Report

Zhang Weini
Student Number: 23020181154257

I. Assignment 1 (2019-07-15)

A. Read, summarize and comment the article

*** Relevance**

The main purpose of this paper is to test the computational accuracy of some numerical scientific platforms, under the diversity of operational systems.

Assessing the accuracy of these platforms is important. In many fields, to find best approximate solutions to the problem, we have high requirements on computational accuracy. Especially when solving complex models with many variables and values, minute errors in each step may have great effect on the final results.

The authors think previous studies were limited to simply comparing results with certified values provided by Statistical Reference Datasets. Thus, based on this method, they design their experiments to test the accuracy of these platforms.

*** Methodology**

To assess the accuracy of *Octave 3.2.4*, *Scilab 5.3* and *Matlab R2011a* platforms, under three operating systems (*Windows*, *Ubuntu* and *Mac OS*) and the hardware(*i386 architecture*), the authors conduct two groups of experiments, in which they compare the results computed by the platforms with certified values:

- *Statistical description tests (certified values provided by NIST)*
  1) Basic statistics
     Assessment: (1)mean, (2)standard deviation (3)first lag coefficient of autocorrelation
     Measurement: LRE + the bootstrap estimates of the standard deviation of LREs
     Using Monte Carlo study
  2) Probability distribution functions
     Assessment: (1)binomial, (2)Poisson, (3)gamma, (4)normal, (5)χ^2, (6)beta, (7)t-Student, (8)F
     Measurement: LRE
  3) Linear regression
     Measurement: the smallest LRE + the LRE of RSD
- *Matrices operations tests*
  1) Determinant computation
     Assessment: Compute the determinant of ill-conditioned 2×2 matrices.
     Measurement: Logical comparesion of computed value and certified value 0.
  2) Spectral graph analysis
     Compute the eigenvalues of the Laplacian matrix of a bipartite graph.
     Measurement: LREs and percentage of some values associated with eigenvalues

*** Results and conclusions**

In this paper, the authors show the results of their experiments in several tables. From the results they have the following conclusion:

1) Basic statistics
   - *Mean*: Octave for Linux presents relatively low accuracy, other platforms perform well on the datasets.
   - *Standard deviation*: Octave presents the best results while Scilab presents an unacceptable low accuracy in a single dataset.
   - *First-lag sample autocorrelation*: none of the platforms provides acceptable results. Scilab has the worst performance.
   - *Stability*: Scilab is the worst platform. Octave for Linux is also instable when computing the sample mean and the sample standard deviation.
2) Probability distribution functions
   - *Binomial and t-Student distributions*: Scilab presents the best performance. Octave provides unacceptable results. Matlab and Octave fails at computing the t-Student distribution.
   - *Poisson law*: Scilab presents the best performance when computing the cumulative distribution function.
   - *Gamma law*: All the three platforms are acceptable.
   - *F distribution*: Octave presents the best performance.

- *Normal distribution*: MatLab and Octave provides the same good results, while Scilab provides bad results.

3) Linear regression
   - No single platform is credible for the linear regression problems.
4) Determinant computation
   - All the three platforms provide the same acceptable results.
5) Spectral graph analysis
   - The bigger the graph, the worse the results.
   - Double-precision computation is recommended. When comparing to known values, use floating point representation at most.
   - Variability: MatLab and Octave are equivalent and more consistent than Scilab.

*** Comments**

First, the main highlights of this paper include: (1) Using Monte Carlo studies to assess the stability of the results with respect to small departures from the original input. (2) Besides statistical description tests, the authors assess a set of matrix operations.

Second, I doubt whether the experiments in this paper can objectively evaluate the accuracy of the platforms. The datasets used in the experiments are limited. I'm not sure if the performance of these platforms is consistent with the conclusions of this paper when using other datasets.

Third, as mentioned in this paper, their results are not consistent with some previous work. I wonder what causes the inconsistency of the results. Is it because the experimental scheme has some problems or the platforms are instable? If the instability of these platforms affects the experimental results, are the results in this paper credible?

Finally, this paper can provide a reference for me when I need to choose a high precision and stable numerical scientific platform to solve problems.

II. Assignment 2 (2019-07-16)

A. Exercise 4.4: Obtain the expression of the density of $\mathcal{K}$-distributed amplitude data via the transformation $Z_A = \sqrt{Z_I}$. Compute its moments. Illustrate.

For $Z_I \sim \mathcal{K}(\alpha, \lambda, L)$, the density of Z_I is:

$$f_{Z_I}(z_I; \alpha, \lambda, L) = \frac{2\lambda L}{\Gamma(\alpha)\Gamma(L)}(\lambda L z_I)^{\frac{\alpha+L}{2}-1} k_{\alpha-L}(2\sqrt{\lambda L z_I}) \tag{1}$$

The density of amplitude data $Z_A = g(Z_I) = \sqrt{Z_I}$ is given by:

$$\begin{aligned} f_{Z_A}(z_A; \alpha, \lambda, L) &= \frac{2\lambda L}{\Gamma(\alpha)\Gamma(L)}(g^{-1}(z_A))^{'}(\lambda L g^{-1}(z_A))^{\frac{\alpha+L}{2}-1} k_{\alpha-L}(2\sqrt{\lambda L g^{-1}(z_A)}) \\ &= \frac{2\lambda L}{\Gamma(\alpha)\Gamma(L)} 2z_A(\lambda L z_A^2)^{\frac{\alpha+L}{2}-1} k_{\alpha-L}(2\sqrt{\lambda L z_A^2}) \\ &= \frac{4\lambda L}{\Gamma(\alpha)\Gamma(L)}(\lambda L)^{\frac{\alpha+L}{2}-1} z_A^{\alpha+L-1} k_{\alpha-L}(2\sqrt{\lambda L z_A^2}) \end{aligned} \tag{2}$$

The k-order moments of Z_A are:

$$\begin{aligned} E(Z_A^k) &= E(Z_I^{\frac{k}{2}}) \\ &= (\lambda L)^{-\frac{k}{2}} \frac{\Gamma(L+\frac{k}{2})\Gamma(\alpha+\frac{k}{2})}{\Gamma(L)\Gamma(\alpha)} \end{aligned} \tag{3}$$

B. Exercise 4.8: Obtain the expression of the density of $\mathcal{G}^0$-distributed amplitude data via the transformation $Z_A = \sqrt{Z_I}$. Compute its moments. Illustrate.

For $Z_I \sim \mathcal{G}^0(\alpha, \gamma, L)$, the density of Z_I is:

$$f_{Z_I}(z_I; \alpha, \gamma, L) = \frac{L^L\Gamma(L-\alpha)}{\gamma^\alpha\Gamma(L)\Gamma(-\alpha)} \frac{z_I^{L-1}}{(\gamma + L z_I)^{L-\alpha}} \tag{4}$$

The density of amplitude data $Z_A = g(Z_I) = \sqrt{Z_I}$ is given by:

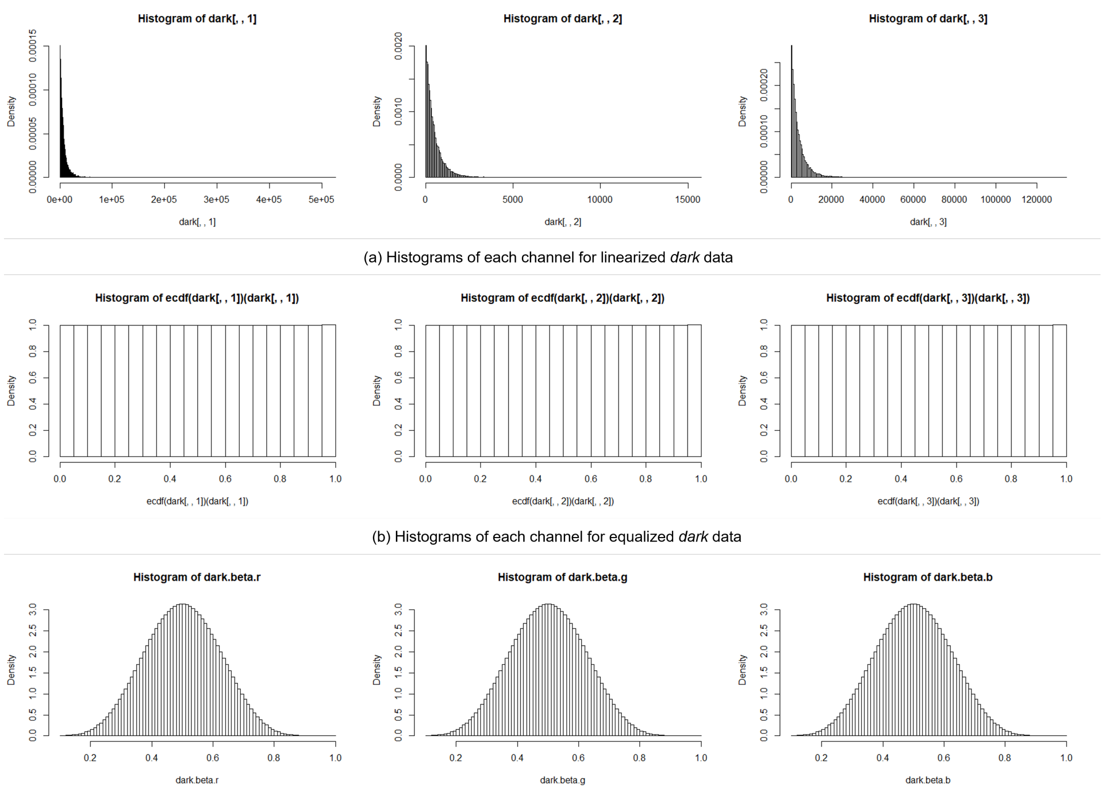
$$\begin{aligned} f_{Z_A}(z_A;\alpha,\gamma,L) &= \frac{L^L\Gamma(L-\alpha)}{\gamma^\alpha\Gamma(L)\Gamma(-\alpha)}(g^{-1}(z_A))'\frac{(g^{-1}(z_A))^{L-1}}{(\gamma+Lg^{-1}(z_A))^{L-\alpha}} \\ &= \frac{L^L\Gamma(L-\alpha)}{\gamma^\alpha\Gamma(L)\Gamma(-\alpha)}2z_A\frac{(z_A^2)^{L-1}}{(\gamma+Lz_A^2)^{L-\alpha}} \\ &= \frac{2L^L\Gamma(L-\alpha)}{\gamma^\alpha\Gamma(L)\Gamma(-\alpha)}\frac{z_A^{2L-1}}{(\gamma+Lz_A^2)^{L-\alpha}} \end{aligned} \tag{5}$$

The k-order moments of Z_A are:

$$\begin{aligned} E(Z_A^k) &= E(Z_I^{\frac{k}{2}}) \\ &= (\gamma/L)^{\frac{k}{2}}\frac{\Gamma(L+\frac{k}{2})\Gamma(-\alpha-\frac{k}{2})}{\Gamma(L)\Gamma(-\alpha)} \end{aligned} \tag{6}$$

C. Assign Beta histograms to the channels of both the dark and bright image to obtain different visualizations.

First, I equalize each channel of *dark* and *bright* data, and also assign Beta histograms to each channel. Fig.2 is the visualization of *dark* data. Fig.2(a), 2(b) and 2(c) show the linearization, equalization and $\mathcal{B}(8,8)$ stipulation of *dark* data, respectively. Because the data contains a very broad range and most of the values are clustered in low interval(shown in Fig.1), little is visible in Fig.2(a), and Fig. 2(b) looks exaggerated. Compared with linearized data and equalized data, the data with $\mathcal{B}(8,8)$ histograms is more visible and less exaggerated.

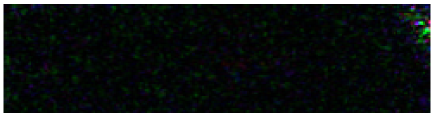


(a) Histograms of each channel for linearized *dark* data

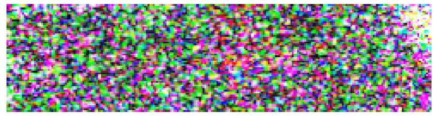
(b) Histograms of each channel for equalized *dark* data

(c) Histograms of each channel for *dark* data with Beta(8,8) histograms

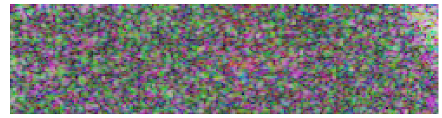
Fig. 1: Histograms of each channel for Linearized, equalized and $\mathcal{B}(8,8)$ stipulated *dark* data



(a) Linearized *dark* data

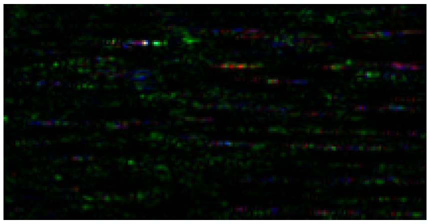


(b) Equalized *dark* data

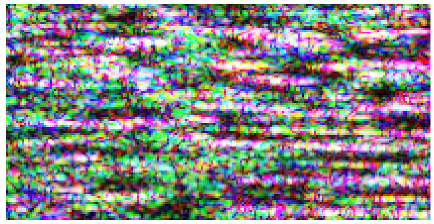


(c) *dark* data with Beta histograms

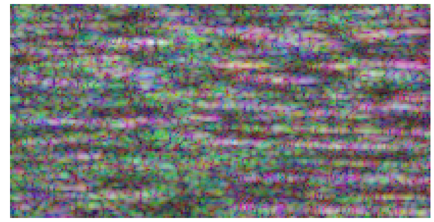
Fig. 2: Linearization, equalization and $\mathcal{B}(8,8)$ histogram stipulation of *dark* data



(a) Linearized *bright* data



(b) Equalized *bright* data



(c) *bright* data with Beta histograms

Fig. 3: Linearization, equalization and $\mathcal{B}(8,8)$ histogram stipulation of *bright* data

Similar to Fig.2, Fig.3 is the visualization of *bright* data. For the sake of simplicity, I only show the codes(in Listing 1) corresponding to the *dark* data.

```
# [dark] Linearized
plot(imagematrix(normalize_indep(dark)))

# [dark] Equalized
dark.equalize_indep <- c(ecdf(dark[,,1])(dark[,,1]),ecdf(dark[,,2])(dark[,,2]),ecdf(
    dark[,,3])(dark[,,3]))
plot(imagematrix(normalize_indep(array(dark.equalize_indep, dim=dim(dark)))))

# [dark] ~beta(8,8)
dark.beta.r <- qbeta(ecdf(dark[,,1])(dark[,,1]), shape1 = 8, shape2 = 8)
dark.beta.g <- qbeta(ecdf(dark[,,2])(dark[,,2]), shape1 = 8, shape2 = 8)
dark.beta.b <- qbeta(ecdf(dark[,,3])(dark[,,3]), shape1 = 8, shape2 = 8)
dark.beta_indep <- c(dark.beta.r,dark.beta.g,dark.beta.b)
plot(imagematrix((normalize_indep(array(dark.beta_indep, dim=dim(dark))))))
```

Listing 1: Linearize, equalize and assign $\mathcal{B}(8,8)$ histogram to *dark* data

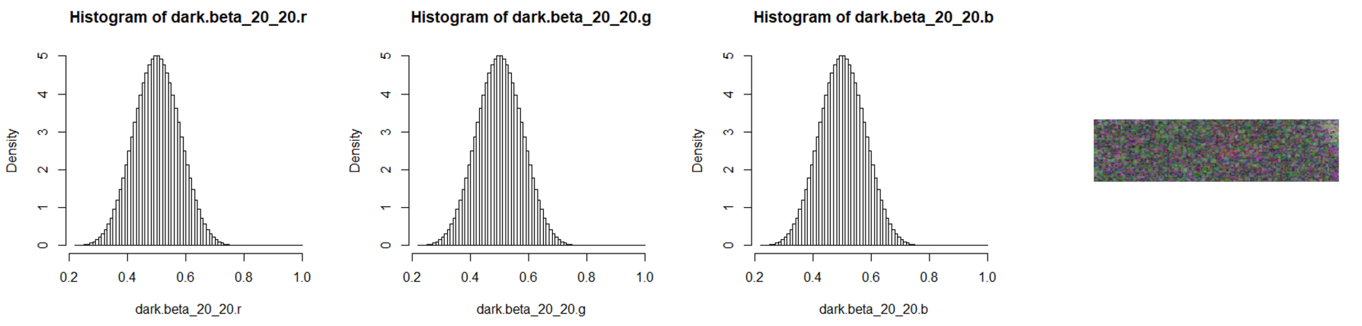
Then I try to specify the histogram as Beta distributions with different parameters. Fig.4 shows the histograms of each channel and the visualization of data specified as $\mathcal{B}(2,2), \mathcal{B}(8,8), \mathcal{B}(20,20)$ with the same mean. From histograms in Fig.4 we can see that the data specified as $\mathcal{B}(20,20)$ has more values clustered in a narrow range. Thus, the image looks smoother with lower contrast. While the data specified as $\mathcal{B}(2,2)$ is relatively evenly distributed, and the image has higher contrast. Fig.5 shows the histograms and the visualization of data specified as $\mathcal{B}(20,4), \mathcal{B}(4,20)$. From histograms in Fig.5 we can see that the data specified as $\mathcal{B}(20,4)$ has more values clustered in high interval while $\mathcal{B}(4,20)$ has more values clustered in low interval. Thus, the image of $\mathcal{B}(20,4)$ is brighter than the image of $\mathcal{B}(4,20)$.

For the sake of simplicity, here I only show the codes(in Listing 2) for drawing histograms and visualizing the data specified as $\mathcal{B}(2,2)$.

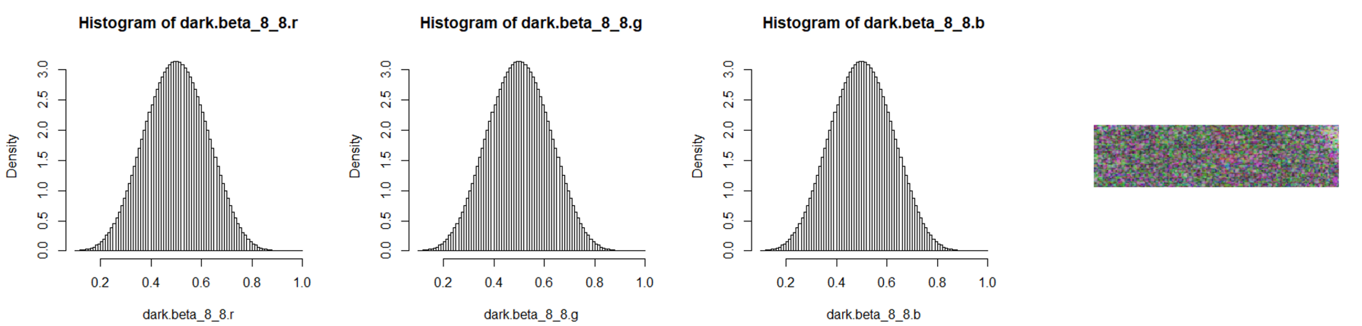
```
# Assign Beta(2,2) to each channel
dark.beta_2_2.r <- qbeta(ecdf(dark[,,1])(dark[,,1]), shape1 = 2, shape2 = 2)
dark.beta_2_2.g <- qbeta(ecdf(dark[,,2])(dark[,,2]), shape1 = 2, shape2 = 2)
dark.beta_2_2.b <- qbeta(ecdf(dark[,,3])(dark[,,3]), shape1 = 2, shape2 = 2)
dark.beta_2_2_indep <- c(dark.beta_2_2.r,dark.beta_2_2.g,dark.beta_2_2.b)

# Draw histograms of each channel
hist(dark.beta_2_2.r,probability = TRUE, breaks = "FD")
hist(dark.beta_2_2.g,probability = TRUE, breaks = "FD")
```

```
hist(dark.beta_2_2.b,probability = TRUE, breaks = "FD")

# Visualize the image
plot(imagematrix((normalize_indep(array(dark.beta_2_2_indep, dim=dim(dark))))))
```

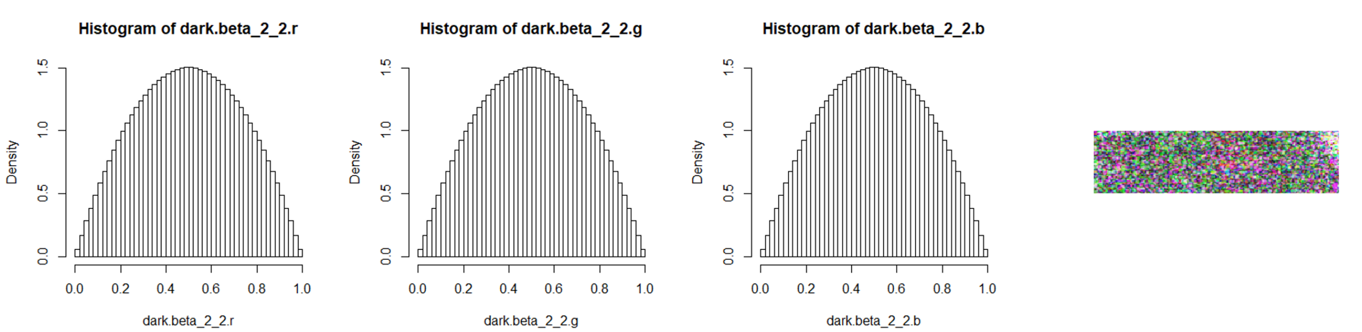
Listing 2: Drawing histograms and visualizing the data specified as $\mathcal{B}(2,2)$



(a) *dark* data specified as B(20,20)

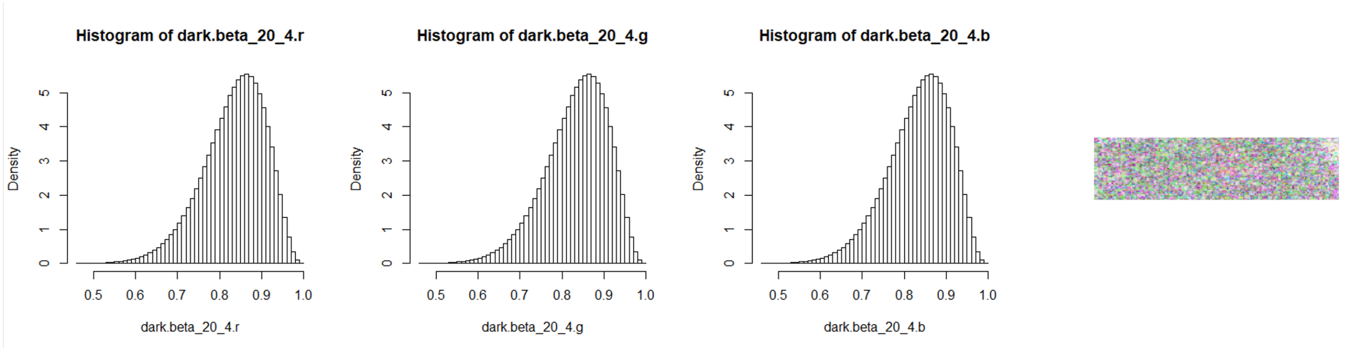


(b) *dark* data specified as B(8,8)

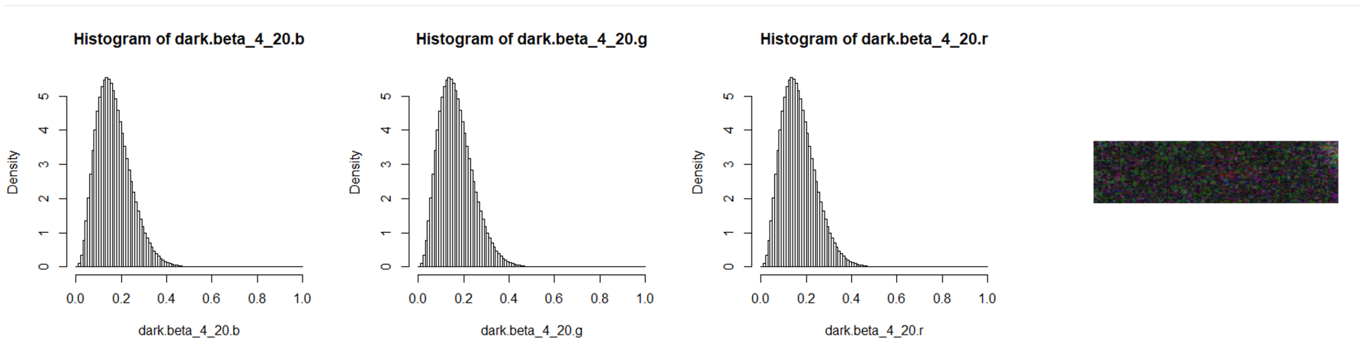


(c) *dark* data specified as B(2,2)

Fig. 4: Histograms and image of *dark* data specified as $\mathcal{B}(2,2), \mathcal{B}(8,8), \mathcal{B}(20,20)$



(a) *dark* data specified as B(20,4)



(b) *dark* data specified as B(4,20)

Fig. 5: Histograms and image of *dark* data specified as $\mathcal{B}(20,4), \mathcal{B}(4,20)$

III. Assignment 3 (2019-07-17)

A. Define and call functions

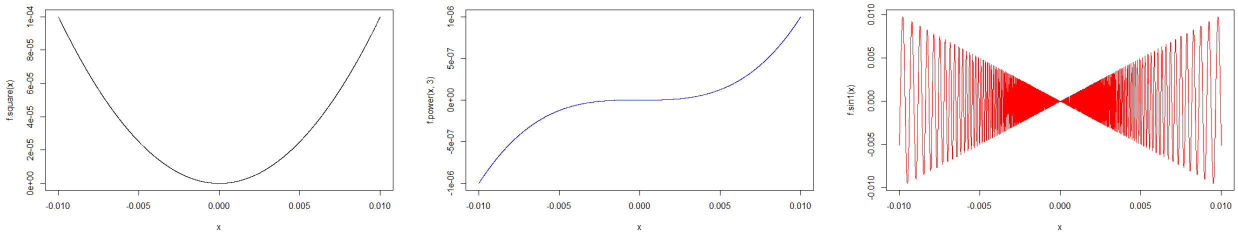
In this part, We define several functions and then call the functions we defined to plot them. The code is shown in Listing 3 and in Fig.6 we plot the functions. Interestingly, Fig.7 shows $f(x) = xsin(\frac{1}{x})$ over different ranges of x, where we can see that function $f(x) = xsin(\frac{1}{x})$ oscillates more and more rapidly when x gets closer to 0, and its amplitude reduces.

```
# Define the functions
f.square <- function(x){
x^2
}
f.power <- function(x,a){
x^a
}
f.sin1 <- function(x){
x*sin(1/x)
}

# Call and plot the functions
x <- seq(-10^-2,10^-2,length.out = 10000)
plot(x,f.square(x),type="l",col="black")
plot(x,f.power(x,3),type="l",col="blue")
plot(x,f.sin1(x),type="l",col="red")
```

Listing 3: Define and call functions

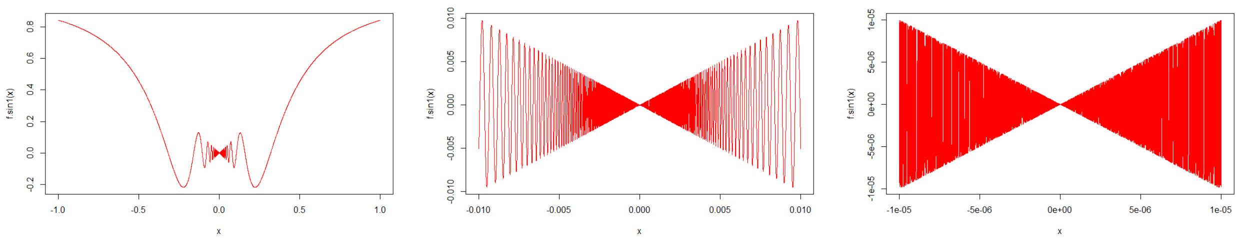
B. Implement $\mathcal{K}$ distribution with R

Fig.8 shows the densities in linear and semilogarithmic scale of $\mathcal{K}$ distribution with L=1, and black, red, blue curves are for $\alpha \in \{2, 5, 10\}$, respectively. To learn the effects of varying α, we show $\mathcal{K}$ distributions with the same unitary mean. From



(a) f.square(x)=x^2 (b) f.power(x,3)=x^3 (c) f.sin1(x)=$xsin(\frac{1}{x})$

Fig. 6: Plot the functions



(a) $x \in (-1, 1)$ (b) $x \in (-0.01, 0.01)$ (c) $x \in (-10^{-5}, 10^{-5})$

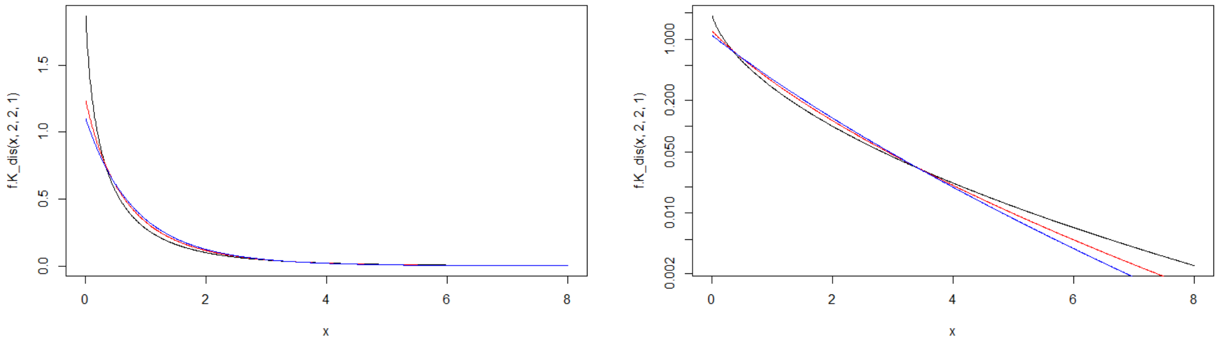
Fig. 7: $f(x) = xsin(\frac{1}{x})$ over different ranges of x

the results we can see that when L is fixed, the smaller the value of α is, the higher contrast the data has, that is smaller α assigns larger probabilities to both small and large values. α is related to the number of elementary backscatterers. When the value of α is large, the density of $\mathcal{K}$ distribution is close to the density of exponential distribution.

Fig.9 shows the densities in linear and semilogarithmic scale of $\mathcal{K}$ distribution with $\alpha = 2, \lambda = 2$, and black, red, blue curves are for $L \in \{1, 4, 7\}$, respectively. As shown in Fig.9, when α and λ is fixed, the larger the value of looks L is, the lower contrast the data has.

```
# K distribution PDF
f.K_dis <- function(x,alpha,lambda,L){
2*lambda*L/(gamma(alpha)*gamma(L))*((lambda*L*x)^((alpha+L)/2-1))*besselK(2*sqrt(
    lambda*L*x),alpha-L)
}

x <- seq(0,8,length.out = 1000)

## K distribution with L=1 alpha in {2,5,10}
plot(x,f.K_dis(x,2,2,1),type="l",col="black")
lines(x,f.K_dis(x,5,5,1),col="red")
lines(x,f.K_dis(x,10,10,1),col="blue")
# semilog scale
plot(x,f.K_dis(x,2,2,1),type="l",col="black",log = "y")
lines(x,f.K_dis(x,5,5,1),col="red")
lines(x,f.K_dis(x,10,10,1),col="blue")

## K distribution with alpha=2 lambda=2  L in {1,4,7}
plot(x,f.K_dis(x,2,2,1),type="l",col="black")
lines(x,f.K_dis(x,2,2,4),col="red")
lines(x,f.K_dis(x,2,2,7),col="blue")
# semilog scale
plot(x,f.K_dis(x,2,2,1),type="l",col="black", log = "y")
lines(x,f.K_dis(x,2,2,4),col="red")
```

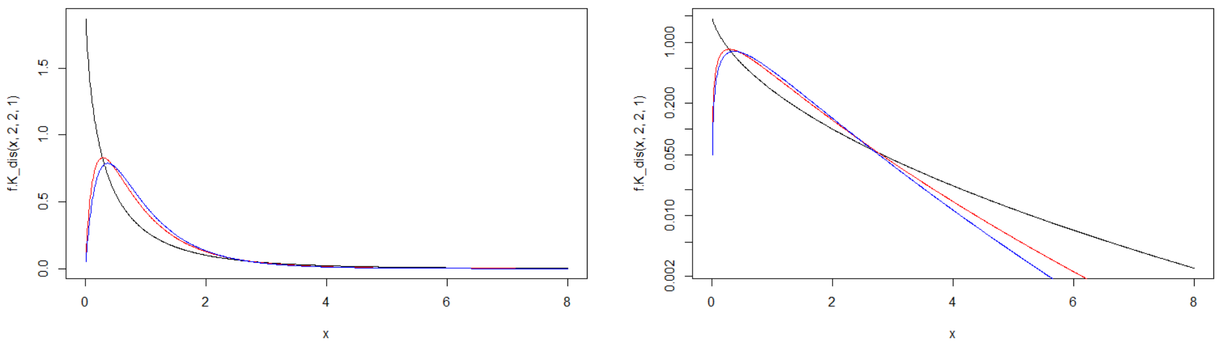
```
lines(x,f.K_dis(x,2,2,7),col="blue")
```
Listing 4: $\mathcal{K}$ distribution



(a) L=1 Densities

(b) L=1 Densities in semilog scale

Fig. 8: Densities in linear and semilogarithmic scale of $\mathcal{K}$ distribution with unitary mean and L=1. Black, red, blue for $\alpha \in \{2, 5, 10\}$, respectively.



(a) $\alpha = 2, \lambda = 2$ Densities

(b) $\alpha = 2, \lambda = 2$ Densities in semilog scale

Fig. 9: Densities in linear and semilogarithmic scale of $\mathcal{K}$ distribution with unitary mean and $\alpha = 2, \lambda = 2$. Black, red, blue for $L \in \{1, 4, 7\}$, respectively.

C. *Implement $\mathcal{G}^0$ distribution with R*

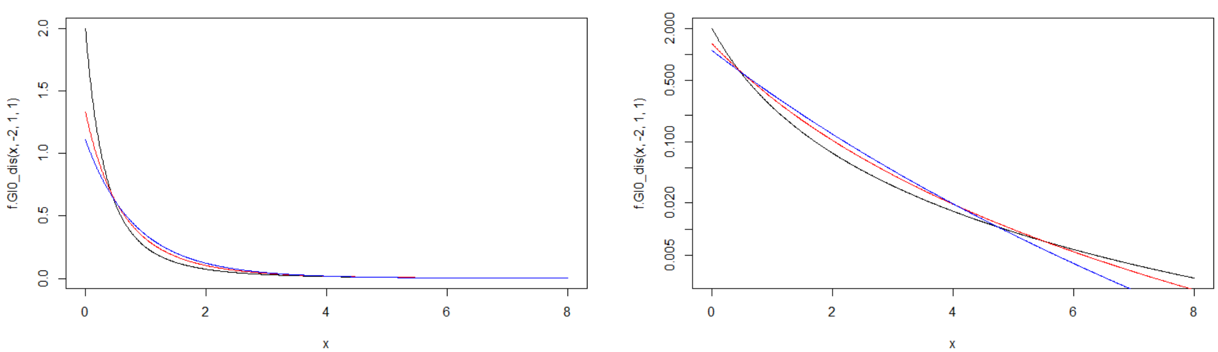
Fig.10 shows the densities in linear and semilogarithmic scale of $\mathcal{G}^0$ distribution with L=1, and black, red, blue curves are for $\alpha \in \{-2, -4, -10\}$, respectively. To learn the effects of varying α, we show $\mathcal{G}^0$ distributions with the same unitary mean. From the results we can see that when L is fixed, the larger the value of α is, the higher contrast the data has.

Fig.11 shows the densities in linear and semilogarithmic scale of $\mathcal{G}^0$ distribution with $\alpha = -2, \gamma = 1$, and black, red, blue curves are for $L \in \{1, 3, 5\}$, respectively. As shown in Fig.9, when α and γ is fixed, the larger the value of looks L is, the lower contrast the data has, which is similar to the $\mathcal{K}$ distribution.

```
# GI0 distribution PDF
f.GI0_dis <- function(x,alpha,g,L){
L^L*(gamma(L-alpha))/(g^alpha*gamma(L)*gamma(-alpha))*x^(L-1)/((g+L*x)^(L-alpha))
}

x <- seq(0,8,length.out = 1000)

## GI0 distribution with L=1 alpha in {-2,-4,-10}
```

```
plot(x,f.GI0_dis(x,−2,1,1),type=”l”,col=”black”)
lines(x,f.GI0_dis(x,−4,3,1),col=”red”)
lines(x,f.GI0_dis(x,−10,9,1),col=”blue”)
# semilog scale
plot(x,f.GI0_dis(x,−2,1,1),type=”l”, col=”black”, log=”y”)
lines(x,f.GI0_dis(x,−4,3,1),col=”red”)
lines(x,f.GI0_dis(x,−10,9,1),col=”blue”)

## GI0 distribution with alpha=−2 g=1  L in {1,3,5}
plot(x,f.GI0_dis(x,−2,1,1),type=”l”, col=”black”)
lines(x,f.GI0_dis(x,−2,1,3),col=”red”)
lines(x,f.GI0_dis(x,−2,1,5),col=”blue”)
# semilog scale
plot(x,f.GI0_dis(x,−2,1,1),type=”l”, col=”black”, log=”y”)
lines(x,f.GI0_dis(x,−2,1,3),col=”red”)
lines(x,f.GI0_dis(x,−2,1,5),col=”blue”)
```

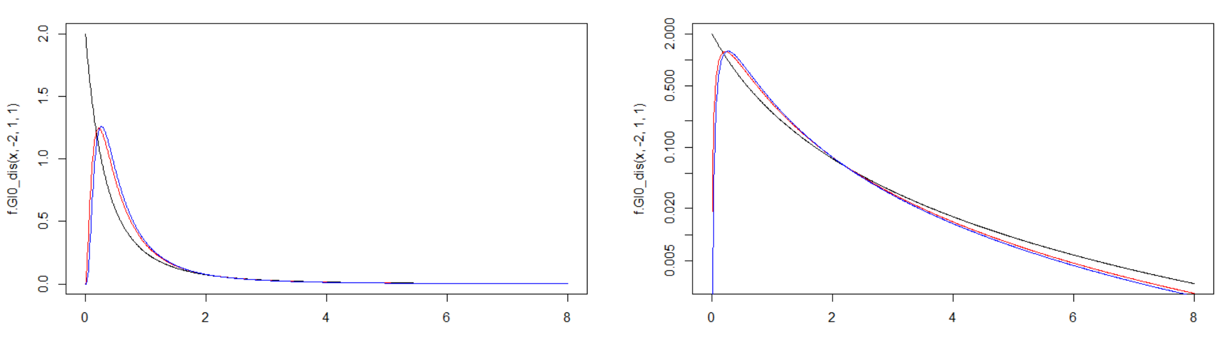
Listing 5: $\mathcal{G}^0$ distribution



(a) L=1 Densities

(b) L=1 Densities in semilog scale

Fig. 10: Densities in linear and semilogarithmic scale of $\mathcal{G}^0$ distribution with unitary mean and L=1. Black, red, blue for $\alpha \in \{-2, -4, -10\}$, respectively.



(a) $\alpha = -2, \gamma = 1$ Densities

(b) $\alpha = -2, \gamma = 1$ Densities in semilog scale

Fig. 11: Densities in linear and semilogarithmic scale of $\mathcal{G}^0$ distribution with unitary mean and $\alpha = -2, \gamma = 1$. Black, red, blue for $L \in \{1, 3, 5\}$, respectively.

IV. Assignment 4 (2019-07-18/19)

A. Parameter estimation: Inference by analogy

For the Uniform distribution, the k-order moment estimator of θ is :

$$\widehat{\theta}_k = \sqrt[k]{\frac{k+1}{n}\sum_{i=1}^{n} U_i^k} \tag{7}$$

Here, we can obtain three estimators of θ formed with $E(x), E(x^2), E(x^3)$:

$$\begin{aligned}
\widehat{\theta}_1 &= 2n^{-1}\sum_{i=1}^{n} U_i \\
\widehat{\theta}_2 &= \sqrt{3n^{-1}\sum_{i=1}^{n} U_i^2} \\
\widehat{\theta}_3 &= \sqrt[3]{4n^{-1}\sum_{i=1}^{n} U_i^3}
\end{aligned} \tag{8}$$

For the (SAR) Gamma distribution characterized by the density:

$$f_\Gamma(z; L, \mu) = \frac{L^L}{\mu^L \Gamma(L)} z^{L-1} \exp\big\{ - Lz/\mu\big\}. \tag{9}$$

We need two linearly independent equations to estimate $\boldsymbol{\theta} = (L, \mu)$. Knowing $E(Z) = \mu$ and $\mathrm{Var}(Z) = \mu^2/L$, we obtain the estimator of $\boldsymbol{\theta}$:

$$\begin{aligned}
\widehat{\mu} &= n^{-1}\sum_{i=1}^{n} Z_i \\
\widehat{L} &= \widehat{\mu}^2/(n^{-1}\sum_{i=1}^{n}(Z_i - \widehat{\mu})^2)
\end{aligned} \tag{10}$$

Listing 6 shows the estimators for the Uniform and the (SAR) Gamma distribution infered by analogy. For the Uniform distribution $\mathcal{U}_{(0,\pi)}$, we obtain the estimators $\widehat{\theta}_1 = 3.143274, \widehat{\theta}_2 = 3.142731, \widehat{\theta}_3 = 3.142359$. For the (SAR) Gamma distribution $\Gamma(2, 7)$, we obtain the estimator $\widehat{\mu} = 7.10646, \widehat{L} = 1.961167$.

```r
## Uniform distribution  U(0,theta)
> x <- runif(1000000,min=0,max=pi)
>
> # Inference from E(x)
> (theta.hat.Mom1 <- 2*mean(x))
[1] 3.143274
> # Inference from E(x^2)
> (theta.hat.Mom2 <- sqrt(3*mean(x^2)))
[1] 3.142731
> # Inference from E(x^3)
> (theta.hat.Mom3 <- (4*mean(x^3))^(1/3))
[1] 3.142359

## Gamma distribution Gamma(mu,L) ---- rgamma(n,L,L/mu)
> x <- rgamma(1000,2,2/7)
>
> (mu.hat <- mean(x))
[1] 7.10646
> (L.hat <- mu.hat^2 /var(x))
[1] 1.961167
```

Listing 6: Parameter estimation for the Uniform and the (SAR) Gamma distribution (Inference by analogy)

B. Fit the histogram with Exponential and $\mathcal{G}^0$ distribution.

In this part, I will first give some conclusions about parameter estimation for the Exponential and $\mathcal{G}^0$ distribution. Then I will analyze a real data from an urban area.

For the Exponential distribution characterized by the density:

$$f_Z(z;\theta) = \theta^{-1} e^{-z/\theta} \tag{11}$$

The k-order moment of Z is given by:

$$\mathrm{E}\, Z^k = k!\theta^k \tag{12}$$

- Inference by analogy
  Based on $\mathrm{E}(Z) = \theta$, we obtain the estimator $\breve{\theta}$:

$$\breve{\theta} = N^{-1} \sum_{n=1}^{N} Z_n \tag{13}$$

- Inference by maximum likelihood
  The maximum likelihood estimator of $\hat{\theta}$ is the maximum point of the reduced log-likelihood:

$$\ell(\theta; \boldsymbol{Z}) = -N ln(\theta) - \frac{1}{\theta} \sum_{n=1}^{N} Z_n \tag{14}$$

  When the derivative of $\ell(\theta; \boldsymbol{Z})$ equals to zero, we obtain the ML estimator $\hat{\theta} = N^{-1} \sum_{n=1}^{N} Z_n$. Note that $\hat{\theta} = \breve{\theta}$.

For the $\mathcal{G}^0$ distribution characterized by the density:

$$f_Z(z;\alpha,\gamma,L) = \frac{L^L \Gamma(L-\alpha)}{\gamma^\alpha \Gamma(L)\Gamma(\alpha)} \frac{z^L}{(\gamma + Lz)^\alpha} \tag{15}$$

The k-order moment of Z is given by:

$$\mathrm{E}\, Z^k = \Big(\frac{\gamma}{L}\Big)^k \frac{\Gamma(-\alpha-k)}{\Gamma(-\alpha)} \frac{\Gamma(L+k)}{\Gamma(L)} \tag{16}$$

- Inference by analogy
  Based on $\mathrm{E}(Z) = \frac{\gamma}{-\alpha-1}$ and $\mathrm{E}(Z^2) = \frac{\gamma^2}{L} \frac{L+1}{(-\alpha-1)(-\alpha-2)}$, we obtain the estimators $\breve{\alpha}, \breve{\gamma}$:

$$\begin{aligned} \breve{\alpha} &= -2 - \frac{L+1}{L\, m_2/m_1^2} \\ \breve{\gamma} &= m_1 \Big(2 + \frac{L+1}{L\, m_2/m_1^2}\Big) \end{aligned} \tag{17}$$

  where m_1, m_2 are the first and second sample moments.
- Inference by maximum likelihood
  The maximum likelihood estimator of $\hat{\alpha}, \hat{\gamma}$ is the maximum point of the reduced log-likelihood:

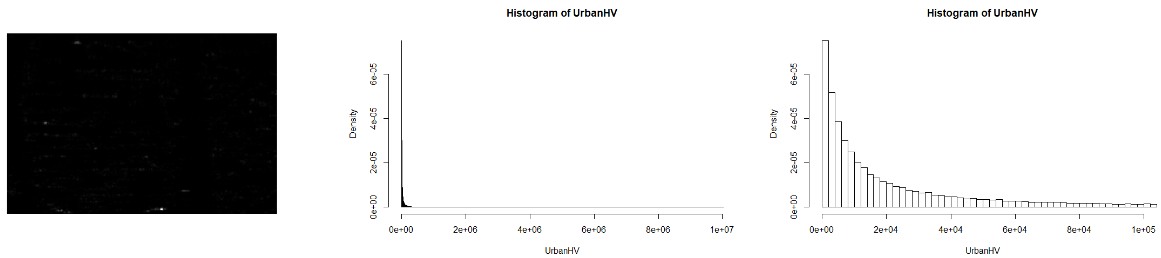
$$\ell(\alpha,\gamma;\widehat{L},\boldsymbol{Z}) = \frac{\Gamma(\widehat{L}-\alpha)}{\gamma^\alpha \Gamma(-\alpha)} + \widehat{L} \sum_{n=1}^{N} \log \frac{Z_n}{\gamma + \widehat{L} Z_n} + \alpha \sum_{n=1}^{N} \log(\gamma + \widehat{L} Z_n) \tag{18}$$

Here we will analyze *UrbanHV* data, shown in Fig.12(a). Fig.12(b)(c) show the histogram and the restricted histogram of the data.

The codes in Listing 8, 9, 10 implement the estimator of Exponential distribution, the moment estimator of $\mathcal{G}^0$ distribution and the ML estimator of $\mathcal{G}^0$ distribution, respectively. Here we assume L = 1 for $\mathcal{G}^0$ distribution. For exponential distribution, we obtain $\hat{\theta} = 45337.62$, and for $\mathcal{G}^0$ distribution, we obtain $\breve{\alpha} = -2.20801, \breve{\gamma} = 100105.9$ estimated by the moments, $\hat{\alpha} = -2.662928, \hat{\gamma} = 62808.457105$ estimated by the maximum likelihood. Fig.13 shows the restricted histogram and fitted exponential densities (black) and $\mathcal{G}^0$ densities with the parameters estimated by the moments (blue) and maximum likelihood (red). According to Fig.13, $\mathcal{G}^0$ distribution with parameters estimated by maximum likelihood fit this data better.

```
## Plot data
plot(imagematrix(normalize(UrbanHV)))
hist(UrbanHV, probability = TRUE, breaks = "FD")
hist(UrbanHV, probability = TRUE, breaks = "FD", xlim = c(0,100000))
x <- seq(0,100000, length.out = 100000)
```

Listing 7: Plot the data



(a) Image (b) Histogram (c) Restricted histogram

Fig. 12: Image and histograms of data *UrbanHV*

```
## estimator of Exponential distribution
E.Est.m1 <- function(z)
{
mean(z)
}
E.estim.Mom <- E.Est.m1(UrbanHV)
E.Mom.fit <- dexp(x,rate=1/E.estim.Mom)
```

Listing 8: The estimator of Exponential distribution

```
## Moment estimator of GI0 distribution
G.Est.m1m2 <- function(z,L)
{
m1<-mean(z)
m2<-mean(z^2)
m2m12 <- m2/m1^2
a <- -2-(L+1)/(L*m2m12)
g <- m1*(2+(L+1)/(L*m2m12))
return(list("alpha"=a,"gamma"=g))
}
G.estim.Mom <- G.Est.m1m2(UrbanHV,1)
G.Mom.fit <- f.GI0_dis(x,G.estim.Mom$alpha,G.estim.Mom$gamma,1)
```

Listing 9: The moment estimator of $\mathcal{G}^0$ distribution

```
# Loglikelihood of GI0(alpha,gamma,1)
G.LogLikelihood_L1 <- function(params)
{
p_alpha <- -abs(params[1])
p_gamma <- abs(params[2])
p_L <- abs(params[3])
n <- length(z)
return (n*(lgamma(p_L-p_alpha)-p_alpha*log(p_gamma)-lgamma(-p_alpha))+(p_alpha-p_L)*
    sum(log(p_gamma+z*p_L)))
}
z <- UrbanHV
G.estim.ML <- maxNR(G.LogLikelihood_L1, start=c(G.estim.Mom$alpha,G.estim.Mom$gamma,1)
    ,activePar=c(TRUE,TRUE,FALSE))$estimate[1:2]
G.ML.fit <- f.GI0_dis(x,G.estim.ML[1],G.estim.ML[2],1)
```

Listing 10: The ML estimator of $\mathcal{G}^0$ distribution

C. Compare estimators with a Monte Carlo experiment.

Consider the Exponential distribution, we can obtain three estimators:

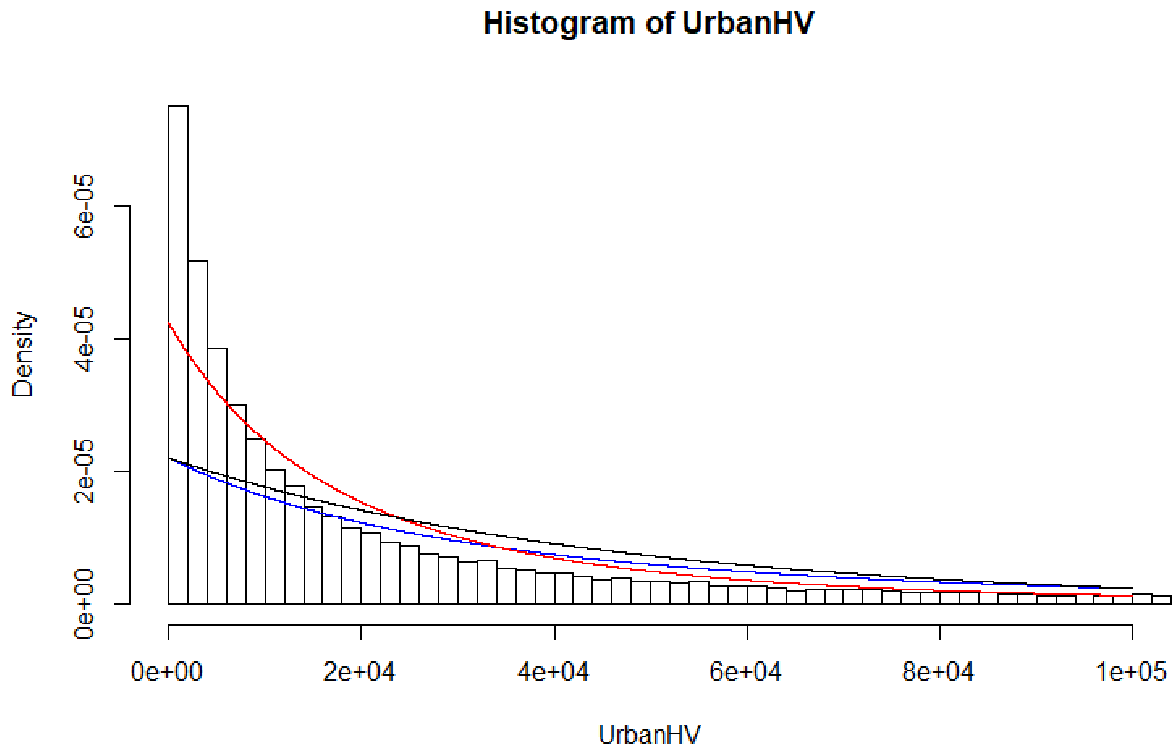


Fig. 13: Restricted histogram and fitted exponential densities (black) and $\mathcal{G}^0$ densities with the parameters estimated by the moments (blue) and maximum likelihood (red) methods.

- The maximum likelihood and first moment estimator $\widehat{\mu}$:

$$\widehat{\mu} = \frac{1}{n}\sum_{i=1}^{n} Z_i \tag{19}$$

- The median estimator $\breve{\mu}$:

$$\breve{\mu} = \frac{1}{\ln 2} q_{1/2}(z_1, \ldots, z_n) \tag{20}$$

- The bootstrapped median estimator $\widetilde{\mu}$.

The codes in Listing 11 implement a Monte Carlo experiment. In the experiment, I set $\mu = 1$, $R = 300$, the sample size $n \in \{3, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 500, 1000, 10000\}$, and the number of replications $r = 2 \times 10^6/n$. For each (n, μ) we can obtain the bias, and the mean quadratic error of $\widehat{\mu}$, $\breve{\mu}$ and $\widetilde{\mu}$. Fig.14 shows how the bias(14(a)) and the mean square error(14(b)) change with the number of samples n.

From Fig.14(a) we can see that when sample size is large enough, the biases of $\widehat{\mu}$, $\breve{\mu}$ and $\widetilde{\mu}$ are close, which means we can choose any one of these three estimators. However, when sample size is small, the bias of $\breve{\mu}$ is large while the bias of $\widehat{\mu}$, $\widetilde{\mu}$ are both small (when sample size $n > 5$). When $n = 3$, though $\widetilde{\mu}$ is much smaller than $\breve{\mu}$, it is still larger than $\widehat{\mu}$. The bias of $\widehat{\mu}$ is always small for almost every sample size.

From Fig.14(b) we can see that when sample size is large enough, the MSE of $\widehat{\mu}$, $\breve{\mu}$ and $\widetilde{\mu}$ are close too. While when sample size is small, the MSE of $\breve{\mu}$ is smaller than $\widetilde{\mu}$ and larger than $\widehat{\mu}$. The MSEs of all these estimators show a trend of decreasing with the increase of sample size n.

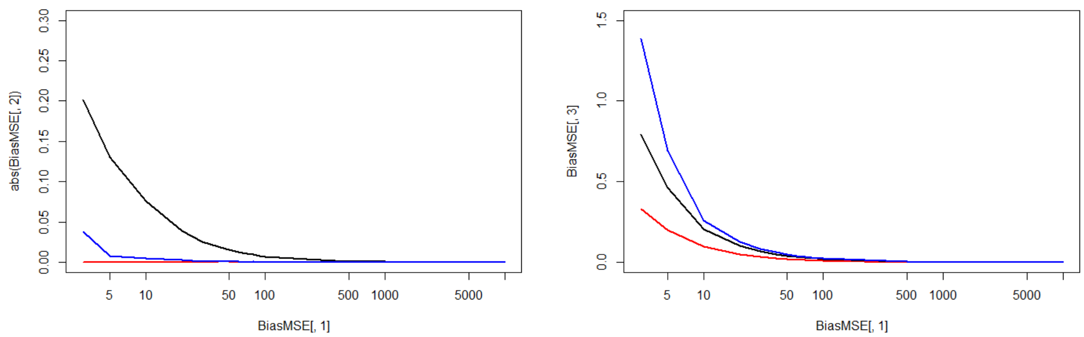
Considering both bias and MSE, maximum likelihood estimator has the best performance.

```
library(gtools)
require(gtools)
set.seed(123)
## bootstrapped median estimator
E.est.median.bootstrap <- function(z,R){
```

```
theta.hat <- median(z)/log(2)
sample_size <- length(z)

if(R>(sample_size^sample_size)){
m.Bootstrap <- permutations(sample_size,sample_size,z,set=TRUE, repeats.allowed=TRUE)
return(2*theta.hat - mean(unlist(lapply(m.Bootstrap,median)))/log(2))
}
else{
v.Bootstrap <- rep(0,R)
for(idx in 1:R){
x <- sample(z,replace = TRUE)
v.Bootstrap[idx] <- median(x)/log(2)
}
}
return (2*theta.hat-mean(v.Bootstrap))
}

## N for sample size
N<-c(3,5,10,20,30,40,50,60,70,80,90,100,500,1000,10000)
BiasMSE <- matrix(nrow=15,ncol=7)

i<-0
for(n in N){
i<-i+1
r<- ceiling(2*10^6/n)
v.mu.ML <- array(rep(0,r))
v.mu.Med <- array(rep(0,r))
v.mu.BootMed <- array(rep(0,r))
for (j in 1:r)
{
z<- rexp(n)
# First moment estimator or ML estimator
v.mu.ML[j]<- mean(z)
# Median estimator
v.mu.Med[j]<- median(z)/log(2)
# Bootstrapped median estimator
v.mu.BootMed[j]<- E.est.median.bootstrap(z,300)
}

bias.ML<- mean(v.mu.ML)-1
mse.ML <- mean((v.mu.ML-1)^2)
bias.Med<- mean(v.mu.Med)-1
mse.Med <- mean((v.mu.Med-1)^2)
bias.BootMed<- mean(v.mu.BootMed)-1
mse.BootMed <- mean((v.mu.BootMed-1)^2)

BiasMSE[i,]<-c(N[i],bias.ML,mse.ML,bias.Med,mse.Med,bias.BootMed,mse.BootMed)
}

# abs(Bias)
plot(BiasMSE[,1],abs(BiasMSE[,2]),type="l",col="red",log = "x",ylim=c(0,0.3),lwd=2)
lines(BiasMSE[,1],abs(BiasMSE[,4]),col="black",lwd=2)
lines(BiasMSE[,1],abs(BiasMSE[,6]),col="blue",lwd=2)
# MSE
plot(BiasMSE[,1],BiasMSE[,3],type="l",col="red",log = "x",ylim=c(0,1.5),lwd=2)
lines(BiasMSE[,1],BiasMSE[,5],col="black",lwd=2)
```

```
lines(BiasMSE[,1],BiasMSE[,7],col="blue",lwd=2)
}
```

Listing 11: Monte Carlo experiment for comparing estimators of Exponential distribution



(a) The bias

(b) The mean square error

Fig. 14: The dependence of the bias and the mean square error on the sample size n.