

MTH3028 Statistical Inference: Theory and Practice CourseWork

690011024

30/12/2021

1.a.i)

$X = (1, 9, 5, 2, 1, 0, 1, 24, 3, 0, 4, 12, 2, 1, 1, 14, 7, 2, 4, 0)$

$\therefore n = 20$ and $\mu^{\wedge} = \overline{X} = \frac{1}{n}$

$P_{n=1}[X_i]$

and $\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n [(X_i - \overline{X})^2]$

$\therefore \hat{\sigma}^2 = \frac{\frac{1}{n} \sum_{i=1}^n P_n[X_i]}{\frac{1}{n-1} \sum_{i=1}^n [(X_i - \overline{X})^2]}$

```
X_vals = c(1, 9, 5, 2, 1, 0, 1, 24, 3, 0, 4, 12, 2, 1, 1, 14, 7, 2, 4, 0)
```

```
mu_hat <- mean(X_vals) mu_hat
```

```
## [1] 4.65
```

```
n_of_X_vals <- length(X_vals)
```

```
X_vals_minus_X_bar = numeric(n_of_X_vals) for (i in 1:
```

```
n_of_X_vals) {
```

```
  X_vals_minus_X_bar[i] = X_vals[i] - mu_hat
```

```
}
```

```
X_vals_minus_X_bar
```

```
## [1] -3.65 4.35 0.35 -2.65 -3.65 -4.65 -3.65 19.35 -1.65 -4.65 -0.65 7.35 ## [13] -2.65 -3.65 -3.65 9.35  
2.35 -2.65 -0.65 -4.65
```

```
X_vals_minus_X_bar_all_squared = numeric(n_of_X_vals) for (i in 1:
```

```
n_of_X_vals) {
```

```
  X_vals_minus_X_bar_all_squared[i] = (X_vals_minus_X_bar[i])^2
```

```
}
```

```
X_vals_minus_X_bar_all_squared
```

```
## [1] 13.3225 18.9225 0.1225 7.0225 13.3225 21.6225 13.3225 374.4225
```

```
## [9] 2.7225 21.6225 0.4225 54.0225 7.0225 13.3225 13.3225 87.4225
```

```
## [17] 5.5225 7.0225 0.4225 21.6225
```

```
sigma_hat_squared = (1/(n_of_X_vals- 1))*sum(X_vals_minus_X_bar_all_squared) sigma_hat_squared
```

```
## [1] 36.66053
```

```
phi_hat = mu_hat/sigma_hat_squared phi_hat
```

```
## [1] 0.1268394 =
```

$\hat{\varphi} \approx 0.127(3.d.p.)$

1.a.ii)

$$\hat{\varphi} = \frac{\hat{\mu}_2}{\sigma^2} = \frac{Mean(X)}{Var(X)} = \frac{E[X^2]}{E[X]^2}$$

$$\text{So } P_{j6=i}[X_j] = P_{nj=1}[X_j] - X_i$$

$$\text{So } \frac{1}{n-1} P_{j6=i}[X_j] = \frac{1}{n-1} (P_{nj=1}[X_j] - X_i) \therefore \frac{1}{n-1} P_{j6=i}[X_j] = \frac{1}{n-1} (P_{nj=1}[X_j] - X_i)$$

$$\frac{1}{n-1} (P_{nj=1}[X_j] - X_i)$$

$$\frac{1}{n-1} P_{nj=1}[X_j]$$

$$\therefore \hat{\varphi}_{-i} = \frac{Mean(X_{-i})}{Var(X_{-i})} = \frac{E[X_{-i}^2]}{E[X_{-i}]^2}$$

$$\therefore \hat{\varphi}_i = n\hat{\varphi} - (n-1)\hat{\varphi}_{-i}$$

$$\therefore \text{The jackknife version of } \hat{\varphi} \text{ is: } \hat{\varphi}_J = n \frac{1}{n-1} P_{i=1} [\hat{\varphi}_{-i}]$$

```
phi_hat_i_function = function(x) { phi_hat_of_function =
  mean(x)/var(x) n_of_function = length(x)
  phi_hat_negative_i = numeric(n_of_function) for (i in
  1:n_of_function) { phi_hat_negative_i[i] = mean(x[-i])/var(x[-
  i])
  }
  phi_hat_negative_i
  phi_hat_i_of_function = n_of_function*phi_hat_of_function - (n_of_function-
  1)*phi_hat_negative_i
  phi_hat_i_of_function
}

# X_vals = c(1, 9, 5, 2, 1, 0, 1, 24, 3, 0, 4, 12, 2, 1, 1, 14, 7, 2, 4, 0)

phi_hat_i = phi_hat_i_function(X_vals) phi_hat_i

## [1] 0.1105084 0.3021898 0.2623024 0.1599746 0.1105084 0.0523329
## [7] 0.1105084 -1.5700685 0.2014319 0.0523329 0.2354124 0.2575388
## [13] 0.1599746 0.1105084 0.1105084 0.1845335 0.2957034 0.1599746
## [19] 0.2354124 0.0523329
```

```
phi_hat_J = mean(phi_hat_i_function(X_vals)) phi_hat_J
```

```
## [1] 0.07969603
```

1.a.iii)

MonteCarlo resampling so use 10000 resamples.

```
B = 10000 m =
length(X_vals) t_star =
numeric(B)

for(b in 1:B) {
  X_star = sample(X_vals, replace = TRUE) # resample t_star[b] = mean(X_star)/((1/(n_of_X_vals-
  1))*sum((X_star - mean(X_vals))^2))- phi_hat #
} phi_hat- mean(t_star) # bias-corrected estimate

calculate T*
```

```
## [1] 0.103772
```

Is the nonparametric bootstrap bias-corrected version of $\hat{\varphi}$

1.b.i)

$$\sqrt{V\hat{\varphi}_J} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{\varphi}_i - \hat{\varphi}_J)^2}$$

So two equal-tailed 90% confidence interval has: $\alpha = 5\%$ for the student distribution with degrees of freedom

$$= df = v = n - 1 = 20 - 1 = 19 \therefore Stu(n - 1) = t_{\alpha} = t_{\alpha;v} = t_{0.05;19} \approx 1.73(3.d.p.)$$

$$\text{So two equal-tailed 90\% confidence interval is: } \hat{\varphi}_J \pm t_{\alpha} \sqrt{V\hat{\varphi}_J} = (\hat{\varphi}_J - t_{\alpha} \sqrt{V\hat{\varphi}_J}, \hat{\varphi}_J + t_{\alpha} \sqrt{V\hat{\varphi}_J})$$

```
n = length(X_vals)
```

```
phi_hat_i_minus_phi_hat_J_all_squared = numeric(n)
```

```
for (i in 1: n) {
  phi_hat_i_minus_phi_hat_J_all_squared[i] = (phi_hat_i[i] - phi_hat_J)^2
} phi_hat_i_minus_phi_hat_J_all_squared
```

```
## [1] 0.0009494009 0.0495034876 0.0333450713 0.0064446561 0.0009494009
```

```
## [6] 0.0007487409 0.0009494009 2.7217229740 0.0148196104 0.0007487409
```

```
## [11] 0.0242475803 0.0316280592 0.0064446561 0.0009494009 0.0009494009
```

```
## [16] 0.0109909023 0.0466591917 0.0064446561 0.0242475803 0.0007487409
```

```
v_hat_J_SquareRoot = sqrt((1/(n*(n-1)))*sum(phi_hat_i_minus_phi_hat_J_all_squared)) v_hat_J_SquareRoot
```

```
## [1] 0.08860753
```

```
student_quantile_95 = qt(0.95, df = n-1) student_quantile_95
```

```
## [1] 1.729133
```

```
ConfidenceInterval_90_percent = c(phi_hat_J- student_quantile_95*v_hat_J_SquareRoot, phi_hat_J+
                                   student_quantile_95*v_hat_J_SquareRoot)
ConfidenceInterval_90_percent
```

```
## [1] -0.07351815 0.23291022
```

Is the equal-tailed 90% confidence interval for φ using jackknife.

1.b.ii)

$$\hat{q}_p = t_{*pB}$$

So the nonparametric basic bootstrap equal-tailed 90% confidence interval is: $(\hat{\varphi} - \hat{q}_{1-\alpha}, \hat{\varphi} - \hat{q}_{\alpha})$

```
phi_hat- quantile(t_star, c(0.95, 0.05)) # basic
```

```
##           95%           5%
## -0.006988827 0.168044444
```

Is the equal-tailed 90% confidence interval for φ using the nonparametric basic bootstrap.

1.c.i)

Conduct hypothesis $H_0: \varphi = 1$ against $H_1: \varphi < 1$ at 1% significance level

$$\therefore \varphi_0 = 1$$

The null distribution of the test statistic $\frac{\hat{\varphi} - \varphi_0}{\hat{\varphi}}$ $\approx Stu(n-1)$ distribution.

There is significant evidence to reject the null hypothesis $\varphi = 1$ if the p-value < 0.01 because it is a one-tailed test.

So:

```
phi_0 = 1
test_statistic = (phi_hat_J - phi_0)/v_hat_J_SquareRoot test_statistic
```

```
## [1] -10.3863
```

```
degrees_of_freedom = n-1
```

```
p_value = 1 - pt(abs(test_statistic), degrees_of_freedom) p_value
```

```
## [1] 1.424148e-09
```

```
p_value < 0.01
```

```
## [1] TRUE
```

So the p-value is less than 0.01 so there is significant evidence to reject the null hypothesis H_0 that $\varphi = 1$.

So there is significant evidence that $\varphi < 1$

1.c.ii)

Test hypothesis $H_0: \varphi = 1$ against $H_1: \varphi < 1$ at 1% significance level When

$\varphi = 1: X_1, \dots, X_n \sim \text{Poi}(\mu)$ So $X_i \geq 0$ and $E[X] = \text{Var}[X]$ $n = 20 \therefore \hat{\mu} = 4.65 \therefore X \sim$

$\text{Poi}(\mu)$

$\therefore \text{Pr}(X_i; \mu) = \frac{\mu^{X_{ii}}}{X_{ii}!} e^{-\mu}$ for $X_i \geq 0$

So: $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n X_i$

$\therefore \hat{\mu} = \bar{X}$

test statistic $\hat{\varphi} = \frac{\frac{1}{n} \sum_{i=1}^n X_i^2}{\left(\frac{1}{n} \sum_{i=1}^n X_i\right)^2} \approx 0.127$

And reject H_0 in favour of H_1 at level α if the test statistic exceeded the $(1 - \alpha)$ -quantile So:

```
B = 10000 n =
length(X_vals) t_star <-
numeric(B)

for (i in 1:B) {
  x <- rpois(n, mu_hat) t_star[i] <-
  mean(x)/var(x)
} quantile(t_star, 0.01) # The Hypothesis test Critical value for T

##          1%
## 0.5217759
```

```
mean(t_star < 1)
```

```
## [1] 0.4481
```

```
mean(t_star < 1) < quantile(t_star, 0.01)
```

```
##      1%
## TRUE
```

With critical value being:

```
##      1%
## 0.5217759
```

So the test statistic is less than the critical value so there is significant evidence to reject the null hypothesis H_0 that $\varphi = 1$.

So there is significant evidence that $\varphi < 1$.

1.d)

$X \geq 0$ So if: $T = T(X_0, \underline{X}) \therefore T \geq 0$

So: $Pr(t_5 < T < t_{95}) = 0.90$

The following code calculates a bootstrap 90% prediction interval for X_0 on a sample of size $n = 20$ Estimate the parameters, $\mu^{\wedge} = \frac{1}{n} \sum_{i=1}^n [X_i]$

and $\varphi^{\wedge} = \frac{\sum_{i=1}^n P_{n-1}[(X_i - \mu^{\wedge})^2]}{n-1}$

```
B = 100000 n =
length(X_vals)

t_star = numeric(B) for (b in
1:B) {
  x_star = rnbino(n, mu_hat * phi_hat / (1 - phi_hat), phi_hat) # resample X1, ..., Xn
  x0_star = rnbino(1,
mu_hat * phi_hat / (1 - phi_hat), phi_hat) # resample X0
  mu_star = mean(x_star) # estimate mu
  phi_star = mean(x_star) / var(x_star) # estimate phi
  t_star[b] = pnbino(x0_star, mu_hat * phi_hat / (1 - phi_hat), phi_hat) # calculate T*
} qnbino(quantile(t_star, c(0.05, 0.95)), mu_hat * phi_hat / (1 - phi_hat), phi_hat)
```

#

interval

```
## 5% 95%
##      0 17
```

So: [0,17] is the plug-in 90% prediction interval for X_0 and is a closed interval of the form [0,x] as required.

Because an ancillary statistic is unavailable, so the distribution of the ancillary statistic was unknown and instead the probability integral transform (PIT) needed to be used to choose a T that is approximately ancillary (That is, choosing T to have a distribution that is approximately independent of the model parameters), means that the exact prediction interval can not be derived.

A bootstrap error arises since the uncertainty of the parameter estimate is not considered, and from B having to be finite, which is another reason why the coverage of the interval may be less than 90%.

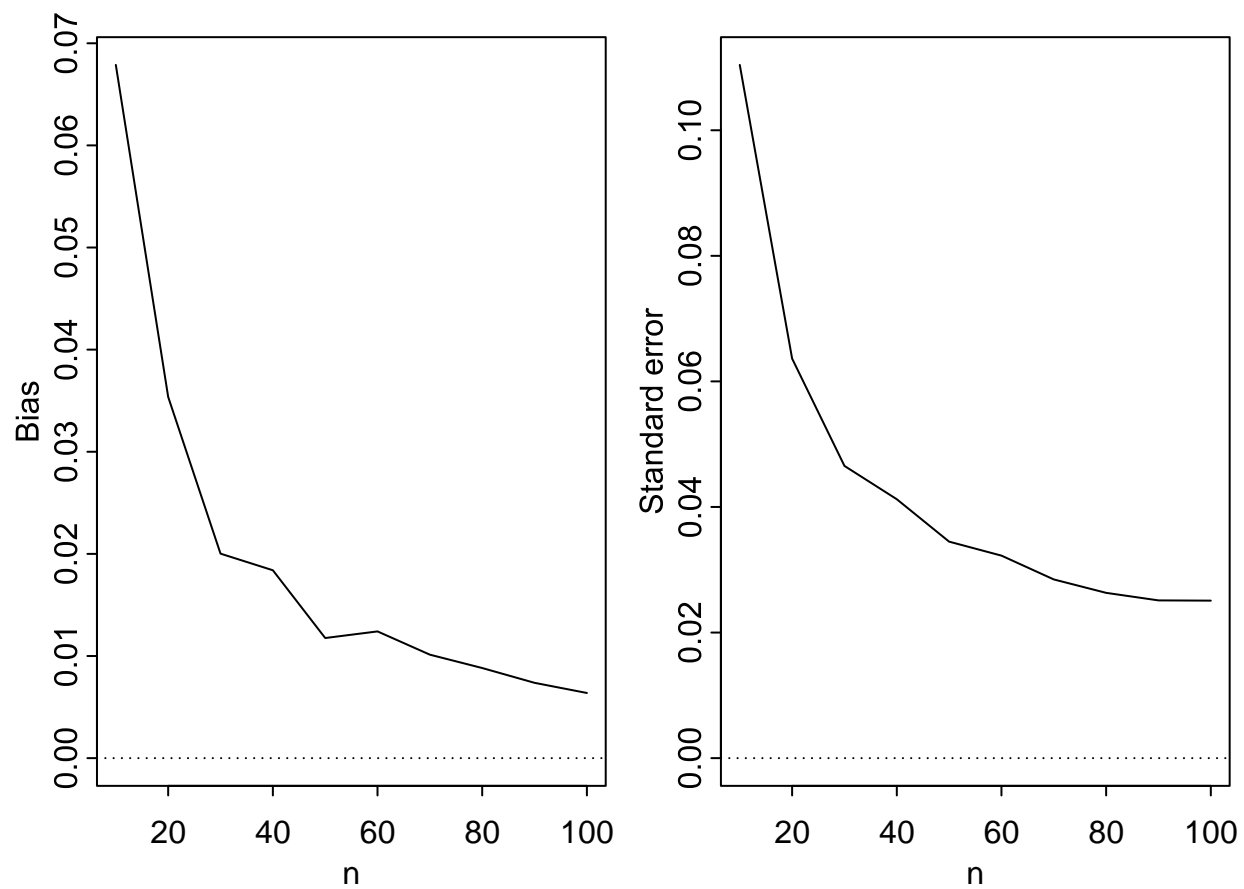
2.a)

$$\text{Bias}(\hat{\varphi}) = E(\hat{\varphi}) - \varphi \text{ and}$$

$$\text{Bias}(\hat{\varphi}_j) = E(\hat{\varphi}_j) - \varphi$$

```
mu = 7
n=seq(10,100,10) # n = 10, 20,..., 100 phi=0.1 #
phi = 0.1 nsim=1000 # monte carlo
approximations ml=numeric(nsim)
bias.ml=numeric(length(n))
se.ml=numeric(length(n)) for(j in 1:length(n)) {
  for(i in 1:nsim) {
    x=rnbinom(n[j], mu * phi / (1 - phi), phi) ml[i]=mean(x)/var(x)
  } bias.ml[j]=mean(ml)-phi se.ml[j]=sd(ml) # standard error
  maximum likelihood
} par(mfrow=c(1,2),mar=c(2.5,2.5,0.5,0.5),mgp=c(1.5,0.5,0),tcl=-0.25)
plot(n,bias.ml,type="l",ylim=range(0,bias.ml),ylab="Bias")

abline(h=0,lty=3) plot(n,se.ml,type="l",ylim=range(0,se.ml),ylab="Standard error")
abline(h=0,lty=3)
```



Is the plot of the Monte Carlo approximations to the bias and standard error of $\hat{\varphi}$ against n


```

phi_hat_i_function = function(x) { phi_hat_of_function =

  mean(x)/var(x) n_of_function = length(x)

  phi_hat_negative_i = numeric(n_of_function) for (i in
1:n_of_function) { phi_hat_negative_i[i] = mean(x[-i])/var(x[-
i])
}
phi_hat_negative_i

  phi_hat_i_of_function = n_of_function*phi_hat_of_function - (n_of_function-1) phi_hat_i_of_function
}

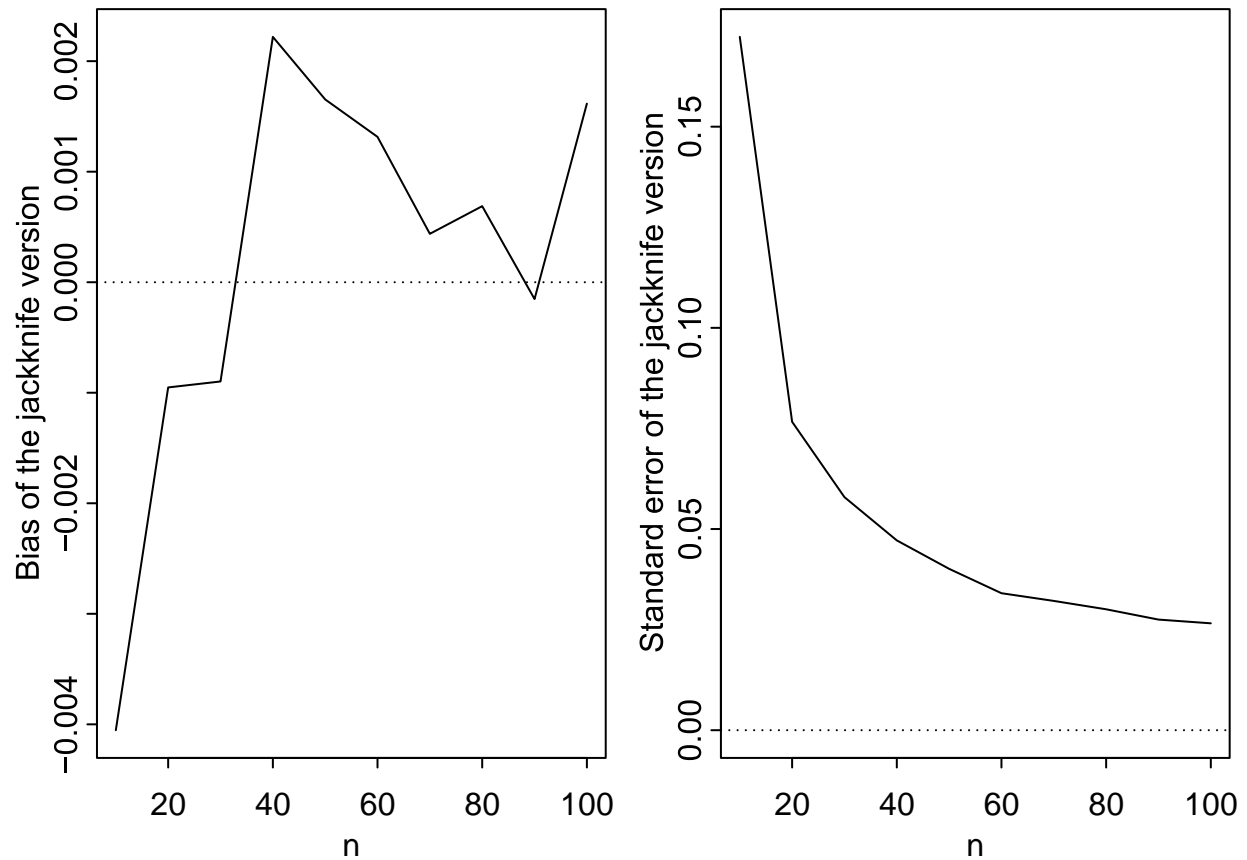
ml_jackknife=numeric(nsim)
bias.ml_jackknife=numeric(length(n))
se.ml_jackknife=numeric(length(n)) for(j in
1:length(n)) {
  for(i in 1:nsim) {
    x=rnbinom(n[j], mu * phi / (1 - phi), phi)
    ml_jackknife[i]=mean(phi_hat_i_function(x))

  } bias.ml_jackknife[j]=mean(ml_jackknife)-phi se.ml_jackknife[j]=sd(ml_jackknife) #
standard error maximum likelihood
} par(mfrow=c(1,2),mar=c(2.5,2.5,0.5,0.5),mgp=c(1.5,0.5,0),tcl=-0.25)
plot(n,bias.ml_jackknife,type="l",ylim=range(0,bias.ml_jackknife),ylab=
abline(h=0,lty=3) plot(n,se.ml_jackknife,type="l",ylim=range(0,se.ml_jackknife),ylab= abline(h=0,lty=3)

```

*phi_hat_negative_i
 "Bias of the jackknife version") "Standard

error of the jackknife vers

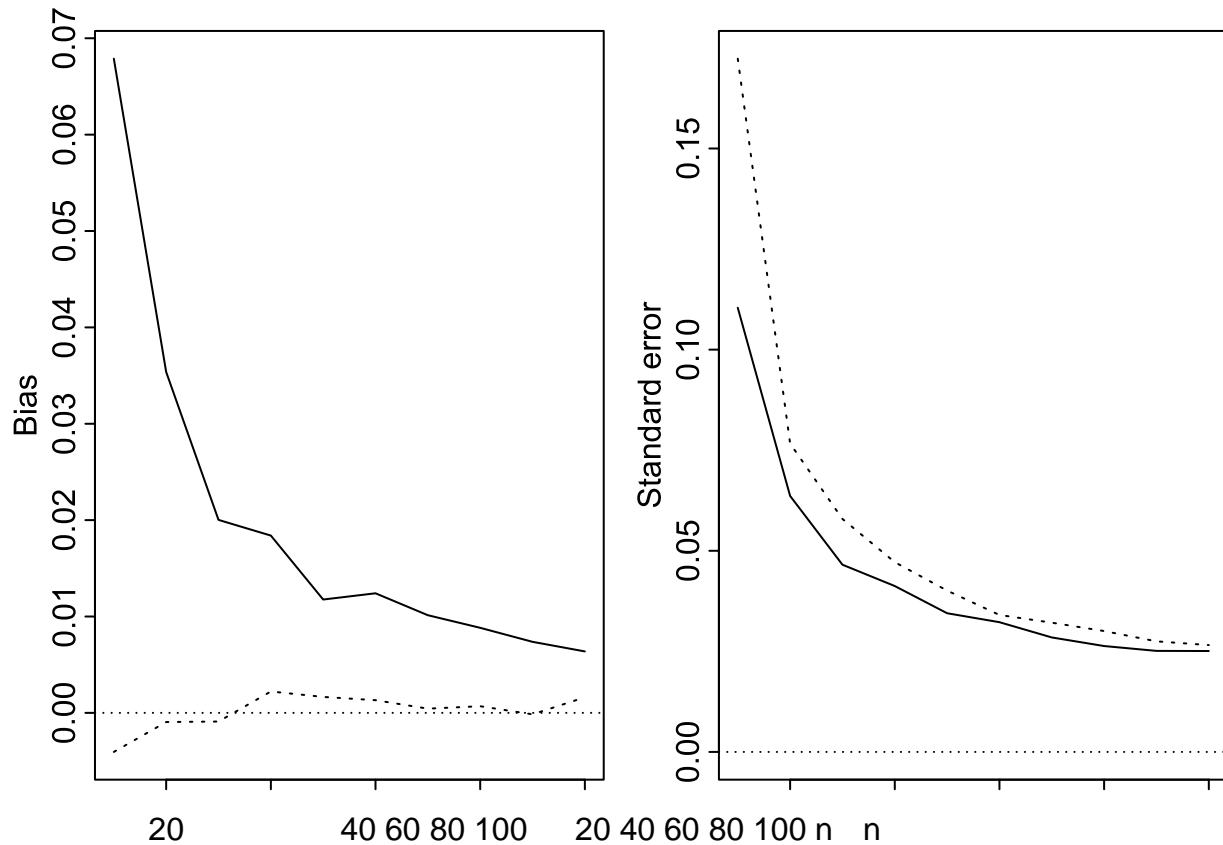


Is the plot of the Monte Carlo approximations to the bias and standard error of the jackknife version of $\hat{\varphi}$ against n

Then putting the two sets of lines on their same respective graphs gives:

```
par(mfrow=c(1,2),mar=c(2.5,2.5,0.5,0.5),mgp=c(1.5,0.5,0),tcl=-0.25)
plot(n,bias.ml,type="l",ylim=range(0,bias.ml, bias.ml_jackknife),ylab="Bias")
lines(n,bias.ml_jackknife,lty=2) abline(h=0,lty=3)

plot(n,se.ml,type="l",ylim=range(0,se.ml, se.ml_jackknife),ylab="Standard error")
lines(n,se.ml_jackknife,lty=2) abline(h=0,lty=3)
```



The bias of the jackknife version of $\hat{\varphi}$ very quickly starts to oscillate around zero by an extremely small number, as n increases but there is not yet enough evidence to say it completely converges to zero. Whereas there is evidence to suggest that the bias of $\hat{\varphi}$ will continue to converge to, as n increases, and if n is increased even further to infinity.

There is evidence that both types of Monte Carlo approximations to the standard errors are asymptotic to a number very close to zero and will only get closer to it as n increases.

Further evidence shows the standard errors are very similar for every n value showing that for an large enough n , they would be very close to the actual standard error.

And the magnitude of the bias of the jackknife version of $\hat{\varphi}$ is smaller than the bias of $\hat{\varphi}$ for all n values.

2.b)

```
phi_hat_i_function = function(x) { phi_hat_of_function =
  mean(x)/var(x) n_of_function = length(x)
  phi_hat_negative_i = numeric(n_of_function) for (i in
  1:n_of_function) { phi_hat_negative_i[i] = mean(x[-i])/var(x[-
  i])
  }
  phi_hat_negative_i phi_hat_i_of_function = n_of_function*phi_hat_of_function -
  (n_of_function-1)*phi_hat_negative_i phi_hat_i_of_function
```

```

} v_hat_J_SquareRoot_function = function(x) {

  phi_hat_i = phi_hat_i_function(x) phi_hat_i

  phi_hat_J = mean(phi_hat_i_function(x))
  phi_hat_J n = length(x)

  phi_hat_i_minus_phi_hat_J_all_squared = numeric(n)
  for (i in 1: n) {
    phi_hat_i_minus_phi_hat_J_all_squared[i] = (phi_hat_i[i] - phi_hat_J)^2
  } phi_hat_i_minus_phi_hat_J_all_squared

  v_hat_J_SquareRoot = sqrt((1/(n*(n-1)))*sum(phi_hat_i_minus_phi_hat_J_all_squared)) v_hat_J_SquareRoot
}

n=seq(10,100,10) # sample size m =
seq(1,10,1)

coverage = numeric(length(n)) phi = 0.1 #
true parameter value alpha = 0.1 # coverage
is 1-2*alpha mu = 7 B = 1000 for (j in m) {

  lo = up = numeric(B) # vectors to store confidence limits for (i in 1:B) { #
    repeat the following B times
      x = rnbino(m*10, mu * phi / (1 - phi), phi) # simulate a sample from Exp(phi) lo[i] =
      mean(phi_hat_i_function(x)) - qt(0.95, df = m*10-1)*v_hat_J_SquareRoot_function(x)
      up[i] = mean(phi_hat_i_function(x)) + qt(0.95, df = m*10-1)*v_hat_J_SquareRoot_function(x)
    } coverage[j] = mean(lo < phi & phi < up) # proportion of times phi is between the lower and upper limit

} coverage

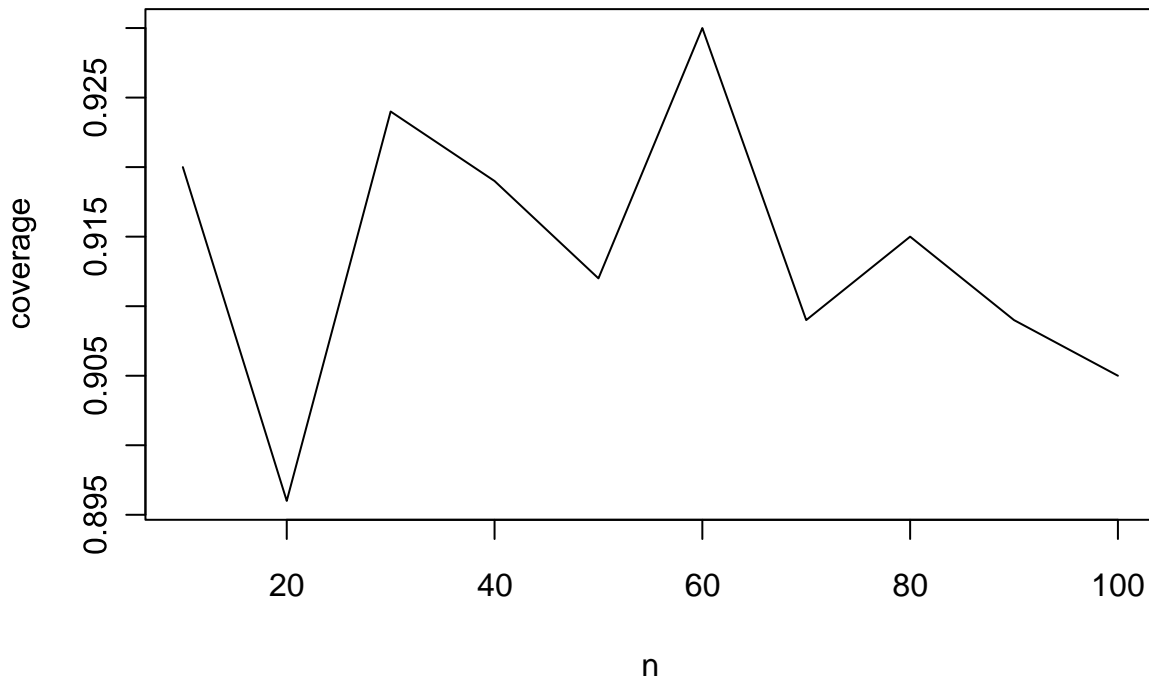
```

calculate and)
calcula

```

## [1] 0.920 0.896 0.924 0.919 0.912 0.930 0.909 0.915 0.909 0.905
plot(n, coverage, type="l")

```



The coverage's of the jackknife confidence interval's do not change significantly as n increases, and there seems to be no correlation between n and the coverage's.

The coverage's are all slightly above 90% which is a very good sign since it's so close to having an coverage percentage equal to the confidence of the equal-tailed 90% confidence interval for the jackknife φ .

Which means the confidence interval has the merits of being largely consistent if n is between 10 and 100, and the calculated confidence interval is extremely representative of the true equal-tailed 90% confidence interval for the jackknife φ , for all those n values.

2.c)

```
phi_hat_i_function = function(x) { phi_hat_of_function =
  mean(x)/var(x) n_of_function = length(x)
  phi_hat_negative_i = numeric(n_of_function) for (i in
  1:n_of_function) { phi_hat_negative_i[i] = mean(x[-i])/var(x[-i])
  }
  phi_hat_negative_i phi_hat_i_of_function = n_of_function*phi_hat_of_function -
  (n_of_function-1)*phi_hat_negative_i phi_hat_i_of_function
```

```

} test_statistic_function = function(x) {

  phi_hat_i = phi_hat_i_function(x) phi_hat_i

  phi_hat_J = mean(phi_hat_i_function(x))
  phi_hat_J n = length(x)

  phi_hat_i_minus_phi_hat_J_all_squared = numeric(n)
  for (i in 1: n) {
    phi_hat_i_minus_phi_hat_J_all_squared[i] = (phi_hat_i[i] - phi_hat_J)^2
  } phi_hat_i_minus_phi_hat_J_all_squared

  v_hat_J_SquareRoot = sqrt((1/(n*(n-1)))*sum(phi_hat_i_minus_phi_hat_J_all_squared)) v_hat_J_SquareRoot

  phi_0 = 1 test_statistic = (phi_hat_J - phi_0)/v_hat_J_SquareRoot
  test_statistic

}

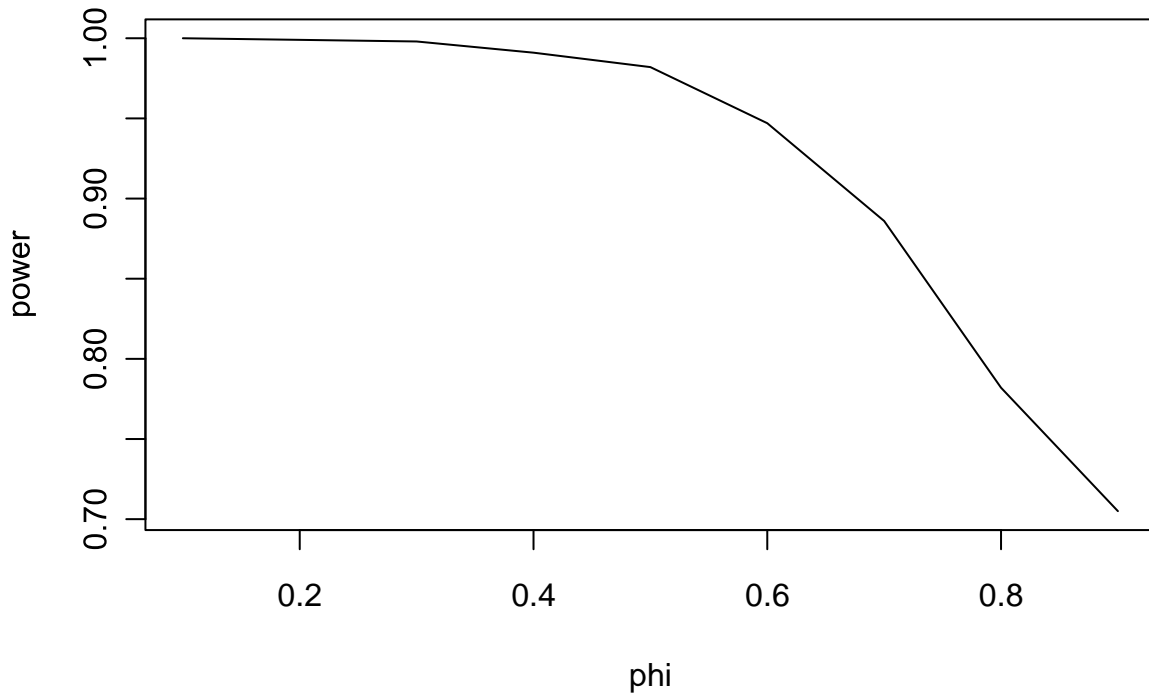
mu = 7 n = 20 # sample
size phi0 = 1

alpha = 0.01 # test size B = 1000 t = numeric(B) # vector to
store test statistics

d = qnbinom(alpha, mu * phi0 / (1 - phi0), phi0) # true value of d phi = seq(0.1, 0.9,
0.1) # range of phi values power = numeric(length(phi)) # vector to store power for (j
in 1:length(phi)) { # for each phi value
  for (i in 1:B) { # repeat the following B times
    x = rnbino(n, mu * phi[j] / (1 - phi[j]), phi[j]) # simulate a sample from nbino(phi) t[i] =
test_statistic_function(x) # calculate and store test statistic
  }
  power[j] = mean(t<d) # Monte Carlo approximation to power
}
power

## [1] 1.000 0.999 0.998 0.991 0.982 0.947 0.886 0.782 0.705 plot(phi, power, type="l") # plot power
function

```



The probability, the data falls in the critical region when H_1 is true is the power of the test, that is; the probability to detect and to reject the a false null hypothesis H_0 . So the higher the power of the test is, the better the test is. The probability of a type two error is the probability of the acceptance of an false null hypothesis H_0 . So the higher the power of the test is, the lower the probability of a type two error.

But the plot of the power against φ shows that as φ increases, the power of the test decreases, meaning the larger φ is, the worse the test becomes. But, even at the largest value of φ , the power of the test it about 70% which is still quite good, and for the lowest value of phi, the power of the test is extremely good.

This plot therefore shows this test is very good because it is extremely powerful, because its power is high for all values of φ .

3.a)

$$\varphi^{\wedge} = h(\mu, \hat{\sigma}^2) = \sigma^{\wedge} - \mu^{\wedge 2}$$

$$\text{So: } h(\mu, \sigma^2) = \varphi$$

$$\text{And: } \text{Bias}(\hat{\varphi}) = E[\hat{\varphi}] - \varphi$$

$$\text{And: } E[\hat{\mu}] = E[\bar{X}] = E\left[\frac{1}{n} \sum_{i=1}^n X_i\right] = \frac{1}{n} E\left[\sum_{i=1}^n X_i\right] = \frac{1}{n} \sum_{i=1}^n E[X_i] = \frac{1}{n} \sum_{i=1}^n \mu = \frac{1}{n} n\mu = \mu$$

$$\text{And: } E\left[\frac{1}{n} \sum_{i=1}^n [(X_i - \bar{X})^2]\right] = \frac{1}{n-1} E\left[\sum_{i=1}^n [(X_i - \bar{X})^2]\right] = \frac{n-1}{n} E\left[\frac{1}{n} \sum_{i=1}^n [(X_i - \bar{X})^2]\right] = \frac{n-1}{n} (1 - [\hat{\sigma}^2]) = E\left[\frac{1}{n-1} \sum_{i=1}^n [(X_i - \bar{X})^2]\right] = \sigma^2$$

$$\text{So: } E[\mu^{\wedge} - E[\mu^{\wedge}]] = E[\mu^{\wedge} - \mu] = E[\mu^{\wedge}] - E[\mu] = E[\mu^{\wedge}] - \mu = \mu - \mu = 0$$

$$\text{And: } E[\sigma^{\wedge 2} - E[\sigma^{\wedge 2}]] = E[\sigma^{\wedge 2} - \sigma^2] = E[\sigma^{\wedge 2}] - E[\sigma^2] = E[\sigma^{\wedge 2}] - \sigma^2 = \sigma^2 - \sigma^2 = 0$$

$$\text{And: } \text{Var}[\mu^{\wedge}] = E[(\mu^{\wedge} - \mu)^2]$$

$$\text{And: } \text{Var}[\sigma^{\wedge 2}] = E[(\sigma^{\wedge 2} - \sigma^2)^2]$$

$$\text{And: } \text{Cov}(\mu^{\wedge}, \sigma^{\wedge 2}) = E[(\mu^{\wedge} - E[\mu^{\wedge}])(\sigma^{\wedge 2} - E[\sigma^{\wedge 2}])] = E[(\mu^{\wedge} - \mu)(\sigma^{\wedge 2} - \sigma^2)]$$

$$\text{So: } \frac{\partial h}{\partial \hat{\mu}}(\mu, \sigma^2) = \frac{\partial}{\partial \hat{\mu}} \left[\frac{\hat{\mu}}{\hat{\sigma}^2} \right](\mu, \sigma^2) = \frac{1}{\hat{\sigma}^2}(\mu, \sigma^2) = \frac{1}{\sigma^2}$$

$$\text{And: } \frac{\partial h}{\partial \hat{\sigma}^2}(\mu, \sigma^2) = \frac{\partial}{\partial \hat{\sigma}^2} \left[\frac{\hat{\mu}}{\hat{\sigma}^2} \right](\mu, \sigma^2) = -\frac{\hat{\mu}}{\sigma^4}$$

$$\text{And: } \frac{\partial^2 h}{\partial \hat{\mu}^2}(\mu, \sigma^2) = \frac{\partial}{\partial \hat{\mu}} \left[\frac{\partial h}{\partial \hat{\mu}}(\mu, \sigma^2) \right] = \frac{\partial}{\partial \hat{\mu}} \left[\frac{1}{\hat{\sigma}^2} \right](\mu, \sigma^2) = 0$$

$$\text{And: } \frac{\partial^2 h}{\partial \hat{\sigma}^4}(\mu, \sigma^2) = \frac{\partial}{\partial \hat{\sigma}^2} \left[\frac{\partial h}{\partial \hat{\sigma}^2}(\mu, \sigma^2) \right] = \frac{\partial}{\partial \hat{\sigma}^2} \left[-\frac{\hat{\mu}}{\hat{\sigma}^4} \right](\mu, \sigma^2) = \frac{2\hat{\mu}}{\sigma^6}$$

$$\text{And: } \frac{\partial^2 h}{\partial \hat{\mu} \partial \hat{\sigma}^2}(\mu, \sigma^2) = \frac{\partial}{\partial \hat{\mu}} \left[\frac{\partial h}{\partial \hat{\sigma}^2}(\mu, \sigma^2) \right] = \frac{\partial}{\partial \hat{\mu}} \left[-\frac{\hat{\mu}}{\hat{\sigma}^4} \right](\mu, \sigma^2) = -\frac{1}{\sigma^4}$$

$$\text{And: } h(\mu, \hat{\sigma}^2) \approx h(\mu, \sigma^2) + (\hat{\mu} - \mu) \frac{\partial h}{\partial \hat{\mu}}(\mu, \sigma^2) + (\hat{\sigma}^2 - \sigma^2) \frac{\partial h}{\partial \hat{\sigma}^2}(\mu, \sigma^2) + \frac{(\hat{\mu} - \mu)^2}{2} \frac{\partial^2 h}{\partial \hat{\mu}^2}(\mu, \sigma^2) + \frac{(\hat{\sigma}^2 - \sigma^2)^2}{2} \frac{\partial^2 h}{\partial \hat{\sigma}^4}(\mu, \sigma^2) + (\hat{\mu} - \mu)(\hat{\sigma}^2 - \sigma^2) \frac{\partial^2 h}{\partial \hat{\mu} \partial \hat{\sigma}^2}(\mu, \sigma^2)$$

$$\text{So: } E[h(\mu, \hat{\sigma}^2)] \approx E[h(\mu, \sigma^2)] + (\hat{\mu} - \mu) \frac{\partial h}{\partial \hat{\mu}}(\mu, \sigma^2) + (\hat{\sigma}^2 - \sigma^2) \frac{\partial h}{\partial \hat{\sigma}^2}(\mu, \sigma^2) + \frac{(\hat{\mu} - \mu)^2}{2} \frac{\partial^2 h}{\partial \hat{\mu}^2}(\mu, \sigma^2) + \frac{(\hat{\sigma}^2 - \sigma^2)^2}{2} \frac{\partial^2 h}{\partial \hat{\sigma}^4}(\mu, \sigma^2) + (\hat{\mu} - \mu)(\hat{\sigma}^2 - \sigma^2) \frac{\partial^2 h}{\partial \hat{\mu} \partial \hat{\sigma}^2}(\mu, \sigma^2)$$

$$= E[h(\mu, \sigma^2)] + E[(\hat{\mu} - \mu) \frac{\partial h}{\partial \hat{\mu}}(\mu, \sigma^2)] + E[(\hat{\sigma}^2 - \sigma^2) \frac{\partial h}{\partial \hat{\sigma}^2}(\mu, \sigma^2)] + E\left[\frac{(\hat{\mu} - \mu)^2}{2} \frac{\partial^2 h}{\partial \hat{\mu}^2}(\mu, \sigma^2)\right] + E\left[\frac{(\hat{\sigma}^2 - \sigma^2)^2}{2} \frac{\partial^2 h}{\partial \hat{\sigma}^4}(\mu, \sigma^2)\right] + E[(\hat{\mu} - \mu)(\hat{\sigma}^2 - \sigma^2) \frac{\partial^2 h}{\partial \hat{\mu} \partial \hat{\sigma}^2}(\mu, \sigma^2)]$$

$$= h(\mu, \sigma^2) + E[(\hat{\mu} - \mu) \frac{\partial h}{\partial \hat{\mu}}(\mu, \sigma^2)] + E[(\hat{\sigma}^2 - \sigma^2) \frac{\partial h}{\partial \hat{\sigma}^2}(\mu, \sigma^2)] + E\left[\frac{(\hat{\mu} - \mu)^2}{2} \frac{\partial^2 h}{\partial \hat{\mu}^2}(\mu, \sigma^2)\right] + E\left[\frac{(\hat{\sigma}^2 - \sigma^2)^2}{2} \frac{\partial^2 h}{\partial \hat{\sigma}^4}(\mu, \sigma^2)\right] + E[(\hat{\mu} - \mu)(\hat{\sigma}^2 - \sigma^2) \frac{\partial^2 h}{\partial \hat{\mu} \partial \hat{\sigma}^2}(\mu, \sigma^2)]$$

$$= h(\mu, \sigma^2) + (0) \frac{\partial h}{\partial \hat{\mu}}(\mu, \sigma^2) + (0) \frac{\partial h}{\partial \hat{\sigma}^2}(\mu, \sigma^2) + E\left[\frac{(\hat{\mu} - \mu)^2}{2}\right](0) + E\left[\frac{(\hat{\sigma}^2 - \sigma^2)^2}{2}\right] \frac{\partial^2 h}{\partial \hat{\sigma}^4}(\mu, \sigma^2) + E[(\hat{\mu} - \mu)(\hat{\sigma}^2 - \sigma^2) \frac{\partial^2 h}{\partial \hat{\mu} \partial \hat{\sigma}^2}(\mu, \sigma^2)]$$

$$= h(\mu, \sigma^2) + E\left[\frac{(\hat{\sigma}^2 - \sigma^2)^2}{2}\right] \frac{\partial^2 h}{\partial \hat{\sigma}^4}(\mu, \sigma^2) + E[(\hat{\mu} - \mu)(\hat{\sigma}^2 - \sigma^2) \frac{\partial^2 h}{\partial \hat{\mu} \partial \hat{\sigma}^2}(\mu, \sigma^2)]$$

$$= h(\mu, \sigma^2) + \frac{1}{2} E[(\hat{\sigma}^2 - \sigma^2)^2] \frac{\partial^2 h}{\partial \hat{\sigma}^4}(\mu, \sigma^2) + E[(\hat{\mu} - \mu)(\hat{\sigma}^2 - \sigma^2) \frac{\partial^2 h}{\partial \hat{\mu} \partial \hat{\sigma}^2}(\mu, \sigma^2)]$$

$$= h(\mu, \sigma^2) + \frac{1}{2} \text{Var}[\hat{\sigma}^2] \frac{\partial^2 h}{\partial \hat{\sigma}^4}(\mu, \sigma^2) + \text{Cov}(\hat{\mu}, \hat{\sigma}^2) \frac{\partial^2 h}{\partial \hat{\mu} \partial \hat{\sigma}^2}(\mu, \sigma^2)$$

$$= \phi + \frac{1}{2} \text{Var}[\hat{\sigma}^2] 2\hat{\mu} \frac{1}{\sigma^6} + \text{Cov}(\hat{\mu}, \hat{\sigma}^2) \left(-\frac{1}{\sigma^4}\right)$$

$$= \phi + \hat{\mu} \frac{1}{\sigma^6} \text{Var}[\hat{\sigma}^2] - \frac{1}{\sigma^4} \text{Cov}(\hat{\mu}, \hat{\sigma}^2)$$

$$\text{So: } \text{Bias}(\hat{\phi}) = E[\hat{\phi}] - \phi \approx \phi + \hat{\mu} \frac{1}{\sigma^6} \text{Var}[\hat{\sigma}^2] - \frac{1}{\sigma^4} \text{Cov}(\hat{\mu}, \hat{\sigma}^2) - \phi = \hat{\mu} \frac{1}{\sigma^6} \text{Var}[\hat{\sigma}^2] - \frac{1}{\sigma^4} \text{Cov}(\hat{\mu}, \hat{\sigma}^2)$$

$$\text{So: } \text{Bias}(\hat{\phi}) \approx \hat{\mu} \frac{1}{\sigma^6} \text{Var}[\hat{\sigma}^2] - \frac{1}{\sigma^4} \text{Cov}(\hat{\mu}, \hat{\sigma}^2)$$

3.b)

$$m_3 = n \frac{1}{n} \sum_{i=1}^n [(X_i - \bar{X})^3]$$

$$\text{And: } m_4 = n \frac{1}{n} \sum_{i=1}^n [(X_i - \bar{X})^4]$$

$$\text{And: } \text{Cov}(\hat{\mu}, \hat{\sigma}^2) = \frac{m_3}{n^3}$$

$$\text{And: } \text{Var}[\hat{\sigma}^2] = \frac{1}{n} [(m_4 - (n-3)\hat{\sigma}^4)/(n-1)] = \frac{1}{n} [(m_4 - (n-3)(\hat{\sigma}^2)^2)/(n-1)]$$

$$\text{And: } \hat{\phi} = \sigma^{-1} \hat{\mu}^2$$

$$\text{And: } \text{Bias}(\hat{\phi}) \approx \hat{\mu} \frac{1}{\sigma^6} \text{Var}[\hat{\sigma}^2] - \frac{1}{\sigma^4} \text{Cov}(\hat{\mu}, \hat{\sigma}^2)$$

So bias-corrected estimator for φ is: $\varphi_e = \hat{\varphi} - \text{Bias}(\hat{\varphi}) = \hat{\varphi} - \mu^{\hat{\sigma}^2} \frac{1}{\sigma^2} \text{Var}[\sigma^2] + \frac{1}{\sigma^4} \text{Cov}(\mu, \hat{\sigma}^2)$
 $= \mu^{\hat{\sigma}^2} \frac{1}{\sigma^2} - \mu^{\hat{\sigma}^2} \frac{1}{(\sigma^2)^2} \text{Var}[\sigma^2] + \frac{1}{(\sigma^2)^2} \text{Cov}(\mu, \hat{\sigma}^2)$

```
X_data = c(1, 9, 5, 2, 1, 0, 1, 24, 3, 0, 4, 12, 2, 1, 1, 14, 7, 2, 4, 0)
```

```
mu_hat = mean(X_data) sigma_hat_squared = var(X_data) X_bar = mu_hat n = length(X_data) m_3 =
```

```
(1/n)*sum((X_data - X_bar)^3) m_4 = (1/n)*sum((X_data - X_bar)^4)
```

```
covariance_of_mu_hat_and_sigma_hat_squared = m_3/n variance_of_sigma_hat_squared =
```

```
(1/n)*((m_4 - (n-3)*sigma_hat_squared^2)/(n-1))
```

```
bias_corrected_estimator = mu_hat/sigma_hat_squared -
```

```
(1/(sigma_hat_squared^3))*variance_of_sigma_hat_squared -
```

```
(1/(sigma_hat_squared^2))*covariance_of_mu_hat_and_sigma_hat_squared bias_corrected_estimator
```

```
## [1] 0.112863
```

Is the bias-corrected estimator for the data.