

## Lab Exercises and Solutions - Statistical Inference + Simulations

### Problem 1. (Computing the MME and MLE using Newton-Raphson Method)

- (a) Let  $\mathbf{X} = (X_1, \dots, X_n)$  be i.i.d. random variables with Kumaraswamy distribution with parameters  $a > 0$  and  $b > 0$ . The probability density function of the Kumaraswamy distribution is

$$f(x; a, b) = abx^{a-1} (1 - x^a)^{b-1}, \quad \text{where } x \in [0, 1].$$

Find the method of moments for the Kumaraswamy distribution using numerical methods.

- (b) Implement a function that takes as input a vector of data values  $X$ , performs the Newton-Raphson iterations to compute the MLEs  $\hat{\alpha}$  and  $\hat{\beta}$  in the Gamma  $(\alpha, \beta)$  model, and outputs  $\hat{\alpha}$  and  $\hat{\beta}$ . (You may use the form of the Newton-Raphson update equation derived in class. You may terminate the Newton-Raphson iterations when  $|\alpha^{(t+1)} - \alpha^{(t)}|$  is sufficiently small.)
- (c) For  $n = 500$ , use your function from part (a) to simulate the sampling distributions of  $\hat{\alpha}$  and  $\hat{\beta}$  computed from  $X_1, \dots, X_n \stackrel{\text{iid}}{\sim} \text{Gamma}(1, 2)$ . Plot histograms of the values of  $\hat{\alpha}$  and  $\hat{\beta}$  across 5000 simulations, and report the simulated mean and variance of  $\hat{\alpha}$  and  $\hat{\beta}$  as well as the simulated covariance between  $\hat{\alpha}$  and  $\hat{\beta}$ . Compute the inverse of the Fisher Information matrix  $I(\alpha, \beta)$  at  $\alpha = 1$  and  $\beta = 2$ . Do your simulations support that  $(\hat{\alpha}, \hat{\beta})$  is approximately distributed as  $\mathcal{N}((1, 2), \frac{1}{n}I(1, 2)^{-1})$ ? (You may use the formula for the Fisher information matrix  $I(\alpha, \beta)$  and/or its inverse derived in class.)

### Solution.

- (a) The Kumaraswamy distribution is an alternative to the Beta distribution, which offers similar flexibility in terms of the shape of the density. An appealing feature of the Kumaraswamy distribution is that it only involves algebraic terms, in contrast to the Beta distribution, which requires the evaluation of the Beta function. The mean and variance of this distribution are

$$\mu_1 = E[X] = \frac{b\Gamma(1 + \frac{1}{a})\Gamma(b)}{\Gamma(1 + \frac{1}{a} + b)},$$

$$\text{Var}[X] = E[X^2] - E[X]^2 = \mu_2 - \mu_1^2 = \frac{b\Gamma(1 + 2/a)\Gamma(b)}{\Gamma(1 + b + 2/a)} - \mu_1^2.$$

The MME of  $a$  and  $b$ ,  $\tilde{a}$  and  $\tilde{b}$ , are the solution to the system of non-linear equations,

$$\bar{\mathbf{X}} = \frac{b\Gamma(1 + \frac{1}{a})\Gamma(b)}{\Gamma(1 + \frac{1}{a} + b)},$$

$$\frac{1}{n} \sum_{i=1}^n X_i^2 - \bar{\mathbf{X}}^2 = \frac{b\Gamma(1 + 2/a)\Gamma(b)}{\Gamma(1 + b + 2/a)} - \mu_1^2.$$

which cannot be found in closed form. Thus, we can only obtain the MME using numerical methods. The following R code shows how to calculate the MME numerically.

```
rm(list = ls()) # Delete memory
# Moments of order n of the Kumaraswamy(a,b) distribution
# https://en.wikipedia.org/wiki/Kumaraswamy_distribution

mn <- function(a,b,n){
  log.num <- log(b) + lgamma(1 + n/a) + lgamma(b)
  log.den <- lgamma(1 + b + n/a)
  return(exp(log.num-log.den))
}
```

```

#-----
# Method of Moments Estimates
#-----
library(nleqslv) # Required packages

# Function containing the implementation of the MME
# It solves a system of nonlinear equations

MME <- function(data){
  x.bar <- mean(data)
  x2.bar <- mean(data^2)

  fn <- function(par){
    mom1 <- mn(exp(par[1]),exp(par[2]),1) - x.bar
    mom2 <- mn(exp(par[1]),exp(par[2]),2) - mn(exp(par[1]),exp(par[2]),1)^2 - x2.bar + x.bar^2
    return(c(mom1,mom2))
  }

  est <- as.vector(exp(nleqslv(c(0,0), fn)$x))

  return(est)
}

#####
# Examples
#####

#-----
# Example 1
#-----
a0 <- 1
b0 <- 1
ns <- 1000

# Simulation of Kumaraswamy distributed data
set.seed(123)
dat <- (rbeta(ns, shape1 = 1, shape2 = b0))^(1/a0)

# MME
MME(dat)
## [1] 1.018995 1.032939
#-----
# Example 2
#-----
a0 <- 0.5
b0 <- 2
ns <- 1000

# Simulation of Kumaraswamy distributed data
set.seed(123)
dat <- (rbeta(ns, shape1 = 1, shape2 = b0))^(1/a0)

# MME
MME(dat)
## [1] 0.5059483 2.1418411
#-----

```

```

# Example 3
#-----
a0 <- 5
b0 <- 3
ns <- 1000

# Simulation of Kumaraswamy distributed data
set.seed(123)
dat <- (rbeta(ns, shape1 = 1, shape2 = b0))^(1/a0)

# MME
MME(dat)
## [1] 4.796745 2.915643

```

(b) We denote the function

$$f(\alpha) = \log \alpha - \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} = \log \alpha - \psi(\alpha),$$

where  $\psi(\alpha) = \Gamma'(\alpha)/\Gamma(\alpha)$  is the digamma function. Its derivative is

$$f'(\alpha) = \frac{1}{\alpha} - \psi'(\alpha),$$

where  $\psi'(\alpha)$  is the trigamma function.

The Newton-Raphson update rule is

$$\alpha^{(t+1)} = \alpha^{(t)} + \frac{-f(\alpha^{(t)}) + \log \bar{X} - \frac{1}{n} \sum_{i=1}^n \log X_i}{f'(\alpha^{(t)})}.$$

We can implement this as follows:

```

gamma.MLE = function(X) {
  ahat = compute.ahat(X)
  bhat = ahat/mean(X)

  return(c(ahat,bhat))
}

# estimate ahat by Newton-Raphson
compute.ahat = function(X) {
  a.prev = -Inf
  a = mean(X)^2/var(X) #initialguess

  #while not converged, do Newton-Raphson update
  while(abs(a-a.prev)>1e-12) {
    a.prev = a
    numerator = -f(a.prev)+log(mean(X))-mean(log(X))
    denominator = f.prime(a.prev)
    a = a.prev+numerator/denominator
  }

  return(a)
}

#define some helper functions
f=function(alpha) {

  return(log(alpha)-digamma(alpha))
}

```

```
f.prime=function(alpha) {
  return(1/alpha-trigamma(alpha))
}
```

(c) We run the simulation:

*# In R you may simulate samples from Gamma (alpha, beta) using*

```
X = rgamma(n , alpha, rate = beta )
```

*# The sample variance of a vector of values of X is given by var(X), and the sample covariance  
# between two vectors of values X and Y (of the same length) is given by cov(X, Y).*

```
n=500
```

```
n.reps=5000
```

```
alpha=1
```

```
beta=2
```

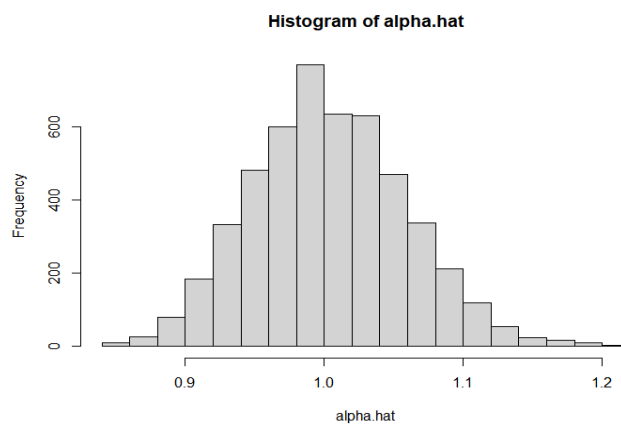
```
alpha.hat=numeric(n.reps)
```

```
beta.hat=numeric(n.reps)
```

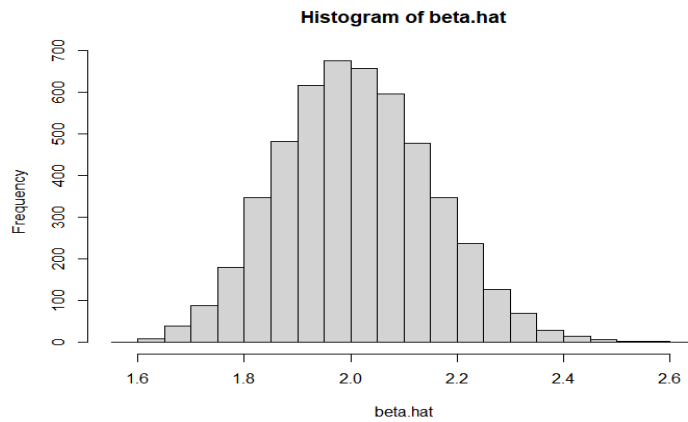
```
for(i in 1:n.reps) {
  X=rgamma(n,shape=alpha,rate=beta)
  estimates=gamma.MLE(X)
  alpha.hat[i] = estimates[1]
  beta.hat[i] = estimates[2]
}
```

Here are the resulting histograms:

```
hist(alpha.hat, breaks=20)
```



```
hist(beta.hat, breaks=20)
```



The inverse of the Fisher information matrix was computed to be

$$I(\alpha, \beta)^{-1} = \frac{1}{\psi'(\alpha)\frac{\alpha}{\beta^2} - \frac{1}{\beta^2}} \begin{pmatrix} \frac{\alpha}{\beta^2} & \frac{1}{\beta} \\ \frac{1}{\beta} & \psi'(\alpha) \end{pmatrix}.$$

Plugging in  $\alpha = 1, \beta = 2$  gives

$$I(1, 2)^{-1} = \frac{4}{\psi'(1) - 1} \begin{pmatrix} 1/4 & 1/2 \\ 1/2 & \psi'(1) \end{pmatrix} \approx \begin{pmatrix} 1.551 & 3.101 \\ 3.101 & 10.202 \end{pmatrix}$$

using  $\psi'(1) = \pi^2/6 \approx 1.645$ .

Now, let's take a look at the empirical moments produced by the simulation. Here are the means:

```
mean(alpha.hat) # should be close to 1
## [1] 1.003308

mean(beta.hat) # should be close to 2
## [1] 2.010126
```

They are close to the true values of  $\alpha = 1$  and  $\beta = 2$ , as expected. Now here are the variance and covariance terms:

```
var(alpha.hat)
## [1] 0.003079451

var(beta.hat)
## [1] 0.02048946

cov(alpha.hat, beta.hat)
## [1] 0.006190339
```

In order to compare to the Fisher information, we need to scale by  $n$ .

```
n*var(alpha.hat) #should be close to 1.551
## [1] 1.539726

n*var(beta.hat) #should be close to 10.202
## [1] 10.24473

n*cov(alpha.hat, beta.hat) #should be close to 3.101
## [1] 3.09517
```

**Problem 2. (MLE in a misspecified model)**

Suppose you fit the model  $\text{Exponential}(\lambda)$  to data  $X_1, \dots, X_n$  by computing the MLE  $\hat{\lambda} = 1/\bar{X}$ , but the true distribution of the data is  $X_1, \dots, X_n \stackrel{\text{IID}}{\sim} \text{Gamma}(2, 1)$ .

- (a) Let  $f(x | \lambda)$  be the PDF of the  $\text{Exponential}(\lambda)$  distribution, and let  $g(x)$  be the PDF of the  $\text{Gamma}(2, 1)$  distribution. Compute an explicit formula for the KL-divergence  $D_{\text{KL}}(g(x) || f(x | \lambda))$  in terms of  $\lambda$ , and find the value  $\lambda^*$  that minimizes this KL-divergence.

(You may use the fact that if  $X \sim \text{Gamma}(\alpha, \beta)$ , then  $\mathbb{E}[X] = \alpha/\beta$  and  $\mathbb{E}[\log X] = \psi(\alpha) - \log \beta$  where  $\psi$  is the digamma function.)

- (b) Show directly, using the Law of Large Numbers and the Continuous Mapping Theorem, that the MLE  $\hat{\lambda}$  converges in probability to  $\lambda^*$  as  $n \rightarrow \infty$ .
- (c) Perform a simulation study for the standard error of  $\hat{\lambda}$  with sample size  $n = 500$ , as follows: In each of  $B = 10000$  simulations, sample  $X_1, \dots, X_n \stackrel{\text{IID}}{\sim} \text{Gamma}(2, 1)$ , compute the MLE  $\hat{\lambda} = 1/\bar{X}$  for the exponential model, compute an estimate of the standard error of  $\hat{\lambda}$  using the Fisher information  $I(\hat{\lambda})$ , and compute also the sandwich estimate of the standard error,  $S_X / (\bar{X}^2 \sqrt{n})$ .

Report the true mean and standard deviation of  $\hat{\lambda}$  that you observe across your 10000 simulations. Is the mean close to  $\lambda^*$  from part (a)? Plot a histogram of the 10000 estimated standard errors using the Fisher information, and also plot a histogram of the 10000 estimated standard errors using the sandwich estimate. Do either of these methods for estimating the standard error of  $\hat{\lambda}$  seem accurate in this setting?

**Solution.**

- (a) The KL-divergence is given by

$$\begin{aligned} D_{\text{KL}}(g(x) || f(x | \lambda)) &= \mathbb{E}_g \left[ \log \frac{g(X)}{f(X | \lambda)} \right] \\ &= \mathbb{E}_g \left[ \log \frac{\frac{1}{\Gamma(2)} X e^{-X}}{\lambda e^{-\lambda X}} \right] \\ &= \mathbb{E}_g [-\log \Gamma(2) + \log X - X - \log \lambda + \lambda X] \\ &= -\log \Gamma(2) - \log \lambda + \mathbb{E}_g[\log X] + (\lambda - 1) \mathbb{E}_g[X] \\ &= -\log \Gamma(2) - \log \lambda + \psi(2) + 2(\lambda - 1). \end{aligned}$$

Setting the derivative with respect to  $\lambda$  equal to 0, is minimized at  $\lambda^* = 1/2$ .

- (b) By the Law of Large Numbers,  $\bar{X} \rightarrow \mathbb{E}_g[X] = 2$  in probability, so  $\hat{\lambda} = 1/\bar{X} \rightarrow 1/2$  in probability by the Continuous Mapping Theorem.
- (c) The Fisher information in the exponential model is given by

$$I(\lambda) = -\mathbb{E}_\lambda \left[ \frac{\partial^2}{\partial \lambda^2} \log f(X | \lambda) \right] = -\mathbb{E}_\lambda \left[ \frac{\partial^2}{\partial \lambda^2} (\log \lambda - \lambda X) \right] = 1/\lambda^2$$

The corresponding plug-in estimate of the standard error is  $\sqrt{\frac{1}{nI(\hat{\lambda})}} = \frac{1}{\bar{X}\sqrt{n}}$ .

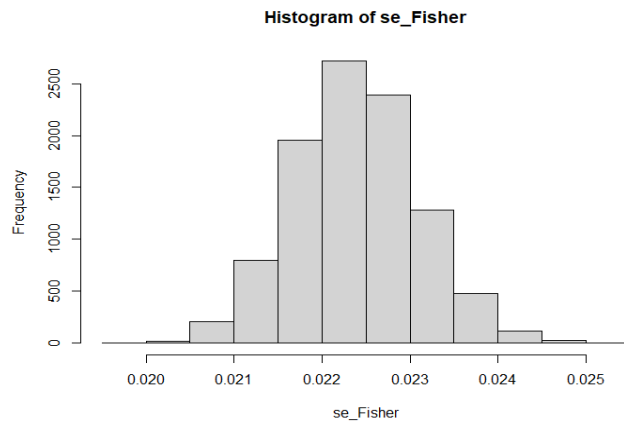
```
n = 500
B = 10000
lamb_hat = numeric(B)
se_Fisher = numeric(B)
se_sandwich = numeric(B)
```

```

for(i in 1:B){
  X = rgamma(n,2,rate=1)
  lamb_hat[i] = 1/mean(X)
  se_Fisher[i] = 1/(mean(X)*sqrt(n))
  se_sandwich[i] = sd(X)/(mean(X)^2*sqrt(n))
}
print(mean(lamb_hat))
## [1] 0.5006192
print(sd(lamb_hat))
## [1] 0.01595704

hist(se_Fisher)

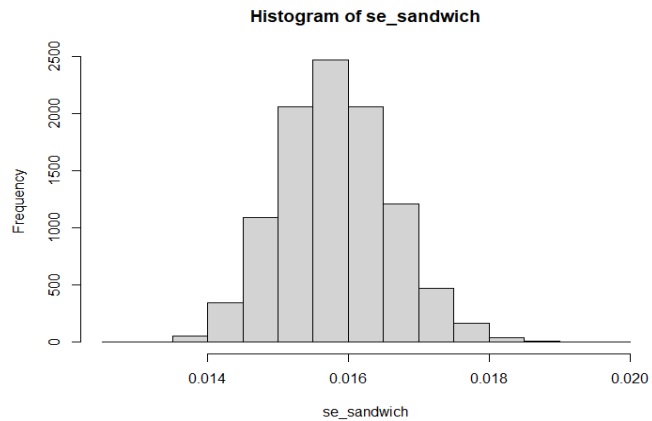
```



```

hist(se_sandwich)

```



The empirical mean and standard error of  $\hat{\lambda}$  are 0.501 and 0.016; the mean is close to  $\lambda^* = 0.5$  from part (a). The Fisher-information-based estimate of the standard error is incorrect – it estimates the standard error as approximately 0.022. The sandwich estimate of the standard error seems correct – it estimates the standard error as 0.016, with some variability in the third decimal place.

**Problem 3. (Confidence intervals for a binomial proportion)**

Let  $X_1, \dots, X_n \stackrel{IID}{\sim} \text{Bernoulli}(p)$  be  $n$  tosses of a biased coin, and let  $\hat{p} = \bar{X}$ . In this problem we will explore two different ways to construct a 95% confidence interval for  $p$ , both based on the Central Limit Theorem result

$$\sqrt{n}(\hat{p} - p) \rightarrow \mathcal{N}(0, p(1-p)). \quad (1)$$

- (a) Use the plugin estimate  $\hat{p}(1 - \hat{p})$  for the variance  $p(1 - p)$  to obtain a 95% confidence interval for  $p$ . (This is the procedure discussed in class, yielding the Wald interval for  $p$ .)
- (b) Instead of using the plugin estimate  $\hat{p}(1 - \hat{p})$ , note that Eqn. (1) implies, for large  $n$ ,

$$\mathbb{P}_p \left[ -\sqrt{p(1-p)}Z_{\alpha/2} \leq \sqrt{n}(\hat{p} - p) \leq \sqrt{p(1-p)}Z_{\alpha/2} \right] \approx 1 - \alpha.$$

Solve the equation  $\sqrt{n}(\hat{p} - p) = \sqrt{p(1-p)}Z_{\alpha/2}$  for  $p$  in terms of  $\hat{p}$ , and solve the equation  $\sqrt{n}(\hat{p} - p) = -\sqrt{p(1-p)}Z_{\alpha/2}$  for  $p$  in terms of  $\hat{p}$ , to obtain a different 95% confidence interval for  $p$ .

- (c) Perform a simulation study to determine the true coverage of the confidence intervals in parts (a) and (b), for the 9 combinations of sample size  $n = 10, 40, 100$  and true parameter  $p = 0.1, 0.3, 0.5$ . (For each combination, perform at least  $B = 100,000$  simulations. In each simulation, you may simulate  $\hat{p}$  directly from  $\frac{1}{n} \text{Binomial}(n, p)$  instead of simulating  $X_1, \dots, X_n$ .) Report the simulated coverage levels in two tables. Which interval yields true coverage closer to 95% for small values of  $n$ ?

**Solution.**

- (a)  $\hat{p} \rightarrow p$  in probability, hence  $\frac{\sqrt{n}(\hat{p} - p)}{\sqrt{\hat{p}(1-\hat{p})}} \rightarrow \mathcal{N}(0, 1)$  in distribution by Slutsky's Lemma. So for large  $n$

$$\mathbb{P} \left[ -Z_{0.025} \leq \frac{\sqrt{n}(\hat{p} - p)}{\sqrt{\hat{p}(1-\hat{p})}} \leq Z_{0.025} \right] \approx 0.95.$$

We may rewrite the above as

$$\mathbb{P} \left[ \hat{p} - \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}Z_{0.025} \leq p \leq \hat{p} + \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}Z_{0.025} \right] \approx 0.95,$$

so an approximate 95% confidence interval is  $\hat{p} \pm \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}Z_{0.025}$ .

- (b) The following conditions are equivalent:

$$\begin{aligned} -\sqrt{p(1-p)}Z_{\alpha/2} &\leq \sqrt{n}(\hat{p} - p) \leq \sqrt{p(1-p)}Z_{\alpha/2} \\ &\Downarrow \\ n(\hat{p} - p)^2 &\leq p(1-p)Z_{\alpha/2}^2 \\ &\Downarrow \\ (n + Z_{\alpha/2}^2)p^2 - (2n\hat{p} + Z_{\alpha/2}^2)p + n\hat{p}^2 &\leq 0. \end{aligned}$$

This occurs when  $p$  is between the two real roots of the above quadratic equation, which are given by

$$\frac{2n\hat{p} + Z_{\alpha/2}^2 \pm \sqrt{(2n\hat{p} + Z_{\alpha/2}^2)^2 - 4(n + Z_{\alpha/2}^2)n\hat{p}^2}}{2(n + Z_{\alpha/2}^2)}.$$

Taking  $\alpha = 0.05$  and simplifying the above, we obtain an approximate 95% confidence interval of

$$\frac{\hat{p} + \frac{Z_{0.025}^2}{2n} \pm Z_{0.025} \sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{Z_{0.025}^2}{4n^2}}}{1 + \frac{Z_{0.025}^2}{n}}.$$



```

(c) ns = c(10,40,100)
ps = c(0.1,0.3,0.5)
B=100000
z = qnorm(0.975)
for (n in ns) {
  for (p in ps) {
    cover_A = numeric(B)
    cover_B = numeric(B)
    for (i in 1:B) {
      phat = rbinom(1,n,p)/n
      U = phat+z*sqrt(phat*(1-phat)/n)
      L = phat-z*sqrt(phat*(1-phat)/n)
      if (p <= U && p >= L) {
        cover_A[i] = 1
      } else {
        cover_A[i] = 0
      }
      U = (phat+z^2/(2*n)+z*sqrt(phat*(1-phat)/n+z^2/(4*n^2)))/(1+z^2/n)
      L = (phat+z^2/(2*n)-z*sqrt(phat*(1-phat)/n+z^2/(4*n^2)))/(1+z^2/n)
      if (p <= U && p >= L) {
        cover_B[i] = 1
      } else {
        cover_B[i] = 0
      }
    }
    print(c(n,p,mean(cover_A),mean(cover_B)))
  }
}

```

For the interval from part (a), we obtain the following coverage probabilities:

	$p = 0.1$	$p = 0.3$	$p = 0.5$
$n = 10$	0.65	0.84	0.89
$n = 40$	0.91	0.93	0.92
$n = 100$	0.93	0.95	0.94

For the interval from part (b), we obtain the following coverage probabilities:

	$p = 0.1$	$p = 0.3$	$p = 0.5$
$n = 10$	0.93	0.92	0.98
$n = 40$	0.94	0.94	0.96
$n = 100$	0.94	0.94	0.94

The intervals from part (b) are more accurate when  $n$  is small.

#### Problem 4. (Power Comparisons)

Consider the problem of testing

$$\begin{aligned}H_0 : X_1, \dots, X_n &\stackrel{\text{iid}}{\sim} \mathcal{N}(0, 1) \\ H_1 : X_1, \dots, X_n &\stackrel{\text{iid}}{\sim} \mathcal{N}(\mu, 1)\end{aligned}$$

at significance level  $\alpha = 0.05$ , where  $\mu > 0$ . We've seen four tests that may be applied to this problem, summarized below:

- Likelihood ratio test: Reject  $H_0$  when  $\bar{X} > \frac{1}{\sqrt{n}} Z_{0.05}$ .
- $t$ -test: Reject  $H_0$  when  $T := \sqrt{n}\bar{X}/S > t_{n-1;0.05}$ , where  $S^2 = \frac{1}{n-1} \sum_i (X_i - \bar{X})^2$ .
- Wilcoxon signed rank test: Reject  $H_0$  when  $W_+ > \frac{n(n+1)}{4} + \sqrt{\frac{n(n+1)(2n+1)}{24}} Z_{0.05}$ , where  $W_+$  is the Wilcoxon signed rank statistic.
- Sign test (from Problem 4 of TD-3): Reject  $H_0$  when  $S > \frac{n}{2} + \sqrt{\frac{n}{4}} Z_{0.05}$ , where  $S$  is the number of positive values in  $X_1, \dots, X_n$ .

(For the Wilcoxon and sign test statistics, we are using the normal approximations for their null distributions.) These tests are successively more robust to violations of the  $\mathcal{N}(0, 1)$  distributional assumption imposed by  $H_0$ .

- For  $n = 100$ , verify numerically that these tests have significance level close to  $\alpha$ , in the following way: Perform 10,000 simulations. In each simulation, draw a sample of 100 observations from  $\mathcal{N}(0, 1)$ , compute the above four test statistics  $\bar{X}$ ,  $T$ ,  $W_+$ , and  $S$  on this sample, and record whether each test accepts or rejects  $H_0$ . Report the fraction of simulations for which each test rejected  $H_0$ , and confirm that these fractions are close to 0.05.
- For  $n = 100$ , numerically compute the powers of these tests against the alternative  $H_1$ , for the values  $\mu = 0.1, 0.2, 0.3$ , and 0.4. Do this by performing 10,000 simulations as in part (a), except now drawing each sample of 100 observations from  $\mathcal{N}(\mu, 1)$  instead of  $\mathcal{N}(0, 1)$ . (You should be able to re-use most of your code from part (a).) Report your computed powers either in a table or visually using a graph.
- How do the powers of the four tests compare, when testing against a normal alternative? Your friend says, "We should always use the testing procedure that makes the fewest distributional assumptions, because we never know in practice, for example, whether the variance is truly 1 or whether data is truly normal." Comment on this statement. Rice says, "It has been shown that even when the assumption of normality holds, the [Wilcoxon] signed rank test is nearly as powerful as the  $t$  test. The [signed rank test] is thus generally preferable, especially for small sample sizes." Do your simulated results support this conclusion?

#### Solution.

- The code below runs the simulations for the null case ( $\mu = 0$ ) as well as for  $\mu = 0.1, 0.2, 0.3, 0.4$ :

```
set.seed(1)
n = 100
B = 10000
for (mu in c(0,0.1,0.2,0.3,0.4)) {
  output.Z = numeric(B)
  output.T = numeric(B)
  output.W = numeric(B)
  output.S = numeric(B)
  for (i in 1:B) {
    X = rnorm(n, mean=mu, sd=1)
    if (mean(X) > 1/sqrt(n)*qnorm(0.95)) {
      output.Z[i] = 1
    } else {
      output.Z[i] = 0
    }
  }
  T = t.test(X)$statistic
```

```

if (T > qt(0.95,df=n-1)) {
  output.T[i] = 1
} else {
  output.T[i] = 0
}
W = wilcox.test(X)$statistic
if (W > n*(n+1)/4+sqrt(n*(n+1)*(2*n+1)/24)*qnorm(0.95)) {
  output.W[i] = 1
} else {
  output.W[i] = 0
}
S = length(which(X>0))
if (S > n/2+sqrt(n/4)*qnorm(0.95)) {
  output.S[i] = 1
} else {
  output.S[i] = 0
}
}
print(paste("mu = ", mu))
print(paste("Z: ", mean(output.Z)))
print(paste("T: ", mean(output.T)))
print(paste("W: ", mean(output.W)))
print(paste("S: ", mean(output.S)))
}

```

Under  $H_0$  (case  $\mu = 0$ ), we obtained the results:

Test stat	Type-I Error
Likelihood ratio test	0.0507
$t$ -test	0.0505
Wilcoxon signed rank test	0.053
Sign test	0.0441

(b) Under these alternatives, we obtained the results:

Test stat	Power over $\mathcal{N}(\mu, 1)$			
	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.3$	$\mu = 0.4$
Likelihood ratio test	0.2631	0.6356	0.913	0.9895
$t$ -test	0.261	0.6306	0.9085	0.9885
Wilcoxon signed rank test	0.252	0.6164	0.8978	0.9847
Sign test	0.1805	0.4617	0.7521	0.9318

(c) The powers of the tests against  $\mathcal{N}(\mu, 1)$  decrease as we increasingly relax the distributional assumptions (from  $\mathcal{N}(0, 1)$  to  $\mathcal{N}(0, \sigma^2)$  to any symmetric PDF  $f$  about 0 to any PDF  $f$  with median 0). The sign test makes the fewest distributional assumptions under  $H_0$ , but its power is substantially lower than the other three tests. Hence if we have good reason to believe that the data distribution under  $H_0$  is symmetric (for example, if each data value is the difference of paired samples  $(X_i, Y_i)$ , and  $(X_i, Y_i)$  should have the same distribution as  $(Y_i, X_i)$  under  $H_0$ ), then we should at least opt for using the Wilcoxon test. The difference in powers between the Wilcoxon test,  $t$ -test, and the most-powerful likelihood ratio test is indeed very small, which supports Rice's claim (at least for the tested sample size  $n = 100$ ).

**Problem 5.** (Testing gender ratios)

In a classical genetics study, Geissler (1889) studied hospital records in Saxony and compiled data on the gender ratio. The following table shows the number of male children in 6115 families having 12 children:

Number of male children	Number of families
0	7
1	45
2	181
3	478
4	829
5	1112
6	1343
7	1033
8	670
9	286
10	104
11	24
12	3

Let  $X_1, \dots, X_{6115}$  denote the number of male children in these 6115 families.

- (a) Suggest two reasonable test statistics  $T_1$  and  $T_2$  for testing the null hypothesis

$$H_0 : X_1, \dots, X_{6115} \stackrel{IID}{\sim} \text{Binomial}(12, 0.5).$$

(This is intentionally open-ended; try to pick  $T_1$  and  $T_2$  to “target” different possible alternatives to the above null.) Compute the values of  $T_1$  and  $T_2$  for the above data.

- (b) Perform a simulation to simulate the null distributions of  $T_1$  and  $T_2$ . (For example: Simulate 6115 independent samples  $X_1, \dots, X_{6115}$  from  $\text{Binomial}(12, 0.5)$ , and compute  $T_1$  on this sample. Do this 1000 times to obtain 1000 simulated values of  $T_1$ . Do the same for  $T_2$ .) Plot the histograms of the simulated null distributions of  $T_1$  and  $T_2$ . Using your simulated values, compute approximate  $p$ -values of the hypothesis tests based on  $T_1$  and  $T_2$ , for the above data. For either of your tests, can you reject  $H_0$  at significance level  $\alpha = 0.05$ ?
- (c) In this example, why might the null hypothesis  $H_0$  not hold? (Please answer this question regardless of your findings in part (b).)

**Solution.**

- (a) There are many possible answers. We may take  $T_1$  to be the average number of male children per family,

$$T_1 = \bar{X},$$

and perform a two-sided test based on  $T_1$  to check whether roughly half of the children are male. We may take  $T_2$  to be Pearson’s chi-squared statistic

$$T_2 = \sum_{k=0}^{12} (O_k - E_k)^2 / E_k$$

where  $O_k$  is the number of families with  $k$  male children and  $E_k$  is the expected number under the hypothesized binomial distribution, i.e.  $E_k = 6115 \times \binom{12}{k} (0.5)^{12}$ , and perform a one-sided test that rejects for large  $T_2$  to check whether the shape of the observed distribution of  $X_1, \dots, X_{6115}$  matches the shape of the binomial PDF.

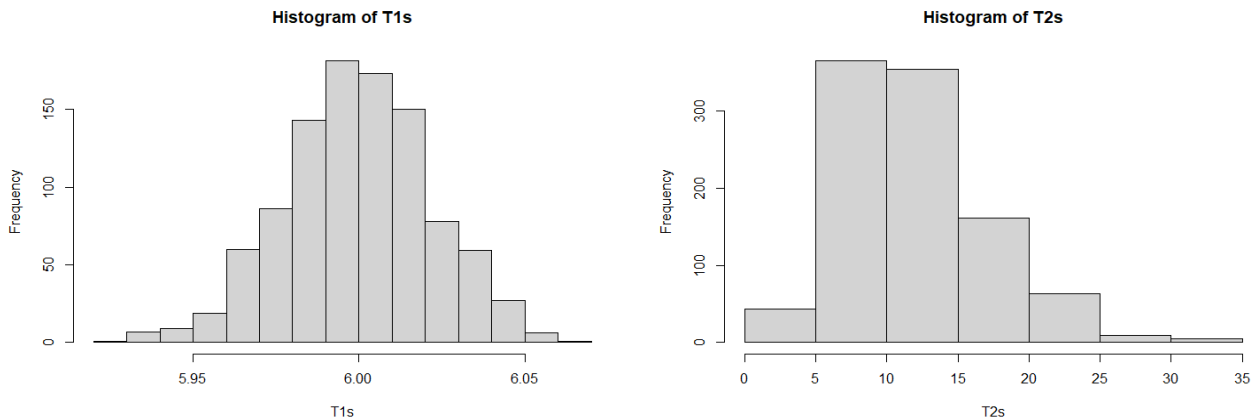
(b) R code corresponding to the above  $T_1$  and  $T_2$  is as follows:

```
ks = seq(0,12)
counts = c(7,45,181,478,829,1112,1343,1033,670,286,104,24,3)
expected = 6115*choose(12,ks)*(0.5^12)

T1_obs = sum(ks*counts)/6115
T2_obs = sum((counts-expected)^2/expected)

T1s = numeric(1000)
T2s = numeric(1000)
for (i in 1:1000) {
  X = rbinom(6115, 12, 0.5)
  T1s[i] = mean(X)
  counts = numeric(13)
  for (k in 0:12) {
    counts[k+1] = length(which(X==k))
  }
  T2s[i] = sum((counts-expected)^2/expected)
}
hist(T1s)
hist(T2s)
T1_pvalue = length(which(T1s<T1_obs))/1000 * 2
T2_pvalue = length(which(T2s>T2_obs))/1000
```

Histograms of the null distributions of  $T_1$  and  $T_2$  are below:



The values of the test statistics for the observed data are  $T_1 = 5.77$  and  $T_2 = 249$ , which are both far outside the range of the simulated null distributions above. The simulated  $p$ -values for the two tests are both  $< 0.001$ , and there is strong evidence that  $H_0$  is not correct.

(c) There may be both biological and sociological reasons why  $H_0$  is false. Biologically, the human male-to-female sex ratio at birth is not exactly 1 : 1. The probability  $p$  that a child is male might also vary from family to family. The sexes of children within a family might be dependent; in particular, one source of dependence is the presence of identical twins.

Sociologically, there may be a relationship between family size and the sex ratio of children in the family, because the current sex ratio influences parents' decision of whether to have another child. Note that the given data is only for families with 12 children, which is quite large even for that time. There is a noticeable bias towards families with more girls than boys, which may be explained if parents tended to continue having children when their current children were predominantly female.

### Problem 6. (Monte Carlo Simulations)

(a) Consider the problem of approximating tail probabilities of a Standard Normal distribution. Find  $\mathcal{P}(X > 2.04)$  for  $X \sim N(0, 1)$  using Monte Carlo methods. While we can easily calculate this in R using the `pnorm` function, check how a Monte Carlo method would perform. How does the estimate change with the number of Monte Carlo samples?

(b) Consider the problem of calculating the expected present value of the payoff of a call option on a stock. Let  $S(t)$  denote the price of the stock at time  $t$ . Consider a European call option, that is, the holder has the right to buy the stock at a fixed price  $K$  at a fixed time  $T$  in the future,  $t = 0$  being the current time. If  $S(T) > K$ , the holder exercises the option for a profit of  $S(T) - K$ . If  $S(T) \leq K$ , the option expires worthless. The payoff at time  $T$  is thus

$$\max\{S(T) - K, 0\}.$$

The expected present value of this payoff is

$$E \left[ e^{-rT} \max\{S(T) - K, 0\} \right],$$

where  $r$  is the compound interest rate. The evolution of the stock price  $S(t)$  can be modeled via the Black-Scholes model expressed as

$$\frac{dS(t)}{S(t)} = rdt + \sigma dW(t),$$

where  $W$  is a standard Brownian motion. The solution to the above Stochastic Differential Equation is given by

$$S(T) = S(0) \exp \left\{ \left( r - \frac{1}{2}\sigma^2 \right) T + \sigma W(T) \right\},$$

where  $S(0)$  is the current price of the stock and  $W(T) \sim N(0, T)$ , that is,  $W(T) = \sqrt{T}Z$ ,  $Z \sim N(0, 1)$ . Thus, the logarithm of  $S(T)$  is Normally distributed, or, in other words, the distribution of  $S(T)$  is log-normal. The expected payoff  $E \left[ e^{-rT} \max\{S(T) - K, 0\} \right]$  is thus an integral w.r.t. to the lognormal density.

Can you perform Monte Carlo sampling to estimate this integral (that is, find the expected payoff)? This will require generating samples from the standard Normal distribution, followed by computing the value of the function inside the expectation and then taking the average. Can you also compute the estimated standard error of the estimator?

### Solution.

```
(a) N <- 1e06 # number of samples
t <- 2.04
Xsamp <- rnorm(N) #simulating from N(0,1)
tail.prob <- mean(Xsamp > t)
exact.tail.prob <- pnorm(t, lower.tail = FALSE)
c(MC.estimate = tail.prob, Exact.value = exact.tail.prob)
# Standard error of the estimate
MC.se <- sqrt(var(Xsamp > t)/N)
MC.se

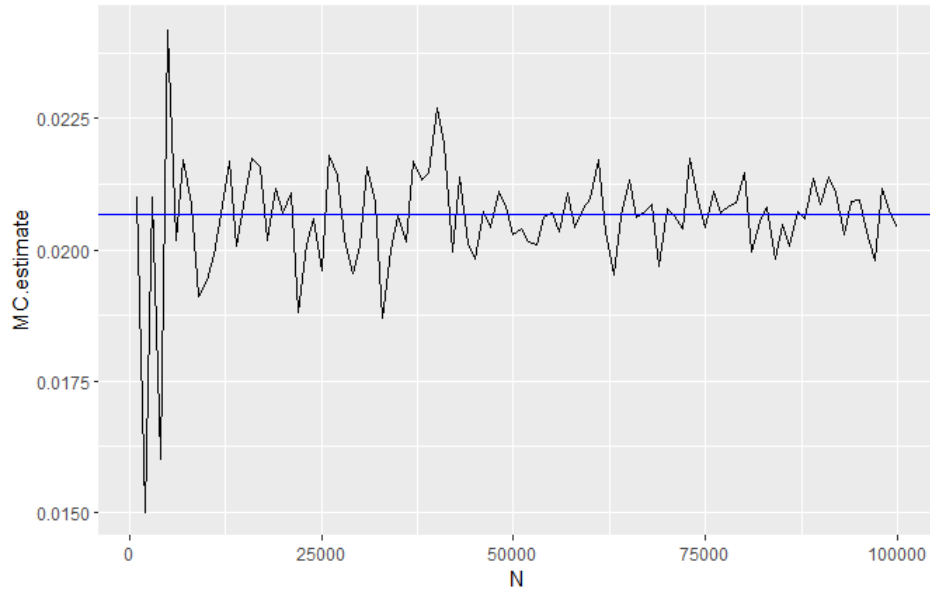
library(tibble)
library(ggplot2)

MC_func <- function(N){
  Xsamp <- rnorm(N) # simulate from N(0,1)
  tail.prob <- mean(Xsamp > t)
  MC.se <- sqrt(var(Xsamp > t)/N)
  c(N = N, MC.estimate = tail.prob, MC.se = MC.se, Exact.value = exact.tail.prob)
}
N <- seq(1000, 100000, by = 1000)
```

```

out <- NULL
for (i in N) {
  out <- rbind(out, MC_func(i))
}
out <- as_tibble(out)
ggplot(out, mapping = aes(N, MC.estimate))
+ geom_line() + geom_hline(yintercept = exact.tail.prob, color = "blue")

```



```

(b) payoff_func <- function(S_0, r, s, T, K, N=1e06){
  Z <- rnorm(N)
  S_T <- S_0*exp((r - s^2/2)*T + s*sqrt(T)*Z)
  payoff <- exp(-r*T)*pmax((S_T - K), 0) # pmax function in the formula
  ex_payoff <- mean(payoff)
  se_ex_payoff <- sqrt(var(payoff)/N)
  out <- c(expected.payoff = ex_payoff, SE = se_ex_payoff)
  return(out)
}
payoff_func(S_0 = 100, r = 0.07, s = 0.3, T = 5, K = 120)

```