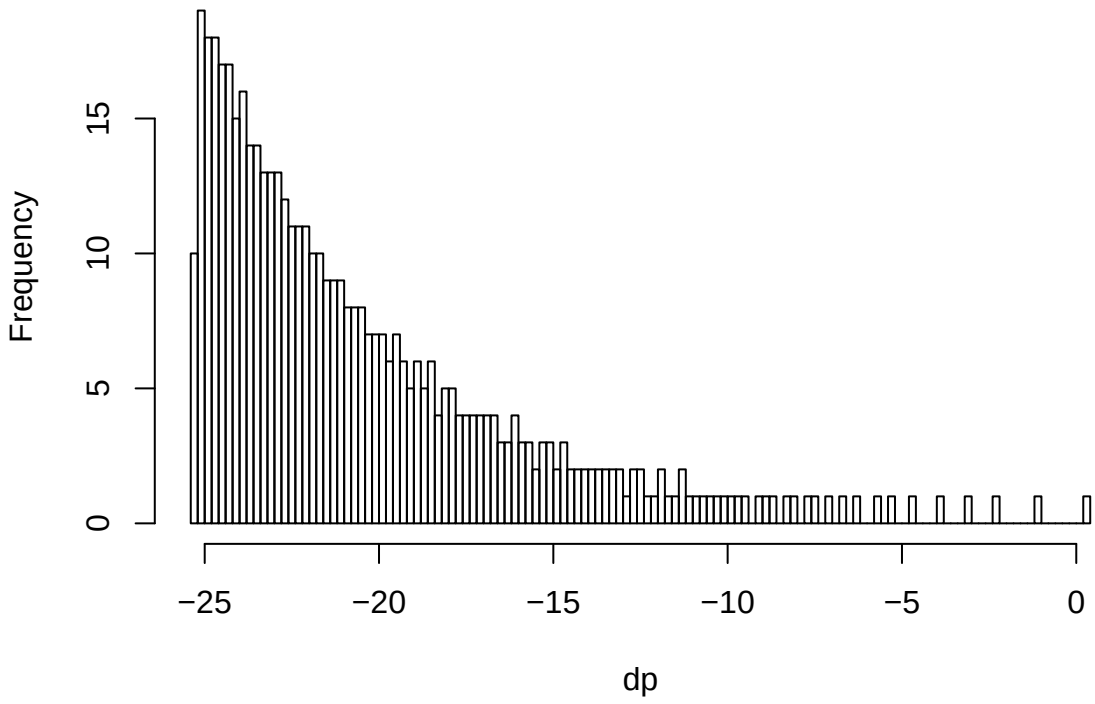
```
# f(x) = (\alpha x_m^\alpha)x^(\alpha+1)  # x >= x_m
# f(x) = 0 #x < x_m
#
# x is random variable
# x_m is the minimum possible value x can have
# alpha is some shape parameter (called "tail index")

#install.packages("rmutil")
#install.packages("VGAM")
library(rmutil)
```

```
##
## Attaching package: 'rmutil'

## The following object is masked from 'package:foreach':
##
##     times

## The following object is masked from 'package:stats':
##
##     nobs
```

```
library(VGAM)
```

```
## Loading required package: stats4

## Loading required package: splines

##
## Attaching package: 'VGAM'

## The following objects are masked from 'package:rmutil':
##
##     dbetabinom, dlaplace, dlevy, dpareto, dsimplex, pbetabinom,
##     plaplace, plevy, ppareto, qlaplace, qlevy, qpareto,
##     rbetabinom, rlaplace, rlevy, rpareto, rsimplex
```

```
dp <- dpareto(c(1:500),3,4, log = TRUE)
hist(dp, breaks = 100)
```




```
?rpareto
```

```
## Help on topic 'rpareto' was found in the following packages:
##
##   Package               Library
##   rmutil                /Library/Frameworks/R.framework/Versions/3.4/Resources/library
##   VGAM                  /Library/Frameworks/R.framework/Versions/3.4/Resources/library
##
##
## Using the first match ...
```

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

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

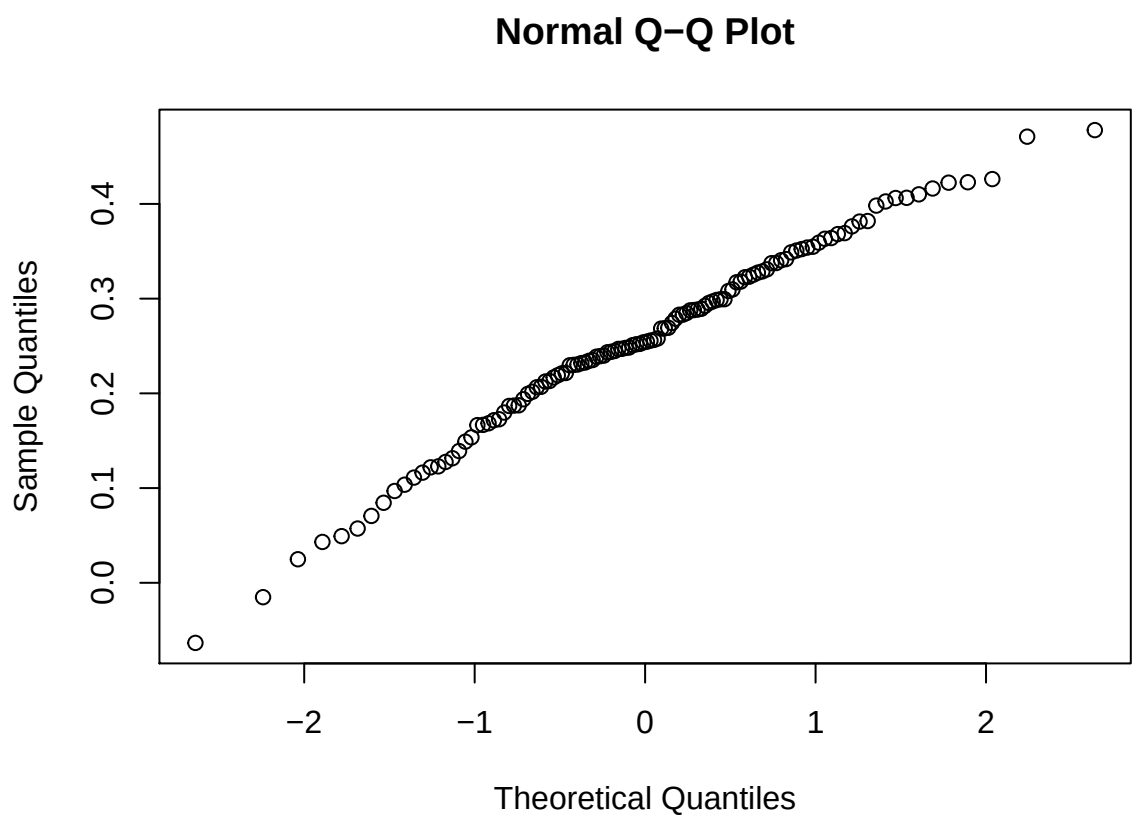
1. pick number of clusters C (10 is a good choice)
2. create cluster sizes N_c using Pareto distribution. Pick a highly unequal version using the shape parameter ≤ 1. You can set the minimum size to 1.
3. create cluster centers μ_c for each cluster $c \in \{1, ..., C\}$ by sampling from $N(0, 1)$.
4. 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?
2. what is the 95% confidence interval of this distribution?
3. compare the outcome with the one above and explain the differences.

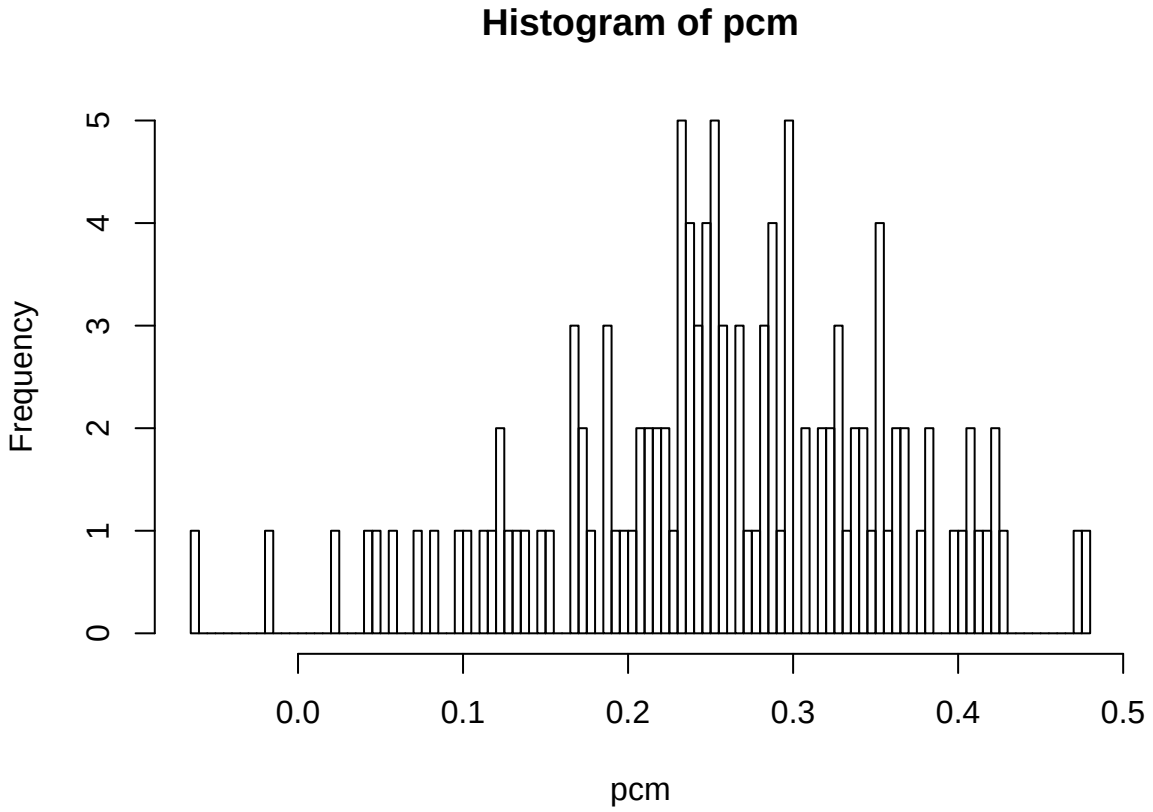
```
C <- 100
#ndp <- rpareto(100,0.1,5)
R <- 120
x <- list()
y <- list()
m <- list()
uc <- list()

for (j in 1:R)
  {
    uc <- rnorm(C)                      # Remove in case 5
    ndp <- rpareto(1000,0.2,4)
    for (i in 1:C)
    {
      x[[i]] <- uc[i] + ndp
    }
      y[[j]] <- unlist(x)
      m[[j]] <- mean(y[[j]])
  }
pcm <- unlist (m)
qqnorm(pcm)
```




```
quantile(pcm,probs=c(.025,.975))
```

```
##       2.5%      97.5%
## 0.04267128 0.42297500
```

```
hist(pcm, breaks = 100)
```



1. Yes, the distribution looks normal
2. The 95% confidence interval of this distribution is: 0.0692815 0.4892455
3. The confidence interval is mostly negative because the distributive is right skewed.

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?
2. What is the expected value and variance of the number of time the athlete falls?
3. Would it be exceptional if the runner falls 4 times?
4. What is the probability that the runner will fall no more than twice during a given (1hr) race.

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?

Bernoulli Distribution

2. Assume the airline never overbooks. What is it's expected revenue? Expected Revenue = 39600

3. Now assume the airline sells 201 tickets for 200 seats. What is the probability that all 201 passengers will show up?
4. What are the expected profits (= revenue − expected losses) in this case? Would you recommend overbooking over booking the just right amount?
5. Now assume the airline sells 202 tickets. What is the probability that all 202 passengers show up?
6. What is the probability that 201 passengers – still one too many – will show up?
7. Would it be advisable to sell 202 tickets?
8. What is the optimal number of seats to sell for the airline? How big are the expected profits?

```
#3
#the probability that all 201 passengers will show up
p <- 201* (0.99)^200 * 0.01
#The probability is approx 27% (0.2692991)


#4
202* (0.99)^20 * 0.01^2
```

```
## [1] 0.01652172
```

```
#The probability is approx 16%
```

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.