Report of Lab 1: Robustness and distribution assumptions

Alberto Nieto Sandino

November 2, 2018

1 Assignment 1.1

In Figure 1, we can see the histogram, pp- and qq-plot of the normally generated data. As we can see, the histogram has the shape of a normally distributed probability distribution function (pdf). The pp- and qq-plots show as well that the probabilities and quantiles follow a normal distribution as there is no significant deviations from the null hypothesis of being normally distributed data.

We also performed the Chi-square formal goodness-of-fit test as well as the Kolmogorov-Smirnov. Both tests reported not rejecting the null hypothesis, meaning that the $p-$value was significant enough ($p > 0.05$) to indicate that the data is normally distributed.

In Figure 2, we can see the histogram, pp- and qq-plot of the gamma generated data. Looking at the histogram, it is obvious the distribution does not correspond to a normal distribution. The pp- and qq-plots deviate significantly from the null hypothesis (diagonal line), especially when looking at the tails, which indicates that it is not a normally distributed data. The formal goodness-of-fit Chi-square and Kolmogorov-Smirnov both reject the null hypothesis, since the $p-$values are non-significant ($p = 0.0187$ and $p = 1.1470 \cdot 10^{-54}$, respectively). All tests, therefore, reject the null hypothesis of this distribution being normal.

We also plotted the qq-plot for the gamma distributed data in R as seen in Figure 3. It is especially noticeable in the tails that the data is not normally distributed. The chi-square test (done in R) also rejected the null hypothesis with a $p-$value of $1.513635 \cdot 10^{-5}$.

2 Assignment 1.2

For the 1000 simulated confidence intervals generated with the normally distributed data, 961 of them covered the true value of the variance σ^2, meaning 96.1% of them covered the true value. Thus, this is what we would expect since we are assuming this confidence interval covers the true value with a 95% confidence.

For the 1000 simulated confidence intervals generated with the gamma distribted data, 801 of them covered the true value of the variance σ^2, meaning 80.1% of them covered the true value. This does not guarantee the 95% confidence that we are supposed to get, which is obvious since the data is not normally distributed, breaking that assumption. Doing the same calculations in R, we obtain that 788 of the 1000 CIs cover the true value, meaning a 78.8% coverage, which is congruent with our results in Matlab.

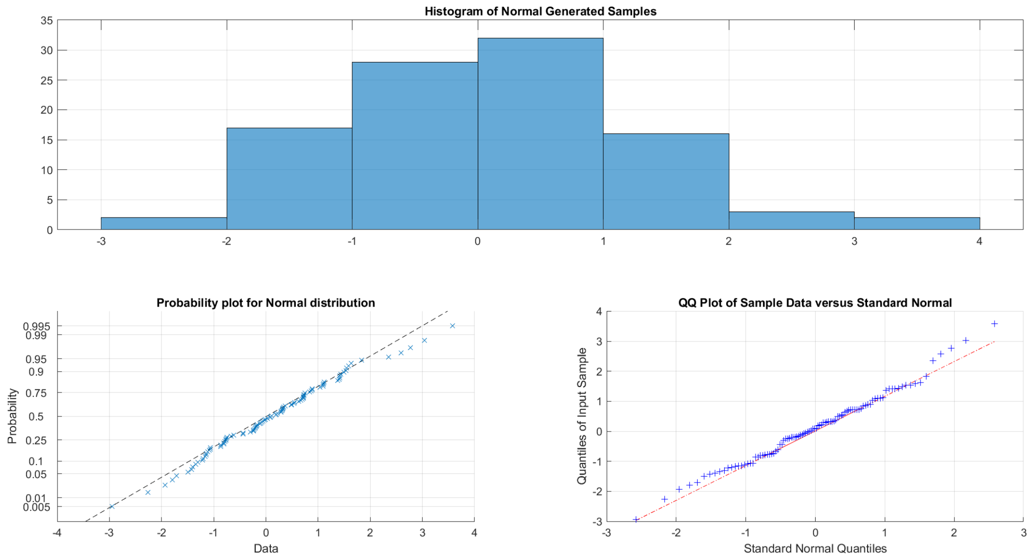

Figure 1: Histogram(top), pp-plot (bottom left) and qq-plot (bottom right) of the simulated normally distributed data

3 Assignment 2.1

In Figure 4, we can see the $\epsilon-$contaminated data. Based on the histogram, the pp- and the qq-plot, we can conclude it is not normally distributed.

4 Assignment 2.2

In Figure 5, we can see the contaminated and non-contaminated normal distribution. The mean of non-contaminated normal distribution at 25 is -0.51852 and at 975 is 0.68989. The mean for ϵ-contaminated normal distribution at 25 is -0.19933 and at 975 is 0.18997.

5 Assignment 2.3

In this task, we repeat Assignment 2.2, but with robust estimators: the sample median and $\alpha-$ *trimmed* mean. The median of non-contaminated normal distribution at 25 is -0.23166 and at 975 is 0.25775. The median for ϵ-contaminated normal distribution at 25 -0.2244 and at 975 is 0.25465 The α-trimmed mean of non-contaminated normal distribution at 25 is -0.19648 and at 975 is 0.2034. The α-trimmed mean for ϵ-contaminated normal distribution at 25 is -0.1926 and at 975 is 0.2058.

In Figure 7 and 8, we can see the histogram for both estimators calculated in R. In the R implementation, we obtain that the 25th value in the $\alpha-$trimmed mean is -0.1981304 and the 975th is 0.2128142. For the median at 25th value, we have -0.2405189 and for the 975th, it is 0.2234931.

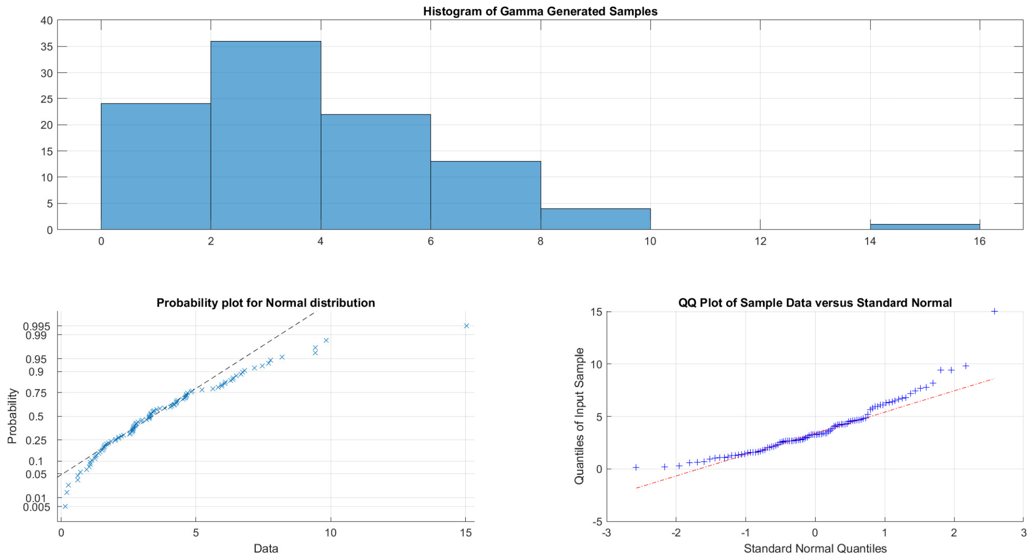

Figure 2: Histogram(top), pp-plot (bottom left) and qq-plot (bottom right) of the simulated gamma distributed data

These results point out that the median has bigger width on the CIs than the $\alpha-$trimmed mean. The median, however, can be seen to be having a more normal and smooth distribution than the trimmed mean. This points to the median as a more robust estimator.

Appendix - Matlab code

```
clear all;close all;clc;
%% Assignment 1 Effect of Distribution Assumptions
%% Assignment 1.1 Creating the data
clear all;close all;clc;
% In this assignmetn we are generating data that are normally and gamma
% distrbutied data and see their histogram, pp-plot and qq-plot
% for their precpetive distrbution. Also, we test the goodness-of-fit
% for each one of the function.
%%% Define the parameters for Normal
% rng default  % for reproducibility of the results

n           = 100; % Number of samples
mu          = 0; % Mean value for the normal distribution
sigma       = 1; % Standard Deviation the normal distribution

```

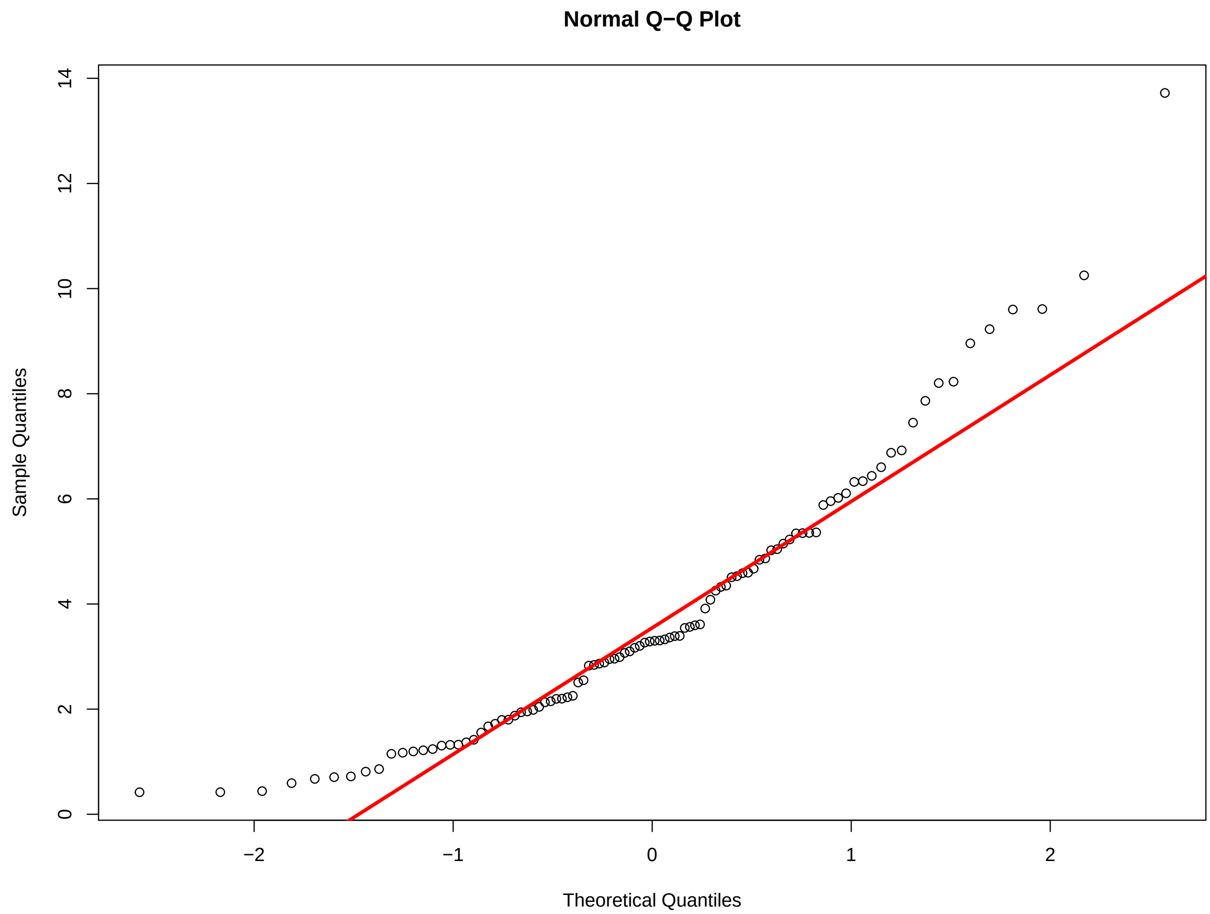

Figure 3: QQ-plot of the gamma distributed data

```
% define Normal probability distribution
npd          = makedist('Normal','mu',mu,'sigma',sigma);

% Generating a normal distrubtion samples with given parameters using ...
% a pre-defined function
nSamples     = normrnd(mu,sigma,1,n);
% nSamples     = mu + sigma.*randn(1,100);

% Normal Cumultive Distribution Function (cdf)
nSamplecdf   = cdf(npd,nSamples);
% nSamplecdf     = normcdf(XSamples,mu,sigma);

%%% Ploting of Histogram, pp-plot and qq-plot Normal Distribution
% plot the histogram of the data
```

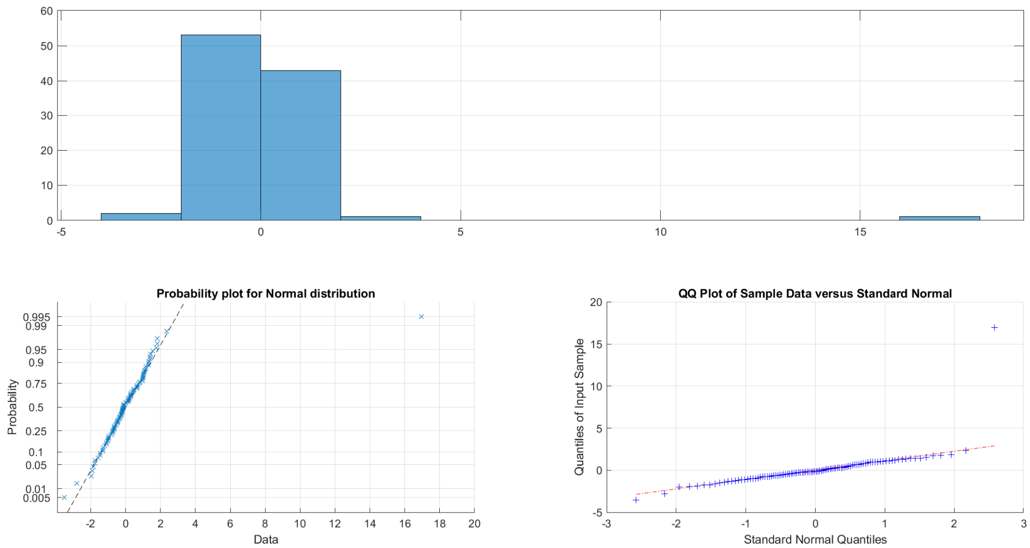

Figure 4: Histogram(top), pp-plot (bottom left) and qq-plot (bottom right) of the simulated contaminated data

```
subplot(211)
histogram(nSamples)
%histfit(nSamples)
title('Histogram of Normal Generated Samples')
grid on
hold on

% Ploting the PP-plot
subplot(223)
probplot(nSamples)
grid on

% Ploting the QQ-plot
subplot(224)
qqplot(nSamples)
grid on

%%%  Goodness-of-fit Test for Normal Distribution
% Chi-square test
[ncqh,ncqp_value] = chi2gof(nSamples);
disp('-----------Chi2test Normal----------------')
x= 'The decision for Goodness-of-fit test using Chi-square is %d. for Normal Distribution\n';
fprintf(x,ncqh)
```

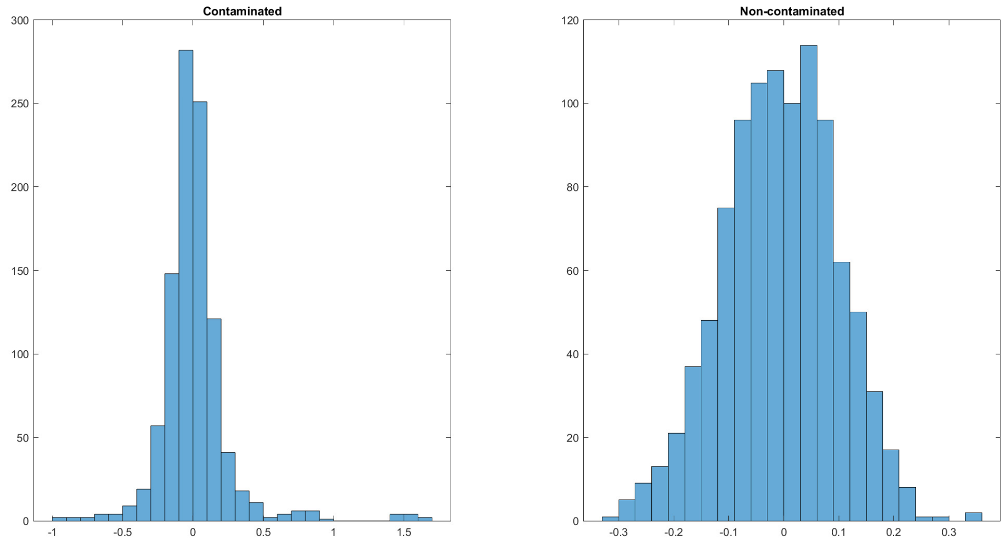

Figure 5: Histograms for the contaminated (left) and non-contaminated (right) distributions

```

if ncqh == 0
    X = ['\n Since the test decision for the null hypothesis of the data = ',num2str(ncqh),...
        '.\n Then, we do not reject the null hypothesis \n that the samples follows the Normal Distribution.\n'];
    fprintf(X)
else
    X = ['\n Since the test decision for the null hypothesis of the data = ',num2str(ncqh),...
        '.\n Then, we reject the null hypothesis \n that the samples follows the Normal Distribution.\n'];
    fprintf(X)
end

disp('----------------------------')

% Kolmogorov-Smirnov test
% If h = 1, this indicates the rejection of the null hypothesis.
% If h = 0, this indicates a failure to reject the null hypothesis.
[nkh,nkp_value] = kstest(nSamples);
disp('-----------KStest Normal----------------')
x = 'The decision for Goodness-of-fit test using Kolmogorov-Smirnov is %d. for Normal Distribution\n';
fprintf(x,nkh)
```

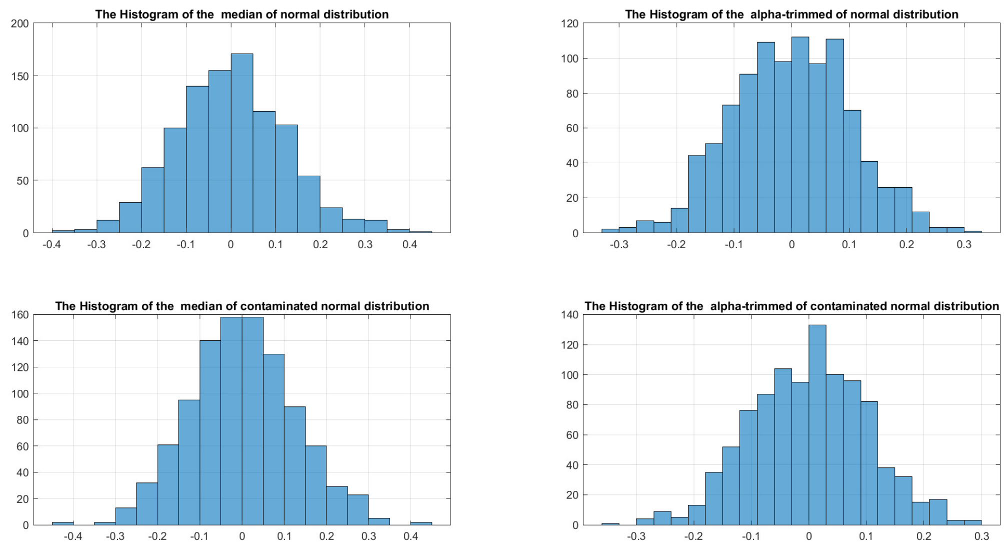

Figure 6: Histograms for the median and $\alpha - trimmed$ mean for a sample of size 1000

```
if nkh == 0
    X = ['\n Since the test decision for the null hypothesis of the data = ',num2str(nkh),...
         '.\n Then, we failed to reject the null hypothesis \n that the samples follows the Normal Distribution.\n'];
    fprintf(X)
else
    X = ['\n Since the test decision for the null hypothesis of the data = ',num2str(nkh),...
         '.\n Then, we reject the null hypothesis \n that the samples follows the Normal Distribution.\n'];
    fprintf(X)
end

disp('-----------------------------')

%%% Define the parameters Gamma
alpha        = 2; % shape parameter for gamma distribution
beta         = 2; % scale parameter for gamma distribution

% define Gamma probability distribution
gpd          = makedist('Gamma','a',alpha,'b',beta);
```

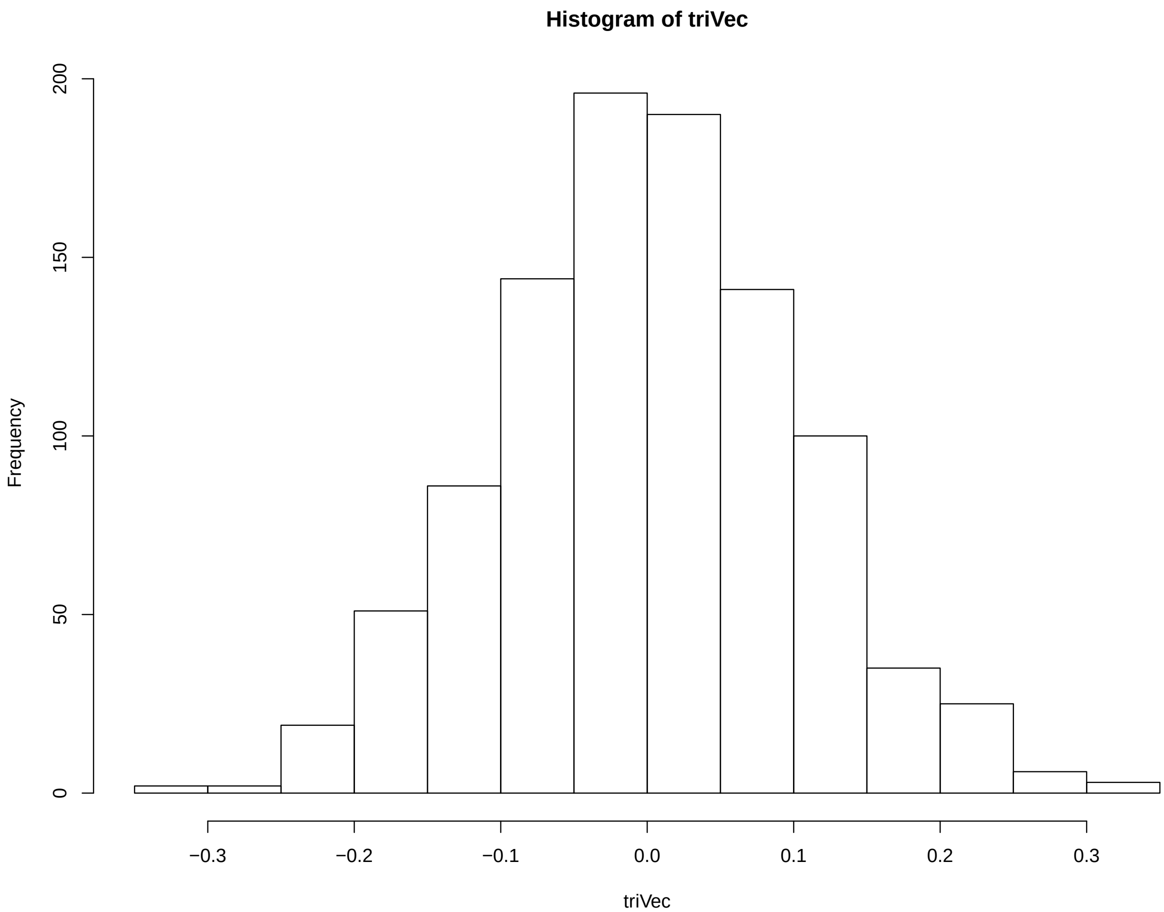

Figure 7: Histogram of the $\alpha - trimmed$ mean for a sample of size 1000

```
% Generating a gamma distribution samples with given parameters
  using ...
% a pre-defined function
gSamples     = gamrnd(alpha,beta,1,n);

%%% Ploting of Histogram, pp-plot and qq-plot Gamma Distribution
figure
% plot the histogram of the data
subplot(211)
histogram(gSamples)
title('Histogram of Gamma Generated Samples')
grid on
hold on

% Ploting the PP-plot
```

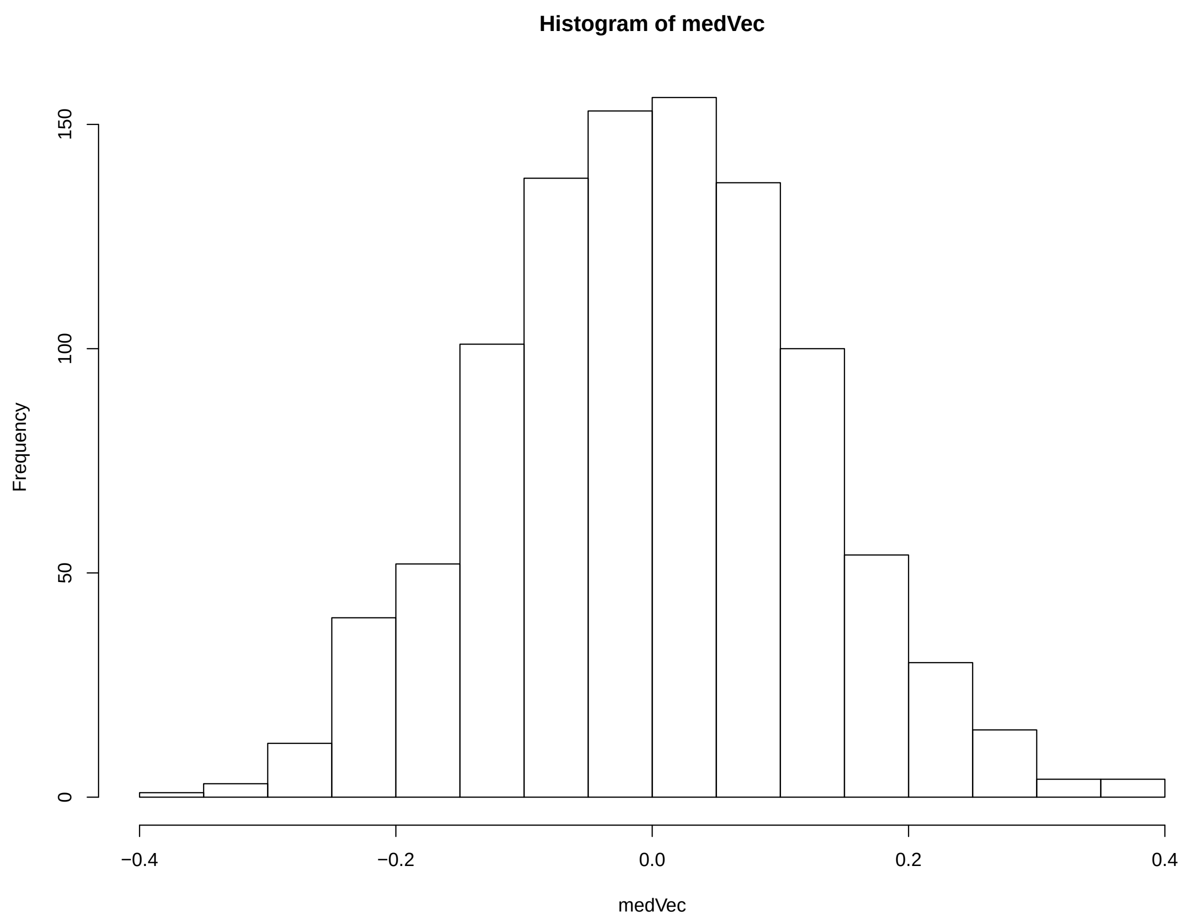

Figure 8: Histogram of the median for a sample of size 1000

```
subplot(223)
probplot(gSamples)
grid on

% Ploting the QQ-plot
subplot(224)
qqplot(gSamples)
grid on

%%  Goodness-of-fit Test for Gamma Distribution
% Chi-square test

% [gh,gp_value] = chi2gof(gSamples,'CDF', gpd);
[gh,gp_value] = chi2gof(gSamples);
disp('-----------Chi2test Gamma----------------')
```

```
x = 'The decision for Goodness-of-fit test using Chi-square is %d. for Gamma Distribution\n';
fprintf(x,gh)

if gh == 0
    X = ['\n Since the test decision for the null hypothesis of the data = ',num2str(gh),...
        '.\n Then, we do not reject the null hypothesis that the samples follows the Gamma Distribution.\n'];
    fprintf(X)
else
    X = ['\n Since the test decision for the null hypothesis of the data = ',num2str(gh),...
        '.\n Then, we reject the null hypothesis that the samples follows the Gamma Distribution.\n'];
    fprintf(X)
end

disp('-----------KStest Gamma----------------')

% Kolmogorov-Smirnov test
% If h = 1, this indicates the rejection of the null hypothesis.
% If h = 0, this indicates a failure to reject the null hypothesis.
[gkh,gkp_value] = kstest(gSamples);

x= 'The decision for Goodness-of-fit test using Kolmogorov-Smirnov is %d, for Gamma Distribution.\n';
fprintf(x,gkh)

if gkh == 0
    X = ['\n Since the test decision for the null hypothesis of the data = ',num2str(gkh),...
        '.\n Then, we failed to reject the null hypothesis that the samples follows the Normal Distribution.\n'];
    fprintf(X)
else
    X = ['\n Since the test decision for the null hypothesis of the data = ',num2str(gkh),...
        '.\n Then, we reject the null hypothesis that the samples follows the Normal Distribution.\n'];
    fprintf(X)
end

%% Assignment 1.2
disp('----------------A1.2-----------')
```

```
% rng default  % for reproducibility of the results
n            = 100; % Number of samples
mu           = 0; % Mean value for the normal distribution
sigma        = 1; % Standard Deviation the normal distribution
ncount        = 0; % counter
NN           = 1000; % testing cycle
width_CI     = zeros(1,NN); % width of the intervals
norm_CI = zeros(2,NN); % define size of variable

% Generating a normal distrubtion samples with given parameters using ...
% a pre-defined function
for i = 1:NN
    nSam_test         = normrnd(mu,sigma,1,n);
    [~,~,~, sigmaCI] = normfit(nSam_test); % calculate CI for sigma
    norm_CI(:,i) = sigmaCI;
    width_CI(:,i) = diff(sigmaCI);
    if (norm_CI(1,i) <= sigma^2 && norm_CI(2,i) >= sigma^2)
        ncount = ncount + 1;
    end
end
mean_width_of_intervals = mean(width_CI); % the mean of the width of intervals
disp(['Number of intervals which contain the true variance is ',...
    num2str(ncount),' and the average width of the intervals is ',...
    num2str(mean_width_of_intervals)])

%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(['------------------Gamma Dist----------------'])
alpha        = 2; % shape parameter for gamma distribution
beta         = 2; % scale parameter for gamma distribution
gsigma_sq     = alpha*beta^2; % variance for the gamma
gcount       = 0;            % Counter
width_gCI    = zeros(1,NN); % width of the intervals
gci = zeros(2,NN); % define size of variable

% Generating a gamma distribution samples with given parameters using ...
% a pre-defined function
for i = 1:NN
    gSam_test         = gamrnd(alpha,beta,1,n);
    [~,~,~, gsigmaCI] = normfit(gSam_test); % calculate CI for sigma
    gci(:,i) = gsigmaCI.^2;
    width_gCI(i)     = diff(gsigmaCI);
```

```
    if (gci(1,i) <= gsigma_sq && gci(2,i) >= gsigma_sq)
        gcount = gcount + 1;
    end
end
mean_width_of_intervals_g = mean(width_gCI);
disp(['Number of intervals which contain the true variance is ',...
    num2str(gcount),' and the average width of the intervals is ',...
    num2str(mean_width_of_intervals_g)])
%% Assignment 2 Robust Estimation
disp('--------------A2------------------')
%% Assignment 2.1
disp('--------------A2.1------------------')
%%% Define the parameters
n          = 100; % number of Samples
mu         = 0;
var        = 1;
eps        = 0.05;

% nSamp       = normrnd(mu_y,sqrt(var_y),1,n);
nSamp       = randn(1,n);
W = binornd(1,1 - eps,1,n); %bernoulli distr. is binomial distr . with n=1
Y = mu + sqrt(var).*nSamp;
Z = trnd(1,n,1).'; %student distr with parameter 1 is Cauchy distr.
contaimed_dist = W.*Y + (1-W).*Z; % make it a contaminated distribution

%%% Ploting of Histogram, pp-plot and qq-plot contaminated Distribution
figure
% plot the histogram of the data
subplot(211)
histogram(contaimed_dist)
grid on
% Ploting the PP-plot
subplot(223)
probplot(contaimed_dist)
grid on
% Ploting the QQ-plot
subplot(224)
qqplot(contaimed_dist)
grid on

[csqh,csqp_value] = chi2gof(contaimed_dist);
```

```
disp('-----------Chi2test----------------')
x = 'The decision for Goodness-of-fit test using Chi-square is %d. for contaminated Distribution\n';
fprintf(x,csqh)

if csqh == 0
    X = ['\n Since the test decision for the null hypothesis of the data = ',num2str(csqh),...
        '.\n Then, we do not reject the null hypothesis that the samples follows the contaminated Distribution.\n'];
    fprintf(X)
else
    X = ['\n Since the test decision for the null hypothesis of the data = ',num2str(csqh),...
        '.\n Then, we reject the null hypothesis that the samples follows the contaminated Distribution.\n'];
    fprintf(X)
end
%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Assignment 2.2
disp('--------------A2.2------------------')
%%% Define the parameters
NN          = 1000; % repteation number

resam_contaimed_dist = zeros(1,n);
resam_norm_dist = zeros(1,n);

mu_cd    = zeros(1,NN);
mu_norm      = zeros(1,NN);

for i = 1:NN
    % extract 100 non-contaminated r.v.
    resam_contaimed_dist = randn(1,n);
    W = binornd(1,1 - eps,1,n); %bernoulli dist. is binomial dist. with n=1
    Y = mu + sqrt(var).*resam_contaimed_dist;
    Z = trnd(1,n,1).'; %student distr with parameter 1 is Cauchy distr.
    % extract a set of 100 r.v. from the set
    resam_norm_dist = W.*Y + (1-W).*Z; % make it a contaminated distribution

    mu_norm(i)      = mean(resam_norm_dist);
    mu_cd(i)    = mean(resam_contaimed_dist);
end
```

```
figure
subplot(211)
histogram(mu_norm)
title('The Histogram of the  means of normal distribution')
grid on

subplot(212)
histogram(mu_cd)
title('The Histogram of the  means of eps-contaminated normal distribution')
grid on

mu_norm_sort = sort(mu_norm);
mu_X_test_sort = sort(mu_cd);

fprintf(...
    ['Ordering the means values and given the values at the 25 and 975\n',...
    '\n The mean of non-contaminated normal distribution at 25 = ' ...
    ,num2str(mu_norm_sort(round(0.025*NN))), ' and at 975 = ', ...
    num2str(mu_norm_sort(round(0.975*NN))),...
    '.\n The mean for eps-contaminated normal distribution at 25 =' ...
    ,num2str(mu_X_test_sort(round(0.025*NN))), ' and at 975 = ', ...
    num2str(mu_X_test_sort(round(0.975*NN))),'\n'])

%%%%%%
%% Assignment 2.3
disp('--------------A2.3------------------')
%%% Define the parameters
alpha       = 0.1;

median_X_test    = zeros(1,NN);
median_norm      = zeros(1,NN);

trimmean_X_test    = zeros(1,NN);
trimmean_norm      = zeros(1,NN);

for i = 1:NN

    % extract 100 non-contaminated r.v.
    resam_contaimed_dist = randn(1,n);
```

```
    W = binornd(1,1 - eps,1,n); %bernoulli dist. is binomial dist.
       with n=1
    Y = mu + sqrt(var).*resam_contaimed_dist;
    Z = trnd(1,n,1).'; %student distr with parameter 1 is Cauchy
       distr.
    % extract a set of 100 r.v. from the set
    resam_norm_dist = W.*Y + (1-W).*Z; % make it a contaminated
       distribution

    median_norm(i)     = median(resam_norm_dist);
    median_X_test(i)   = median(resam_contaimed_dist);

    trimmean_norm(i)   = trimmean(resam_norm_dist, alpha*100);
    trimmean_X_test(i)= trimmean(resam_contaimed_dist, alpha*100);
end

figure
subplot(221)
histogram(median_norm)
title('The Histogram of the  median of normal distribution')
grid on

subplot(222)
histogram(trimmean_norm)
title('The Histogram of the  alpha-trimmed of normal distribution')
grid on

subplot(223)
histogram(median_X_test)
title('The Histogram of the  median of contaminated normal
   distribution')
grid on


subplot(224)
histogram(trimmean_X_test)
title('The Histogram of the  alpha-trimmed of contaminated normal
   distribution')
grid on


med_norm_sort = sort(median_norm);
med_X_test_sort = sort(median_X_test);

trimu_norm_sort = sort(trimmean_norm);
```

```
trimu_X_test_sort = sort(trimmean_X_test);

fprintf(...
    ['Ordering the median and the apha-trimmed mean values and given the values at the 25 and 975\n',...
    '\n The median of non-contaminated normal distribution at 25 = '...
    ,num2str(med_norm_sort(round(0.025*NN))), ' and at 975 = ', ...
    num2str(med_norm_sort(975)),...
    '.\n The median for eps-contaminated normal distribution at 25 = '...
    ,num2str(med_X_test_sort(round(0.025*NN))), ' and at 975 = ', ...
    num2str(med_X_test_sort(round(0.975*NN))),'\n'...
    '\n The apha-trimmed mean of non-contaminated normal distribution at 25 = '...
    ,num2str(trimu_norm_sort(round(0.0250*NN))), ' and at 975 = ', ...
    num2str(trimu_norm_sort(round(0.975*NN))),...
    '.\n The apha-trimmed mean for eps-contaminated normal distribution at 25 ='...
    ,num2str(trimu_X_test_sort(round(0.0250*NN))), ' and at 975 = ', ...
    num2str(trimu_X_test_sort(round(0.975*NN))),'\n'
    ])
```

Appendix - R code

```
######
## Assignment 1.1
rm(list = ls()) # Clear data
# Clear figures
graphics.off()

N <- 100 # Number of observation

# parameter for the gamma Dist.
alpha <- 2
beta <- 2
var_g <- alpha*beta^2;

gSamp <- rgamma(n=N, shape = alpha, scale = beta)

# create qq-plot
```

```r
qqnorm(gSamp, distribution = qnorm)
qqline(gSamp, col = "red", lwd = 3)

# Chi-square test
chig <- chisq.test(gSamp)
chig$p.value

# confidence interval
Rmisc::CI(gSamp, ci = 0.95)

######
## Assignment 1.2

# A function that calculates the CI for given data
CIinterval = function(x, conf.level = 0.95) {
  len = length(x) - 1
  lower <- qchisq((1 - conf.level )/2, len)
  upper <- qchisq((1 - conf.level )/2, len, lower.tail = FALSE)
  var_t <- var(x) #variance
  c(len*var_t/upper, len*var_t/lower) #calculate the confidence
     interval
}

NN <- 1000;
width = rep(0,NN)
counter = 0;
CI.vec = rep(0,NN)
width <- rep(0,1000)
counter <- 0;

for(i in 1:1000){
  gsample = rgamma(n=N, shape = alpha, scale = beta)
  CI.vec = CIinterval(gsample)
  width[i]= (CI.vec[2] - CI.vec[1]);
  if (var_g>=CI.vec[1] && var_g<=CI.vec[2]){
    #counting if the real sigma^2 is in the confidence interval
    counter = counter + 1;
  }
}
print(counter)
print(mean(width))

## Assignment 2.3
rm(list = ls()) # Clear data

```

```
eps <- 0.05;
mu <- 0;
var <- 1;
counter <- 0;
alpha <- 0.1;
N <- 100;
NN <- 1000;
k <- N*alpha;

triVec = rep(0,NN);
medVec = rep(0,NN);

for(i in 1:NN){
  Xalpha <- 0;

  Y <- rnorm(N, mu, var);
  W <- rbinom(N, 1, 1 - eps); #bernoulli distr. is binomial distr. with n=1
  Z <- rt(N, 1); #student distr with paramter 1 is Cauchy distr.
  X <- W*Y+(1-W)*Z;
  X <- sort(X);
  for(j in (k+1):(N - k)){
    Xalpha = Xalpha+X[j];
  }
  Xalpha = Xalpha/(N - 2*k);
  triVec[i] = Xalpha;
  medVec[i] = median(X);
}

triVec = sort(triVec); #sort m from small to large
medVec = sort(medVec); #sort med from small to large
CI_tri = c(triVec[25], triVec[975]);
CI_Med = c(medVec[25], medVec[965])
print(CI_tri)
print(CI_Med)
hist(triVec)
hist(medVec)
```