

# 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  $\sigma^2$ , 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 distributed data, 801 of them covered the true value of the variance  $\sigma^2$ , 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.

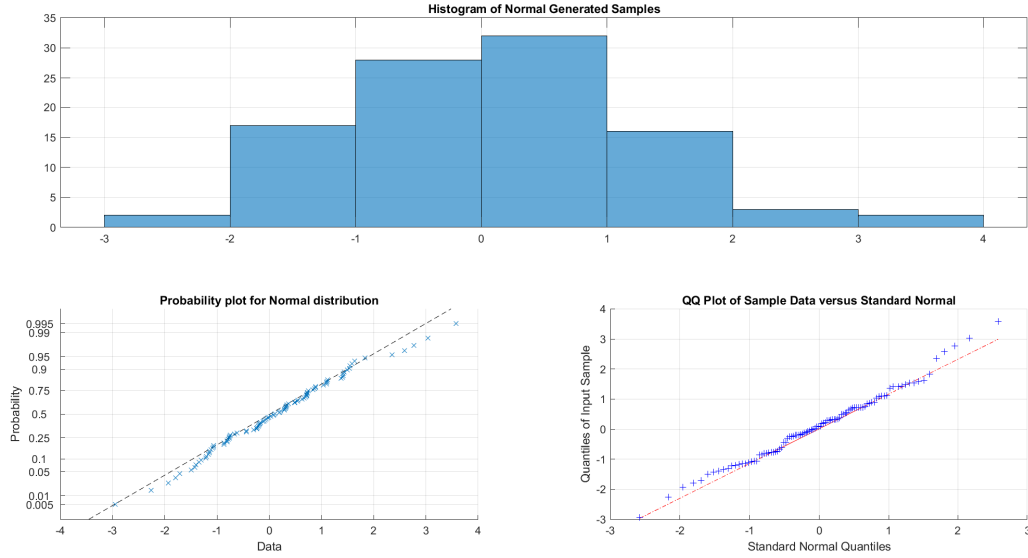


Figure 1: Histogram(top), pp-plot (bottom left) and qq-plot (bottom right) of the simulated normally distributed data

### 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  $\epsilon$ -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  $\epsilon$ -contaminated normal distribution at 25 is  $-0.2244$  and at 975 is  $0.25465$ . The  $\alpha$ -trimmed mean of non-contaminated normal distribution at 25 is  $-0.19648$  and at 975 is  $0.2034$ . The  $\alpha$ -trimmed mean for  $\epsilon$ -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  $0.2234931$ .

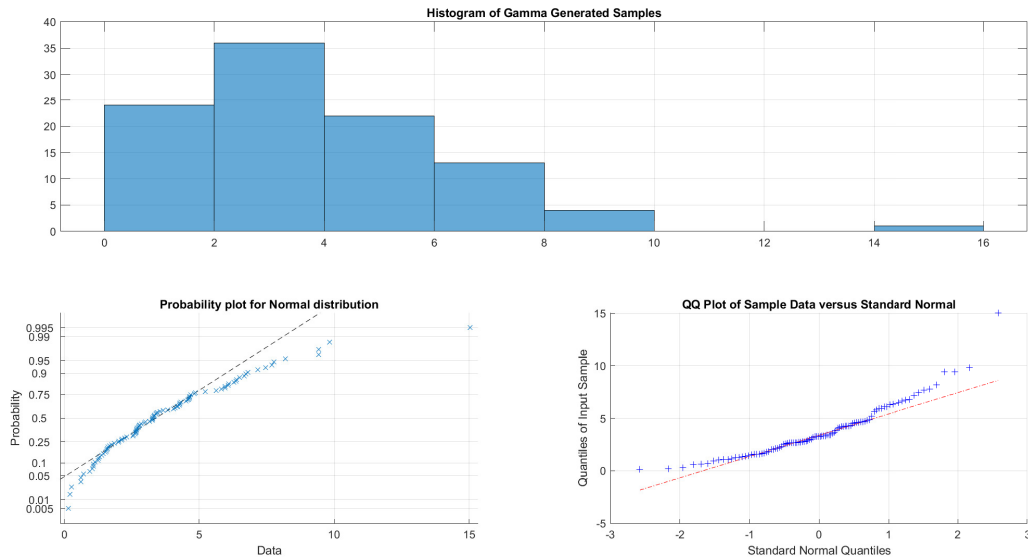


Figure 2: Histogram(top), pp-plot (bottom left) and qq-plot (bottom right) of the simulated gamma distributed data

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

```

1 clear all;close all;clc;
2 %% Assignment 1 Effect of Distribution Assumptions
3 %% Assignment 1.1 Creating the data
4 clear all;close all;clc;
5 % In this assignmetn we are generating data that are normally and
   gamma
6 % distrbutied data and see their histogram, pp-plot and qq-plot
7 % for their preceptive distribution. Also, we test the goodness-of-
   fit
8 % for each one of the function.
9 %%% Define the parameters for Normal
10 % rng default % for reproducibility of the results
11
12 n          = 100; % Number of samples
13 mu         = 0; % Mean value for the normal distribution
14 sigma      = 1; % Standard Deviation the normal distribution
15

```

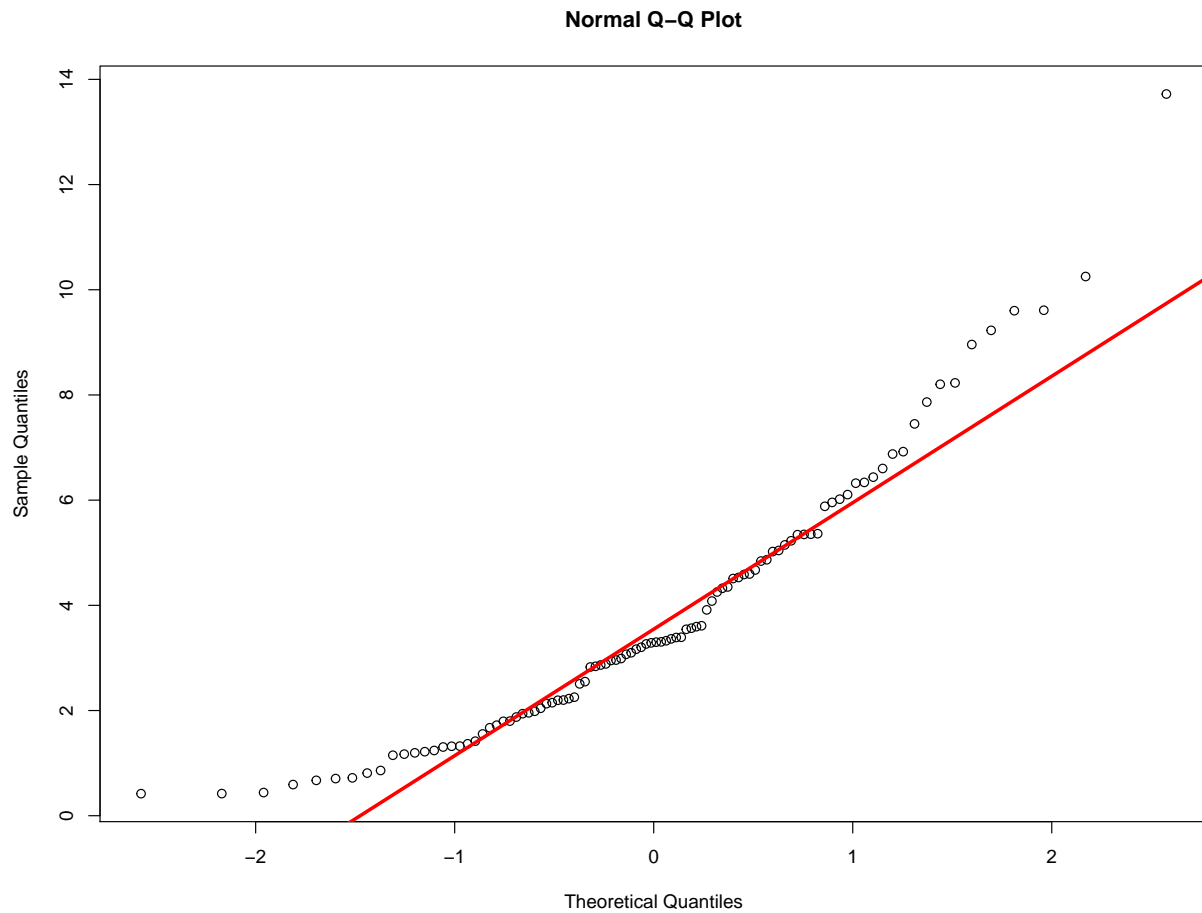


Figure 3: QQ-plot of the gamma distributed data

```

16 % define Normal probability distribution
17 npd      = makedist('Normal','mu',mu,'sigma',sigma);
18
19 % Generating a normal distrubtion samples with given parameters
    using ...
20 % a pre-defined function
21 nSamples  = normrnd(mu,sigma,1,n);
22 % nSamples  = mu + sigma.*randn(1,100);
23
24 % Normal Cumultive Distribution Function (cdf)
25 nSamplecdf = cdf(npd,nSamples);
26 % nSamplecdf  = normcdf(XSamples,mu,sigma);
27
28 %%% Ploting of Histogram, pp-plot and qq-plot Normal Distribution
29 % plot the histogram of the data

```

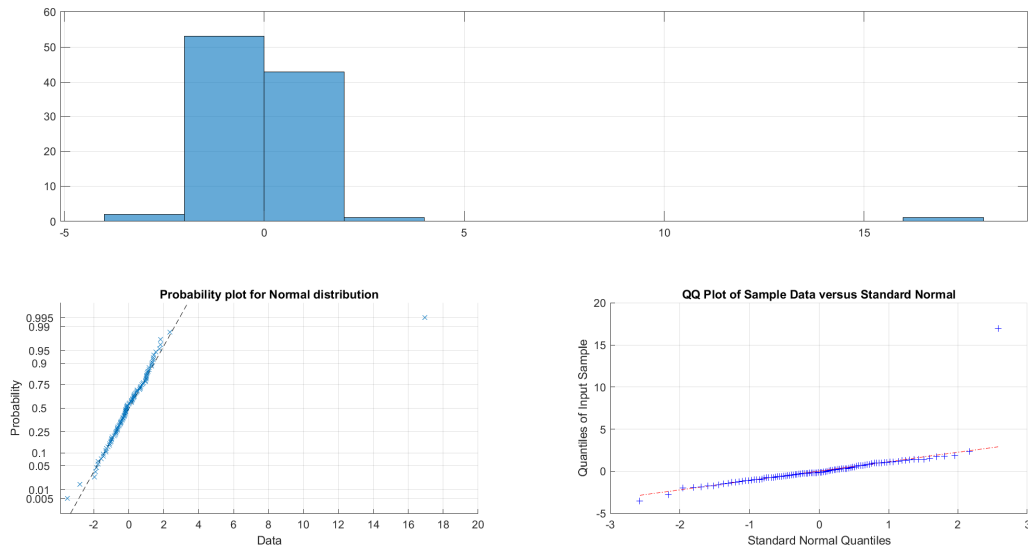


Figure 4: Histogram(top), pp-plot (bottom left) and qq-plot (bottom right) of the simulated contaminated data

```

30 subplot(211)
31 histogram(nSamples)
32 %histfit(nSamples)
33 title('Histogram of Normal Generated Samples')
34 grid on
35 hold on
36
37 % Plotting the PP-plot
38 subplot(223)
39 probplot(nSamples)
40 grid on
41
42 % Plotting the QQ-plot
43 subplot(224)
44 qqplot(nSamples)
45 grid on
46
47 %%% Goodness-of-fit Test for Normal Distribution
48 % Chi-square test
49 [ncqh,ncqp_value] = chi2gof(nSamples);
50 disp('-----Chi2test Normal-----')
51 x= 'The decision for Goodness-of-fit test using Chi-square is %d.
    for Normal Distribution\n';
52 fprintf(x,ncqh)

```

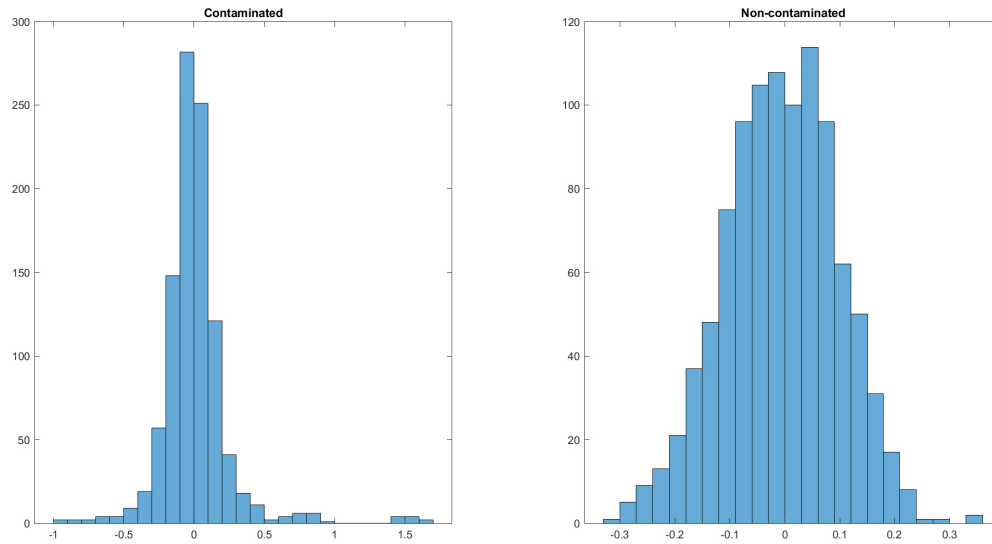


Figure 5: Histograms for the contaminated (left) and non-contaminated (right) distributions

```

53
54 if ncqh == 0
55     X = ['\n Since the test decision for the null hypothesis of the
56         data = ', num2str(ncqh), ...
57         '\n Then, we do not reject the null hypothesis \n that the
58         samples follows the Normal Distribution.\n'];
59     fprintf(X)
60 else
61     X = ['\n Since the test decision for the null hypothesis of the
62         data = ', num2str(ncqh), ...
63         '\n Then, we reject the null hypothesis \n that the samples
64         follows the Normal Distribution.\n'];
65     fprintf(X)
66 end
67
68 disp('-----')
69
70 % Kolmogorov-Smirnov test
71 % If h = 1, this indicates the rejection of the null hypothesis.
72 % If h = 0, this indicates a failure to reject the null hypothesis.
73 [nkh, nkp_value] = kstest(nSamples);
74 disp('-----KStest Normal-----')
75 x = 'The decision for Goodness-of-fit test using Kolmogorov-Smirnov
76     is %d. for Normal Distribution\n';
77 fprintf(x, nkh)

```

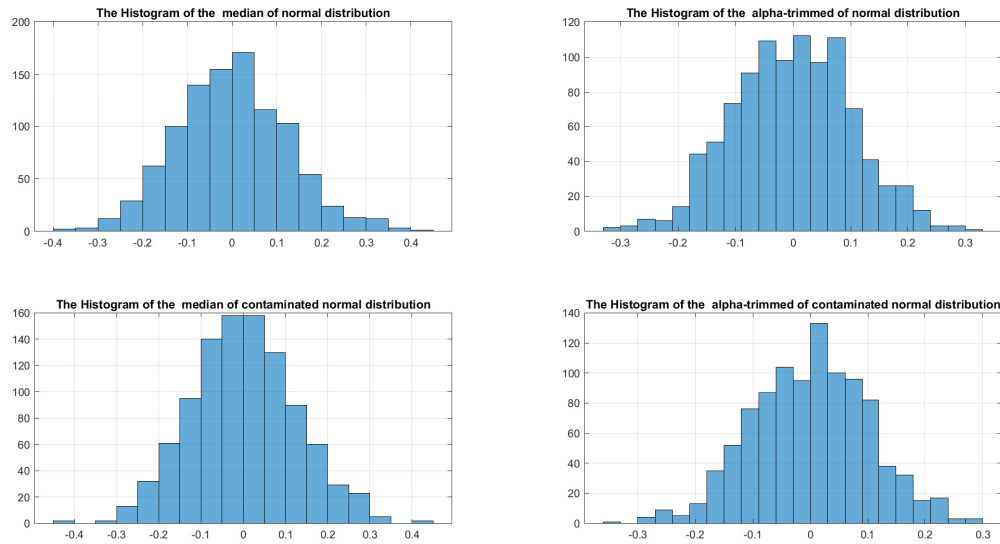


Figure 6: Histograms for the median and  $\alpha$  – trimmed mean for a sample of size 1000

```

73
74 if nkh == 0
75     X = ['\n Since the test decision for the null hypothesis of the
76         data = ', num2str(nkh), ...
77         '\n Then, we failed to reject the null hypothesis \n that
78         the samples follows the Normal Distribution.\n'];
79     fprintf(X)
80 else
81     X = ['\n Since the test decision for the null hypothesis of the
82         data = ', num2str(nkh), ...
83         '\n Then, we reject the null hypothesis \n that the samples
84         follows the Normal Distribution.\n'];
85     fprintf(X)
86 end
87
88 disp('-----')
89
90 %% Define the parameters Gamma
91 alpha = 2; % shape parameter for gamma distribution
92 beta = 2; % scale parameter for gamma distribution
93
94 % define Gamma probability distribution
95 gpd = makedist('Gamma','a',alpha,'b',beta);
96

```

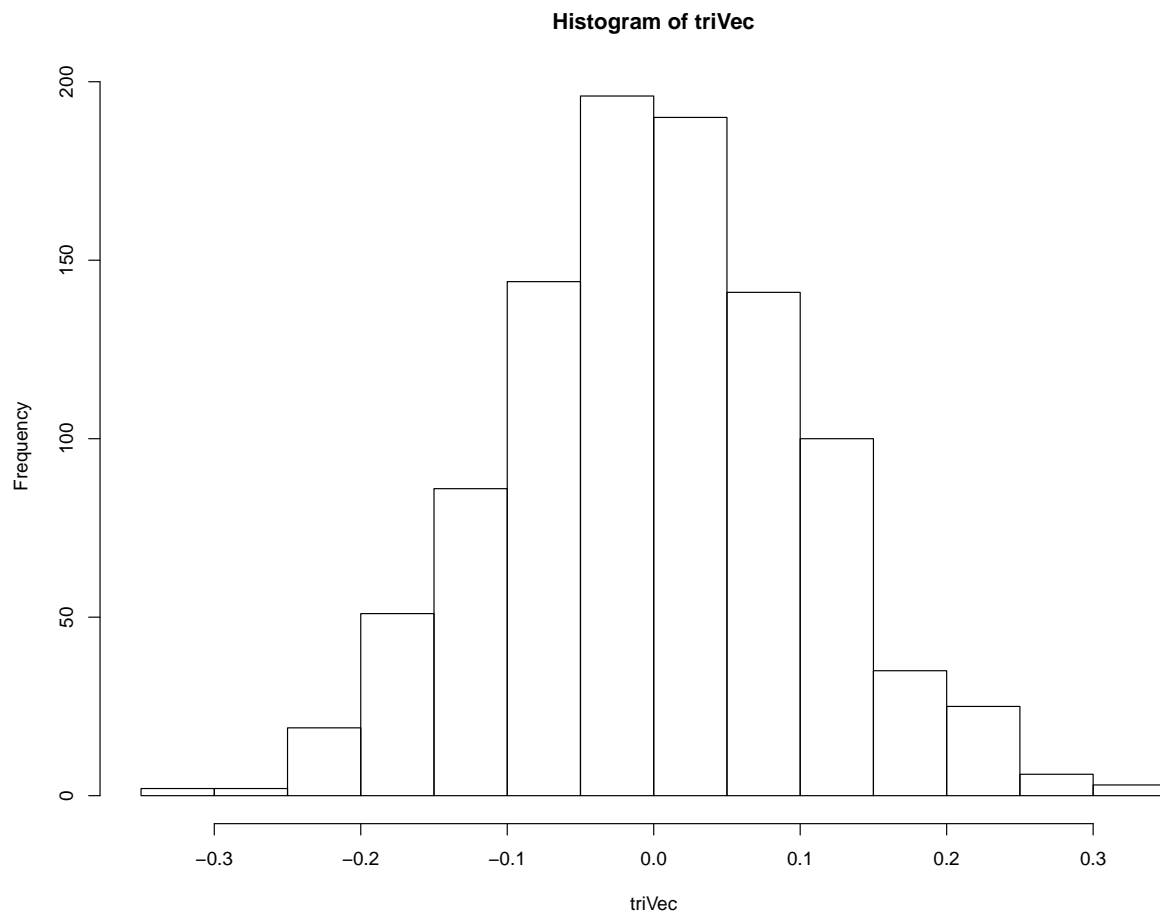


Figure 7: Histogram of the  $\alpha$  – trimmed mean for a sample of size 1000

```

93 % Generating a gamma distribution samples with given parameters
    using ...
94 % a pre-defined function
95 gSamples      = gamrnd(alpha,beta,1,n);
96
97 %%% Ploting of Histogram, pp-plot and qq-plot Gamma Distribution
98 figure
99 % plot the histogram of the data
100 subplot(211)
101 histogram(gSamples)
102 title('Histogram of Gamma Generated Samples')
103 grid on
104 hold on
105
106 % Ploting the PP-plot

```



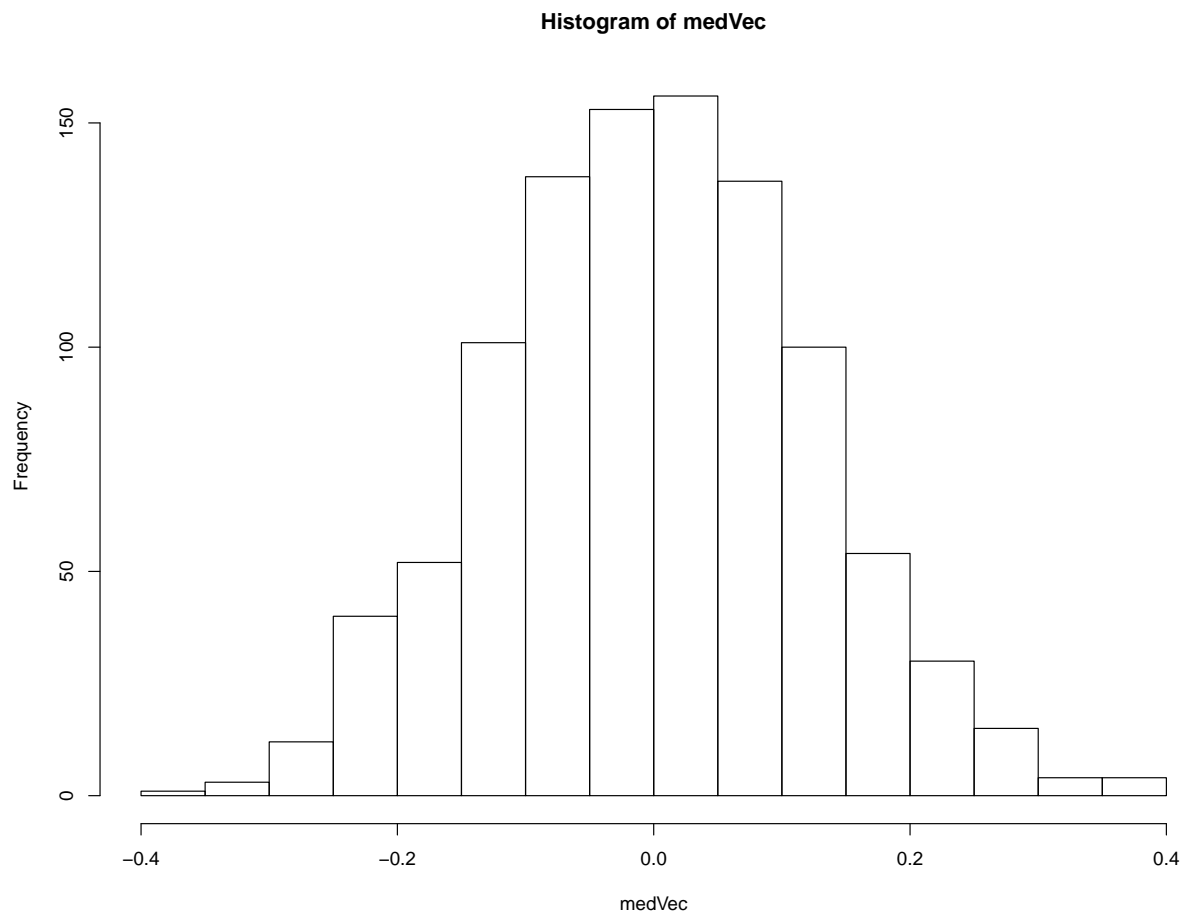


Figure 8: Histogram of the median for a sample of size 1000

```

107 subplot(223)
108 probplot(gSamples)
109 grid on
110
111 % Plotting the QQ-plot
112 subplot(224)
113 qqplot(gSamples)
114 grid on
115
116 %%% Goodness-of-fit Test for Gamma Distribution
117 % Chi-square test
118
119 % [gh,gp_value] = chi2gof(gSamples,'CDF', gpd);
120 [gh,gp_value] = chi2gof(gSamples);
121 disp('-----Chi2test Gamma-----')
```

```

122 x = 'The decision for Goodness-of-fit test using Chi-square is %d.
      for Gamma Distribution\n';
123 fprintf(x,gh)
124
125 if gh == 0
126     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'];
128     fprintf(X)
129 else
130     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'];
132     fprintf(X)
133 end
134
135 disp('-----KStest Gamma-----')
136
137 % Kolmogorov-Smirnov test
138 % If h = 1, this indicates the rejection of the null hypothesis.
139 % If h = 0, this indicates a failure to reject the null hypothesis.
140 [gkh,gkp_value] = kstest(gSamples);
141
142 x= 'The decision for Goodness-of-fit test using Kolmogorov-Smirnov
      is %d, for Gamma Distribution.\n';
143 fprintf(x,gkh)
144
145 if gkh == 0
146     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'];
148     fprintf(X)
149 else
150     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'];
152     fprintf(X)
153 end
154
155 %% Assignment 1.2
156 disp('-----A1.2-----')

```

```

157 % rng default % for reproducibility of the results
158 n           = 100; % Number of samples
159 mu          = 0; % Mean value for the normal distribution
160 sigma       = 1; % Standard Deviation the normal distribution
161 ncount      = 0; % counter
162 NN          = 1000; % testing cycle
163 width_CI    = zeros(1,NN); % width of the intervals
164 norm_CI     = zeros(2,NN); % define size of variable
165
166 % Generating a normal distrubtion samples with given parameters
    using ...
167 % a pre-defined function
168 for i = 1:NN
169     nSam_test      = normrnd(mu,sigma,1,n);
170     [~,~,~, sigmaCI] = normfit(nSam_test); % calculate CI for sigma
171     norm_CI(:,i) = sigmaCI;
172     width_CI(:,i) = diff(sigmaCI);
173     if (norm_CI(1,i) <= sigma^2 && norm_CI(2,i) >= sigma^2)
174         ncount = ncount + 1;
175     end
176 end
177 mean_width_of_intervals = mean(width_CI); % the mean of the width of
    intervals
178 disp(['Number of intervals which contain the true variance is ',...
179     num2str(ncount), ' and the average width of the intervals is '
180     ,...
181     num2str(mean_width_of_intervals)])
182 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
183 disp(['-----Gamma Dist-----'])
184 alpha      = 2; % shape parameter for gamma distribution
185 beta       = 2; % scale parameter for gamma distribution
186 gsigma_sq  = alpha*beta^2; % variance for the gamma
187 gcount     = 0; % Counter
188 width_gCI  = zeros(1,NN); % width of the intervals
189 gci        = zeros(2,NN); % define size of variable
190
191 % Generating a gamma distribution samples with given parameters
    using ...
192 % a pre-defined function
193 for i = 1:NN
194     gSam_test      = gamrnd(alpha,beta,1,n);
195     [~,~,~, gsigmaCI] = normfit(gSam_test); % calculate CI for sigma
196     gci(:,i) = gsigmaCI.^2;
197     width_gCI(i) = diff(gsigmaCI);

```

```

198     if (gci(1,i) <= gsigma_sq && gci(2,i) >= gsigma_sq)
199         gcount = gcount + 1;
200     end
201 end
202 mean_width_of_intervals_g = mean(width_gCI);
203 disp(['Number of intervals which contain the true variance is ',...
204     num2str(gcount), ' and the average width of the intervals is '
205     ,...
206     num2str(mean_width_of_intervals_g)])
207 %% Assignment 2 Robust Estimation
208 disp('-----A2-----')
209 %% Assignment 2.1
210 disp('-----A2.1-----')
211 %%% Define the parameters
212 n = 100; % number of Samples
213 mu = 0;
214 var = 1;
215 eps = 0.05;
216 % nSamp = normrnd(mu_y,sqrt(var_y),1,n);
217 nSamp = randn(1,n);
218 W = binornd(1,1 - eps,1,n); %bernoulli distr. is binomial distr .
219     with n=1
220 Y = mu + sqrt(var).*nSamp;
221 Z = trnd(1,n,1).'; %student distr with parameter 1 is Cauchy distr.
222 contained_dist = W.*Y + (1-W).*Z; % make it a contaminated
223     distribution
224 %%% Ploting of Histogram, pp-plot and qq-plot contaminated
225     Distribution
226 figure
227 % plot the histogram of the data
228 subplot(211)
229 histogram(contained_dist)
230 grid on
231 % Ploting the PP-plot
232 subplot(223)
233 probplot(contained_dist)
234 grid on
235 % Ploting the QQ-plot
236 subplot(224)
237 qqplot(contained_dist)
238 grid on
239 [csqh,csqp_value] = chi2gof(contained_dist);

```

```

239 disp('-----Chi2test-----')
240 x = 'The decision for Goodness-of-fit test using Chi-square is %d.
      for contaminated Distribution\n';
241 fprintf(x,csqh)
242
243 if csqh == 0
244     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'];
246     fprintf(X)
247 else
248     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'];
249     fprintf(X)
250
251 end
252 %%%%%%%%%%%%%%%
253 %% Assignment 2.2
254 disp('-----A2.2-----')
255 %%% Define the parameters
256 NN      = 1000; % reptation number
257
258 resam_contaimed_dist = zeros(1,n);
259 resam_norm_dist = zeros(1,n);
260
261 mu_cd      = zeros(1,NN);
262 mu_norm     = zeros(1,NN);
263
264 for i = 1:NN
265     % extract 100 non-contaminated r.v.
266     resam_contaimed_dist = randn(1,n);
267     W = binornd(1,1 - eps,1,n); %bernoulli dist. is binomial dist.
        with n=1
268     Y = mu + sqrt(var).*resam_contaimed_dist;
269     Z = trnd(1,n,1).'; %student distr with parameter 1 is Cauchy
        distr.
270     % extract a set of 100 r.v. from the set
271     resam_norm_dist = W.*Y + (1-W).*Z; % make it a contaminated
        distribution
272
273     mu_norm(i)      = mean(resam_norm_dist);
274     mu_cd(i)       = mean(resam_contaimed_dist);
275 end

```

```

276
277 figure
278 subplot(211)
279 histogram(mu_norm)
280 title('The Histogram of the means of normal distribution')
281 grid on
282
283 subplot(212)
284 histogram(mu_cd)
285 title('The Histogram of the means of eps-contaminated normal
      distribution')
286 grid on
287
288 mu_norm_sort = sort(mu_norm);
289 mu_X_test_sort = sort(mu_cd);
290
291 fprintf(...
292     ['Ordering the means values and given the values at the 25 and
      975\n',...
293     '\n The mean of non-contaminated normal distribution at 25 = '
      ...
294     ,num2str(mu_norm_sort(round(0.025*NN))), ' and at 975 = ', ...
295     num2str(mu_norm_sort(round(0.975*NN))),...
296     '.\n The mean for eps-contaminated normal distribution at 25 ='
      ...
297     ,num2str(mu_X_test_sort(round(0.025*NN))), ' and at 975 = ', ...
298     num2str(mu_X_test_sort(round(0.975*NN))),'\n']]
299
300 %%%%%%
301 %% Assignment 2.3
302 disp('-----A2.3-----')
303 %%% Define the parameters
304 alpha      = 0.1;
305
306 median_X_test      = zeros(1,NN);
307 median_norm        = zeros(1,NN);
308
309 trimmean_X_test    = zeros(1,NN);
310 trimmean_norm      = zeros(1,NN);
311
312 for i = 1:NN
313
314     % extract 100 non-contaminated r.v.
315     resam_contained_dist = randn(1,n);

```

```

316     W = binornd(1,1 - eps,1,n); %bernoulli dist. is binomial dist.
        with n=1
317     Y = mu + sqrt(var).*resam_contaimed_dist;
318     Z = trnd(1,n,1).'; %student distr with parameter 1 is Cauchy
        distr.
319     % extract a set of 100 r.v. from the set
320     resam_norm_dist = W.*Y + (1-W).*Z; % make it a contaminated
        distribution
321
322     median_norm(i)      = median(resam_norm_dist);
323     median_X_test(i)    = median(resam_contaimed_dist);
324
325     trimmean_norm(i)    = trimmean(resam_norm_dist, alpha*100);
326     trimmean_X_test(i)= trimmean(resam_contaimed_dist, alpha*100);
327 end
328
329 figure
330 subplot(221)
331 histogram(median_norm)
332 title('The Histogram of the median of normal distribution')
333 grid on
334
335 subplot(222)
336 histogram(trimmean_norm)
337 title('The Histogram of the alpha-trimmed of normal distribution')
338 grid on
339
340 subplot(223)
341 histogram(median_X_test)
342 title('The Histogram of the median of contaminated normal
        distribution')
343 grid on
344
345
346 subplot(224)
347 histogram(trimmean_X_test)
348 title('The Histogram of the alpha-trimmed of contaminated normal
        distribution')
349 grid on
350
351
352 med_norm_sort = sort(median_norm);
353 med_X_test_sort = sort(median_X_test);
354
355 trimu_norm_sort = sort(trimmean_norm);

```

```

356 trimu_X_test_sort = sort(trimmean_X_test);
357
358 fprintf(...
359     ['Ordering the median and the apha-trimmed mean values and given
      the values at the 25 and 975\n',...
360     '\n The median of non-contaminated normal distribution at 25 = '
      ...
361     ,num2str(med_norm_sort(round(0.025*NN))), ' and at 975 = ', ...
362     num2str(med_norm_sort(975)),...
363     '.\n The median for eps-contaminated normal distribution at 25 =
      '...
364     ,num2str(med_X_test_sort(round(0.025*NN))), ' and at 975 = ',
      ...
365     num2str(med_X_test_sort(round(0.975*NN))),'\n'...
366     '\n The apha-trimmed mean of non-contaminated normal
      distribution at 25 = '...
367     ,num2str(trimu_norm_sort(round(0.0250*NN))), ' and at 975 = ',
      ...
368     num2str(trimu_norm_sort(round(0.975*NN))),...
369     '.\n The apha-trimmed mean for eps-contaminated normal
      distribution at 25 ='...
370     ,num2str(trimu_X_test_sort(round(0.0250*NN))), ' and at 975 = ',
      ...
371     num2str(trimu_X_test_sort(round(0.975*NN))),'\n'
372     ])

```

## Appendix - R code

```

1 #####
2 ## Assignment 1.1
3 rm(list = ls()) # Clear data
4 # Clear figures
5 graphics.off()
6
7 N <- 100 # Number of observation
8
9 # parameter for the gamma Dist.
10 alpha <- 2
11 beta <- 2
12 var_g <- alpha*beta^2;
13
14 gSamp <- rgamma(n=N, shape = alpha, scale = beta)
15
16 # create qq-plot

```



```

17 qqnorm(gSamp, distribution = qnorm)
18 qqline(gSamp, col = "red", lwd = 3)
19
20 # Chi-square test
21 chig <- chisq.test(gSamp)
22 chig$p.value
23
24 # confidence interval
25 Rmisc::CI(gSamp, ci = 0.95)
26
27 #####
28 ## Assignment 1.2
29
30 # A function that calculates the CI for given data
31 CIinterval = function(x, conf.level = 0.95) {
32   len = length(x) - 1
33   lower <- qchisq((1 - conf.level)/2, len)
34   upper <- qchisq((1 - conf.level)/2, len, lower.tail = FALSE)
35   var_t <- var(x) #variance
36   c(len*var_t/upper, len*var_t/lower) #calculate the confidence
      interval
37 }
38
39 NN <- 1000;
40 width = rep(0, NN)
41 counter = 0;
42 CI.vec = rep(0, NN)
43 width <- rep(0, 1000)
44 counter <- 0;
45
46 for(i in 1:1000){
47   gsample = rgamma(n=N, shape = alpha, scale = beta)
48   CI.vec = CIinterval(gsample)
49   width[i]= (CI.vec[2] - CI.vec[1]);
50   if (var_g >= CI.vec[1] && var_g <= CI.vec[2]){
51     #counting if the real sigma^2 is in the confidence interval
52     counter = counter + 1;
53   }
54 }
55 print(counter)
56 print(mean(width))
57
58 ## Assignment 2.3
59 rm(list = ls()) # Clear data
60

```

```

61 eps <- 0.05;
62 mu <- 0;
63 var <- 1;
64 counter <- 0;
65 alpha <- 0.1;
66 N <- 100;
67 NN <- 1000;
68 k <- N*alpha;
69
70 triVec = rep(0,NN);
71 medVec = rep(0,NN);
72
73 for(i in 1:NN){
74   Xalpha <- 0;
75
76   Y <- rnorm(N, mu, var);
77   W <- rbinom(N, 1, 1 - eps); #bernoulli distr. is binomial distr.
       with n=1
78   Z <- rt(N, 1); #student distr with paramter 1 is Cauchy distr.
79   X <- W*Y+(1-W)*Z;
80   X <- sort(X);
81   for(j in (k+1):(N - k)){
82     Xalpha = Xalpha+X[j];
83   }
84   Xalpha = Xalpha/(N - 2*k);
85   triVec[i] = Xalpha;
86   medVec[i] = median(X);
87 }
88
89 triVec = sort(triVec); #sort m from small to large
90 medVec = sort(medVec); #sort med from small to large
91 CI_tri = c(triVec[25], triVec[975]);
92 CI_Med = c(medVec[25], medVec[965])
93 print(CI_tri)
94 print(CI_Med)
95 hist(triVec)
96 hist(medVec)

```