University at Buffalo

STA511 Statistical Computing, Fall 2015

Final Homework

Ezgi Karaesmen

12/14/15

1 Solutions

1. **Problem 1:** Generating Normal random variables using the given Laplace Distribution:

   (a) Finding the optimal rejection constant c and optimal θ such that $c \geq sup\frac{f(x)}{g(x)}$, where $f(x)$ is the standard normal pdf (i.e. N(0,1)) and $\theta > 0$:

$$c_\theta = sup\left(\frac{f(x)}{g(x)}\right)$$

$$= sup\left(\frac{\frac{1}{\sqrt{2\pi}}\, e^{-x^2/2}}{\frac{\theta}{2}\, e^{-\theta|x|}}\right)$$

$$= sup\left(\frac{1}{\sqrt{2\pi}}\,\frac{2}{\theta}\, e^{\frac{-x^2}{2}+\theta|x|}\right)$$

$$c_\theta = sup\left(\begin{cases} \frac{1}{\sqrt{2\pi}}\,\frac{2}{\theta}\, e^{\frac{-x^2}{2}+\theta x}\ , & x \geq 0 \\ \frac{1}{\sqrt{2\pi}}\,\frac{2}{\theta}\, e^{\frac{-x^2}{2}-\theta x}\ , & x < 0 \end{cases}\right)$$

$$= \begin{cases} \frac{\delta}{\delta x}\,\frac{1}{\sqrt{2\pi}}\,\frac{2}{\theta}\, e^{\frac{-x^2}{2}+\theta x} \overset{\text{set}}{=} 0\ , & x \geq 0 \\ \frac{\delta}{\delta x}\,\frac{1}{\sqrt{2\pi}}\,\frac{2}{\theta}\, e^{\frac{-x^2}{2}-\theta x} \overset{\text{set}}{=} 0\ , & x < 0 \end{cases}$$

$$= \begin{cases} \sqrt{\frac{2}{\pi\theta^2}}\, e^{\frac{-x^2}{2}+\theta x}\,(-x+\theta) = 0\ , & x \geq 0 \\ \sqrt{\frac{2}{\pi\theta^2}}\, e^{\frac{-x^2}{2}-\theta x}\,(-x-\theta) = 0\ , & x < 0 \end{cases}$$

$$\text{critical points} = \begin{cases} x = \theta\ , & x \geq 0 \\ x = -\theta\ , & x < 0 \end{cases}$$

$$c_\theta = \begin{cases} \sqrt{\frac{2}{\pi\theta^2}}\, e^{\frac{-\theta^2}{2}+\theta^2} = \sqrt{\frac{2}{\pi\theta^2}}\, e^{\frac{\theta^2}{2}}, & x \geq 0 \\ \sqrt{\frac{2}{\pi\theta^2}}\, e^{\frac{-(-\theta)^2}{2}-\theta(-\theta)} = \sqrt{\frac{2}{\pi\theta^2}}\, e^{\frac{\theta^2}{2}}, & x < 0 \end{cases}$$

$$c_\theta = \sqrt{\frac{2}{\pi\theta^2}}\, e^{\frac{\theta^2}{2}}, \quad -\infty < 0 < \infty \quad \text{and} \quad \theta > 0$$

The optimal θ value would be the value that allows the smallest c_θ where $c_\theta \geq \frac{f(x)}{g(x)}$. Therefore minimizing the c_θ function would allow us to determine the optimal θ. c_θ function was plotted against the possible values of θ to show the behavior of the function in Figure 1.

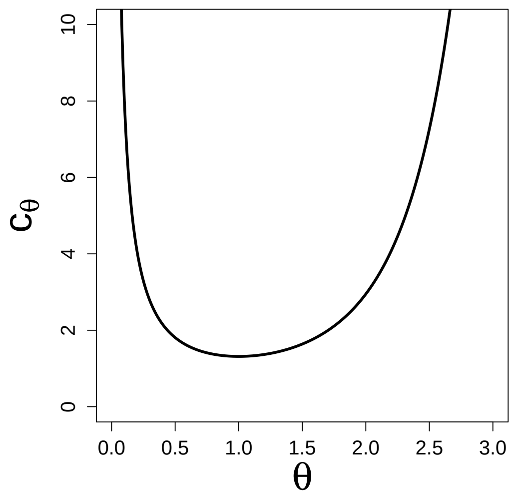

Figure 1: Plot of c_θ function, where $c_\theta = \sqrt{\frac{2}{\pi\theta^2}}\, e^{\frac{\theta^2}{2}} = sup\left(\frac{f(x)}{g(x)}\right)$.

Minima of the c_θ function was determined with `R` using `nlminb()` function and the optimal θ was found as `1.001` $\approx$ `1`, where c_θ was determined as `1.315`.

(b) 1000 observations from N(0,1) using a generalized rejection algorithm was generated. This problem requires generating random variables from g(x) distribution. Therefore, the first part of the problem will be sampling from g(x) distribution via inversion method and in the second part of the problem random variables from f(x) (i.e. N(0,1)) will be generated via generalized rejection method.

Inversion method requires the inverse CDF (i.e. $G^{-1}(x)$) of the Laplace distribution g(x), where θ was previously determined as 1. This can be achieved by determining the $G^{-1}(x)$ function, generating random variables from a Uniform distribution U(0,1) with `R`'s `runif()` function and finally generating Laplace random variables by plugging-in the Uniform random variables to $G^{-1}(x)$ function.

$$g(x|\theta = 1) = \begin{cases} \frac{1}{2}\, e^{x} & , \quad x < 0 \\ \frac{1}{2}\, e^{-x} & , \quad x \geq 0 \end{cases}$$

$$G(x) = \begin{cases} \frac{1}{2}\, e^{x} & , \quad x < 0 \\ 1 - \frac{1}{2}\, e^{-x} & , \quad x \geq 0 \end{cases}$$

$$G^{-1}(u) = \begin{cases} ln(2u) & , \quad u \in \left[0, \frac{1}{2}\right] \\ -ln[2(1-u)] & , \quad u \in \left[\frac{1}{2}, 1\right] \end{cases}$$

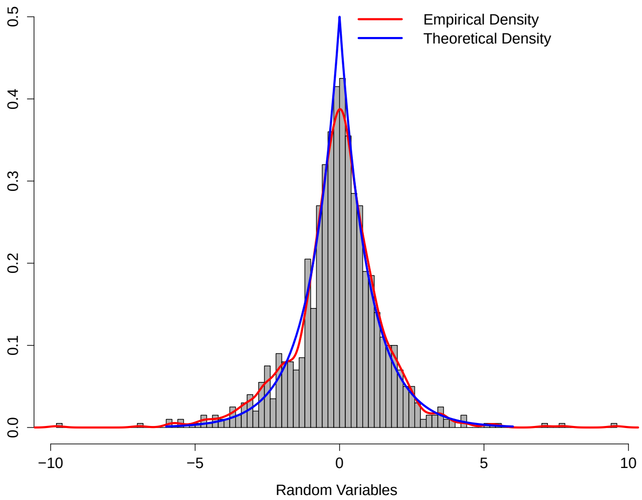

Figure 2: True histogram of the generated Laplace random variables via inversion method

As shown in Figure 2, generated random variables follow the density of theoretical Laplace distribution, suggesting that the sampling method was successful. The generated random variables were then used for the rejection method to generate random variables from a standard normal distribution (i.e. N(0,1)). For the rejection method, $\frac{g(x)}{f(x)}$ function was computed as follows:

$$\frac{g(x)}{f(x)} = \frac{\frac{\theta}{2}\, e^{-\theta|x|}}{\frac{1}{\sqrt{2\pi}}\, e^{-x^2/2}}$$

$$= \frac{\sqrt{2\pi}\,\theta}{2}\, e^{\frac{x^2}{2}-\theta|x|}$$

Upon random variable generation via generalized rejection method, a true histogram of accepted random variables was produced and is shown in Figure 3, where the acceptance rate was determined as `0.784` (i.e. 78% of the observations were accepted). The goodness of fit is tested in the next part of the problem.

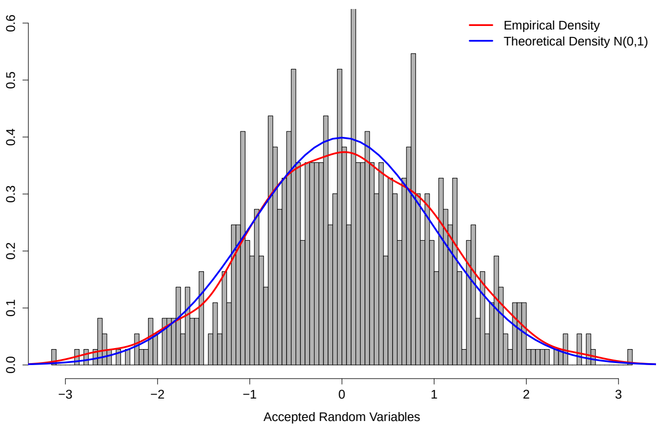

Figure 3: True histogram of the generated standard normal random variables via generalized rejection method.

(c) In order to determine if the generated samples are consistent with a standard normal distribution, a Kolmogorov-Smirnov test was performed. The test statistic was determined as `D = 0.036935` and the null hypothesis failed to be rejected with a p-value of `p-value = 0.2351`, where the null hypothesis suggests that the sampled random variables are coming from the same theoretical distribution that they were tested for. This means that the generated samples have a N(0,1) distribution.

2. **Problem 2:** This problem uses accident count data where the data are the counts of accident insurance policies reporting exactly y_i claims during a particular year with the following model:

$$P(X=x|\alpha,\beta)=\frac{\beta^a}{\Gamma(\alpha)}\frac{\Gamma(\alpha+x)}{x!}\frac{1}{(\beta+1)^{\alpha+x}} \quad , \text{ where } \alpha>0,\ \beta=3.2,\ \text{and } x=0,1,2...$$

(a) Find the method of moments estimators for α, where $E[X]=\frac{\alpha}{\beta}$ and $Var[X]=\frac{\alpha(\beta+1)}{\beta^2}$.

$$E[X] = \frac{\alpha}{3.2} = \bar{X}$$

$$\tilde{\alpha}_{MOM} = 3.2\bar{X} \quad , \quad \text{where } \bar{X} \text{ was computed as } \texttt{0.214}$$

$$\tilde{\alpha}_{MOM} = 0.686$$

(b) Finding the maximum likelihood estimator for α cannot be achieved with the usual approach due to $\Gamma(\alpha)$ and $\Gamma(\alpha+x)$ functions, and requires computation with `R`. Knowing that the data is coming from a discrete distribution, maximum likelihood function can be rearranged and used for computation in `R`:

$$L(\alpha) = \prod_{i=1}^{n}\frac{3.2^a}{\Gamma(\alpha)}\frac{\Gamma(\alpha+x)}{x!}\frac{1}{(3.2+1)^{\alpha+x}} \quad \text{where } n = \text{total number of claim counts}$$

$$\text{If } \ f_\alpha(x) = \frac{3.2^a}{\Gamma(\alpha)}\frac{\Gamma(\alpha+x)}{x!}\frac{1}{(3.2+1)^{\alpha+x}} \quad \text{then,}$$

$$L(\alpha) = \prod_{i=1}^{n} f_\alpha(X_i)$$

$$= \prod_{i=1}^{f} f_\alpha(X_i) \prod_{i=f+1}^{g} f_\alpha(X_i) \prod_{i=g+1}^{h} f_\alpha(X_i) ... \prod_{i=l+1}^{x} f_\alpha(X_i)$$

Where $f,\ g,\ h,\ i,\ j,\ k,\ l\ ,m$ are the counts of $claims = \{0,1,2,3...,7\}$ respectively,

$$L(\alpha) = \left[f_\alpha(0)\right]^f \left[f_\alpha(1)\right]^g \left[f_\alpha(2)\right]^h \left[f_\alpha(3)\right]^i \left[f_\alpha(4)\right]^j \left[f_\alpha(5)\right]^k \left[f_\alpha(6)\right]^l \left[f_\alpha(7)\right]^m$$

$$\ell(\alpha) = f\,log(f_\alpha(0)) + g\,log(f_\alpha(1)) + h\,log(f_\alpha(2)) + i\,log(f_\alpha(3)) + ... + m\,log(f_\alpha(7))$$

Using the above defined log-likelihood function, $\ell(\alpha)$ can be computed for different values of α and the maximum can be found. Plot of log-likelihood function ($\ell(\alpha)$) is shown in Figure 4.

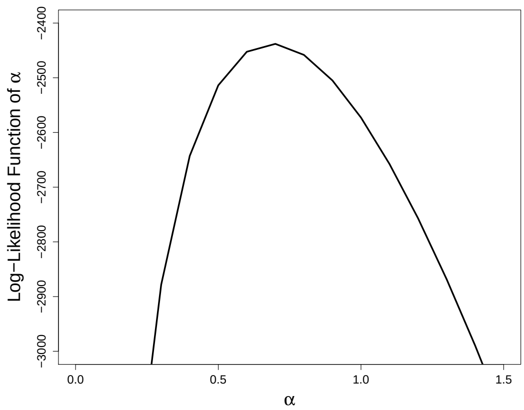

Figure 4: Plot of log-likelihood function ($\ell(\alpha)$) for different values of α around its maximum.

The maximum of the $\ell(\alpha)$ was then computed using `optimize(, maximum = T)` function and was determined as `0.687`. Then the 95% confidence interval for $\hat{\alpha}_{MLE}$ was determined with pivotal bootstrapping method as $\text{CI}\big(0.6870921, 0.6870923\big)$.

(c) Estimate π, the probability that a randomly selected policy has more than 2 claims in the year.

$$\hat{\pi} = P(X > 2|\,\alpha = \hat{\alpha}_{MLE}\ ,\beta = 3.2) \quad \text{, where } \hat{\pi} \text{ is the maximum likelihood estimator of } \pi$$

$$= P(X = 3 \cup X = 4 \cup X = 5 \cup X = 6 \cup X = 7\,|\,\alpha = \hat{\alpha}_{MLE}\ ,\beta = 3.2)$$

$$= P(X = 3) + P(X = 4) + P(X = 5) + P(X = 6) + P(X = 7)\ \text{, where } \alpha = \hat{\alpha}_{MLE}\ ,\beta = 3.2$$

$$= \sum_{n=3}^{7} \frac{3.2^{0.687}}{\Gamma(0.687)} \frac{\Gamma(0.687+2)}{x_i!} \frac{1}{(3.2+1)^{0.687+x_i}}$$

$\hat{\pi}$ was computed `0.0075` using `R` and the 95% confidence interval was determined as $\text{CI}\big(0.007452261, 0.007452262\big)$ using pivotal bootstrapping method.

3. **Problem 3:** The the distribution is described by the following density function:

$$f(x) = \begin{cases} x+1 & , \quad x \in \big[-1, 0\big] \\ -x+1 & , \quad x \in \big[0, 1\big] \\ 0 & , \quad \text{elsewhere} \end{cases}$$

(a) Provide the simple rejection algorithm and generate 100 random observations from f using simple rejection sampling. For the simple rejection sampling method, rejection constant (i.e. c) is defined as $c = max\{f(x)\ ;\ -1 \leq x \geq 1\}$ and computed with `optimize()` function. c was determined as 1 and acceptance rate was `0.52` (i.e. 52% of the random variables were accepted).

(b) Provide the inversion algorithm to sample from F and generate 100 random observations using inversion sampling. For the inversion method, CDF function $F(x)$ and inverse CDF function $F^{-1}(x)$ was determined as follows:

$$F(x) = \begin{cases} 0 & , \quad x \leq -1 \\ \frac{(x+1)^2}{2} & , \quad x \in \big(-1, 0\big] \\ 1 - \frac{(1-x)^2}{2} & , \quad x \in \big(0, 1\big) \\ 1 & , \quad x \geq 1 \end{cases}$$

$$F^{-1}(u) \;=\; \begin{cases} \sqrt{2u} - 1 & , \quad u \in \big(0, \frac{1}{2}\big) \\ 1 - \sqrt{2(-u+1)} & , \quad u \in \big[\frac{1}{2}, 1\big) \end{cases}$$

(c) Histograms of the generated random variables using simple rejection and inversion method are presented in Figure 5.

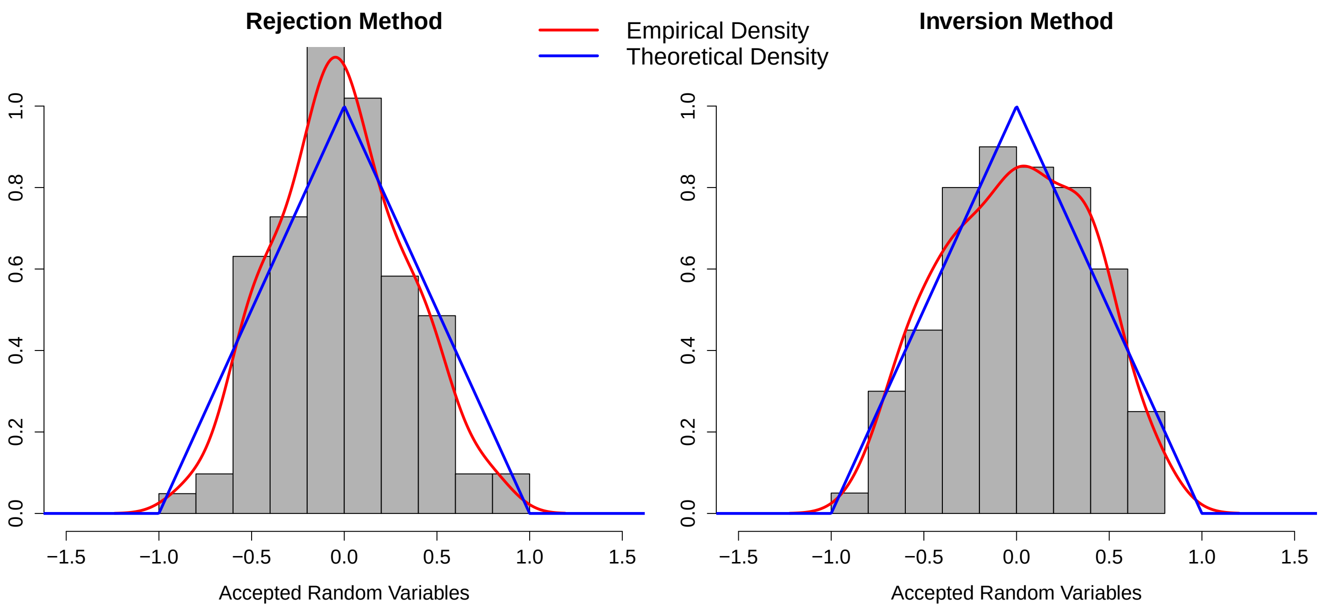

Figure 5: True histograms of generated random variables.

4. **Problem 5:** $X_1,\, X_2,\, ...\, , X_n \sim Uniform\,(a, 5)$ were a is an unknown parameter.

   (a) Find method of moments estimator for a.

$$E[X] \;= \frac{a+5}{2} \;=\; \bar{X}$$

$$\tilde{a}_{MOM} \;= 2\bar{X} - 5$$

$$= \frac{2\sum_{i=1}^{n} X_i}{n} - 5$$

(b) Find MLE of a.

$$f(x) = \begin{cases} \frac{1}{5-a} & , \quad a \le x \le 5 \\ 0 & , \quad otherwise \end{cases}$$

$$L(x|a) = \begin{cases} \prod_{i=1}^{n} \frac{1}{5-a} & , \quad a \le x \le 5 \\ 0 & , \quad otherwise \end{cases}$$

$$L(x|a) = \begin{cases} \left[\frac{1}{5-a}\right]^n & , \quad a \le x \le 5 \\ 0 & , \quad otherwise \end{cases}$$

Since $a \le X_i$, the value maximizing the $L(x|a)$ can only be the $min(X_1, X_2, ..., X_n)$.

$$Therefore, \quad \hat{a}_{MLE} = X_{[1]}$$

(c) Find the MLE of τ, where $\tau = E[X] = \int_a^5 xf(x)\,dx$ and $f(x) = \frac{1}{5-a}$.

$$\tau = E[X] = \int_a^5 xf(x)\,dx$$

$$= \frac{a+5}{2}$$

$$\hat{\tau}_{MLE} = \frac{\hat{a}_{MLE}+5}{2} = \frac{X_{[1]}+5}{2}$$

(d) Supposing that $a = 1$ and $n = 10$ the coverage probability for a 95% confidence interval was found for τ using bootstrapping with $\hat{\tau}_{MLE}$ (MLE of τ) and $\tilde{\tau}_{MOM}$ (method of moments based estimator of τ). 10 random variables from a $Uniform(1, 5)$ distribution was generated using `runif()` function and plug-in estimators were used to find $\hat{\tau}_{MLE}$ and $\tilde{\tau}_{MOM}$. Then percentile bootstrapping method was used to determine the confidence intervals.

$$
\begin{aligned}
\hat{\alpha}_{MLE} &= \frac{5+\hat{\alpha}_{MLE}}{2} = \frac{5+X_{[1]}}{2} \\
\tilde{\alpha}_{MOM} &= \frac{5+\tilde{\alpha}_{MOM}}{2} = \frac{5+(2\bar{X}-5)}{2} \\
&= \bar{X} = \frac{\sum_{i=1}^{10} X_i}{10}
\end{aligned}
$$

$\hat{\tau}_{MLE}$ was computed `3.166` with a confidence interval of $\texttt{CI}(\texttt{3.166}, \texttt{3.497})$ and $\tilde{\tau}_{MOM}$ was `2.506` with a confidence interval of $\texttt{CI}(\texttt{1.943}, \texttt{3.183})$, indicating that $\hat{\tau}_{MLE}$ was less variable than $\tilde{\tau}_{MOM}$.

2 Appendix

1. **Problem 1:**

   (a) R code for finding the optimal c_θ and θ:

   ```
   xtheta <- seq(0,3,by=0.01)
   ytheta <- sqrt(2/(pi*xtheta^2))*exp((xtheta^2)/2)
   par(mar=c(5,6,2,1))
   plot(xtheta, ytheta, type="l", lwd=3, xlab=expression(theta),
        ylab=expression(c[theta]), ylim=c(0,10), cex.lab=2)
   opttheta <- function(xtheta){sqrt(2/(pi*xtheta^2))*exp((xtheta^2)/2)}
   opt_theta <- nlminb(0.001, opttheta)$par # optimal theta
   round(nlminb(0.001, opttheta)$objective, 3) # optimal rejection constant C
   ```

   (b) R code for inversion method and rejection method is presented below:

   ```
   ###### Inversion method ######

   quant_lap <- function(u){ ## inverse CDF function (i.e. quantile function)
     if (u < 1/2){
       log(2*u)
     } else {
       -log(2*(1-u))
     }
   }

   u <- runif(1000)
   rv_lap <- sapply(u, quant_lap)

   ## True histogram of Laplace RVs ###
   laplace <- function(x){ ## theoretical PDF function
     if(x < 0){
       0.5*exp(x)
     } else {
       0.5*exp(-x)
     }
   }

   lap <- seq(-6,6,by=0.1)
   lap.pdf <- sapply(lap, laplace)

   truehist(rv_lap, col="gray70", ylim=c(0,0.5), nbins=100, ## truehist of generated RVs
            xlab="Random Variables")
   lines(density(rv_lap)$x,density(rv_lap)$y, col="red", lwd=3) #empirical density function
   lines(lap, lap.pdf, col="blue", lwd=3) #theoretical density function
   legend("topright", c("Empirical Density", "Theoretical Density"), col=c("red", "blue"),
          lwd=3)

   ##### Generalized rejection ######


   goverf = function(x,theta=opt_theta){                 # g(x) / f(x) function
     y = (sqrt(2*pi)*theta/2)*exp(x^2/2 - theta*abs(x))
     return(y)}

   xc = rv_lap # generated 1000 Laplace RVs
   uc=runif(1000,0,1)
   tc = c_opt*sapply(xc, goverf) #c_opt as determined in the 1st part of the problem
   ```

```
ut = uc*tc
acc_obs <- xc[ut <=1]
sum(ut<=1)/1000 #acceptance rate = 0.78

## True histogram of N(0,1) RVs ##

par(mar=c(5,3,2,1))
truehist(acc_obs, col="gray70", ylim=c(0,0.7), nbins=100,
         xlab="Accepted Random Variables")
lines(density(acc_obs)$x,density(acc_obs)$y, col="red", lwd=3)
lines(seq(-4,6,by=0.1), dnorm(seq(-4,6,by=0.1)), col="blue", lwd=3)
legend("topright", c("Empirical Density", "Theoretical Density"), col=c("red", "blue"),
       lwd=3)
```

(c) Kolmogorov-Smirnov test of the generated samples.

```
ks.test(acc_obs, "pnorm", 0,1)

# Result
One-sample Kolmogorov-Smirnov test

data:  acc_obs
D = 0.036935, p-value = 0.2351
alternative hypothesis: two-sided
```

2. **Problem 2:**

(a) Find the method of moments estimators for α.

```
claims <- c(rep.int(0, 7840), rep.int(1, 1317), rep.int(2, 239), rep.int(3, 42),
                   rep.int(4, 14), rep.int(5,4), rep.int(6,4), rep.int(7,1))
b = 3.2
a_mom <- round(b*mean(claims), 3)
```

(b) Find the maximum likelihood estimator for α with its approximate 95% confidence interval.

```
LF <- NULL ; logLF <- NULL

log_LF <- function(x, a, b = 3.2){
  xi <- unique(x)
  for(i in 1:length(xi)){
    counts <- length(x[x == xi[i]])
    LF[i] <- (b^a/gamma(a))*(gamma(a+ xi[i])/ factorial(xi[i]))*(1/(b+1)^(a+x[i]))
    logLF[i] <- log(LF[i])*counts
  }
  sum(logLF)
}

a_MLE <- optimize(log_LF, c(0.1,1), maximum = T, x=claims)$maximum ## a_MLE

### Plot of log-likelihood function ####
a <- seq(0.000001,100, by=0.1)
log_lfun <- sapply(a, log_LF, x=claims)
par(mar=c(5,6,2,1))
plot( a, log_lfun, type = "l", xlim=c(0,1.5), ylim=c(-3000, -2000),lwd=3,
     ylab = expression(paste("Log-Likelihood Function of ", alpha)),
     xlab = expression(alpha), cex.lab =1.5 )

### Bootstrapping for CI ###
boot_mle = NULL
```

```
sim=1000
for(i in 1:sim){
  boot <- sample(claims, length(claims))
  boot_mle[i] <- optimize(log_LF, c(0.1,1), maximum = T, x=boot)$maximum
}

pivotal_alpha = c(2*a_MLE-quantile(boot_mle,.975), 2*a_MLE-quantile(boot_mle,.025))
```

(c) Estimate π, the probability that a randomly selected policy has more than 2 claims in the year and fins its 95% confidence interval.

```
prbs=NULL
pi_hat <- function(pr, a, b=3.2){
  for(i in 1:length(pr)){
    prbs[i] <- (b^a/gamma(a))*(gamma(a+ pr[i])/ factorial(pr[i]))*(1/(b+1)^(a+pr[i]))
    }
  sum(prbs)
  }

pi_hat_mle <- pi_hat(pr=c(3:7), a=a_MLE)

pi_mle= NULL
for(i in 1:sim){
  boot <- sample(claims, length(claims))
  boot_mle <- optimize(log_LF, c(0.1,1), maximum = T, x=boot)$maximum
  pi_mle[i] <- pi_hat(pr=c(3:7), a=boot_mle)
}

pivotal_pi = c(2*pi_hat_mle-quantile(pi_mle,.975), 2*pi_hat_mle-quantile(pi_mle,.025))
```

3. **<u>Problem 3:</u>**

(a) Provide the simple rejection algorithm and generate 100 random observations from f using simple rejection sampling.

```
fx <- function(x){
  if(-1 <= x & x <= 0){
    x+1
  }else if(0 <= x & x <= 1){
    -x+1
  }else{0}
}

c_rej <- optimize(fx, c(-1,1), maximum=T)$objective ## rejection constant

xc = runif(200, -1,1)
uc=runif(200,0,1)
fxi = sapply(xc, fx)
tc = c_rej/fxi
ut = uc*tc
acc_obs <- xc[ut <=1]
length(acc_obs)/200 # 0.52 acceptance rate, 104 RVs were generated.
```

(b) Provide the inversion algorithm to sample from F and generate 100 random observations using inversion sampling.

```
f_inv <- function(x){    ## inverse F function
  if(0 < x & x < 0.5){
```

```
    sqrt(2*x)-1
  }else if(0.5 <= x & x < 1){
    1 - sqrt(2*(-x+1))
  }else{0}
}

U <- runif(100)
RVs <- sapply(U, f_inv)
```

(c) Produce a histogram for your answer in (a) and a histogram for your answer in (b).

```
xi=runif(1000, -2,2)
xi <- xi[order(xi)]
plot.fx <- sapply(xi, fx) ## theoretical density computed

par(mar=c(5,3,3,1), mfrow=c(1,2))
truehist(acc_obs, col="gray70",nbins=10, main="Rejection Method",
         xlab="Accepted Random Variables",ylim=c(0,1.1), xlim=c(-1.5,1.5))
lines(density(acc_obs)$x,density(acc_obs)$y, col="red", lwd=3)
lines(xi, plot.fx, col="blue", lwd=3)
 legend(0.9, 1.25, c("Empirical Density", "Theoretical Density"), col=c("red", "blue"),
         lwd=3,xpd=NA,bty = "n", cex=1.2)
truehist(RVs, col="gray70",nbins=10, main="Inversion Method",
         xlab="Accepted Random Variables",ylim=c(0,1.1), xlim=c(-1.5,1.5))
lines(density(RVs)$x,density(RVs)$y, col="red", lwd=3)
lines(xi, plot.fx, col="blue", lwd=3)
```

4. **Problem 5.c:** For $a = 1$ and $n = 1$ 95% confidence intervals for τ using percentile bootstrapping with $\hat{\tau}_{MLE}$ and $\tilde{\tau}_{MOM}$ were computed:

```
nsim <- 10000
set.seed(234829)
xi <- runif(10, 1, 5)
T_mle <- (5+min(xi))/2
T_mom <- mean(xi)

t_MLE = NULL ; t_MOM = NULL
for(i in 1:nsim){
  boot <- sample(xi, 10, replace = T)
  t_MLE[i] <- (5+min(boot))/2
  t_MOM[i] <- mean(boot)
}


CI_mle <- c(quantile(t_MLE,.025), quantile(t_MLE, .975))
CI_mom <- c(quantile(t_MOM,.025),quantile(t_MOM,.975))
```