STAT3023 Statiscal Inference

Lab Week 8

Tutor: Wen Dai

SID: 470408326

School of Mathematics and Statistics

The University of Sydney

Semester 2, 2021

Two-sided tests for normal variance

Suppose $X_1, ..., X_n$ are iid $N(\mu, \sigma^2)$. In week 7 tutorial it was noted that the statistic $Y = (n-1) S^2 = \sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2$ (where $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ and S^2 are the sample mean and sample variance) has a $\sigma^2 \chi^2_{n-1}$ distribution (note we are **not** multiplying by $\frac{1}{2}$ as we did in the week 7 Tutorial!). Consider testing

$$H_0 : \sigma^2 = 1 \qquad \text{against} \qquad H_1 : \sigma^2 \neq 1 \tag{1}$$

1. One possible level-α test is the "equal-tailed" test based on Y , where we reject for $Y < a$ or $Y > b$ where

$$P_0 \{Y < a\} = P_0 \{Y > b\} = \frac{\alpha}{2}$$

(a) Taking $\alpha = 0.04$ and $n = 5$, find appropriate values `a` and `b`.

Solution

since $Y \sim \sigma^2 \chi^2_{n-1}$ under the null hypothesis and taking $n = 5$ we have that $Y \sim \chi^2_4$. With a significance level of $\alpha = 0.04$, to find the values of a and b note the following code:

```
alpha = 0.04
n = 5
# P0(Y < a) = alpha/2 F_Y(a) = P0(Y ≤ a) = alpha/2 [as Y is continuous]
a = qchisq(p = alpha/2, df = n - 1, lower.tail = TRUE)
# P0(Y > b) = alpha/2 1 - P0(Y ≤ b) = alpha/2 P0(Y ≤ b) = 1 - alpha/2
b = qchisq(p = 1 - alpha/2, df = n - 1, lower.tail = TRUE)
# print out the values of a and b
c(a, b)
```

```
[1]  0.4293982 11.6678434
```

(b) Defining `sig.sq=(50:150)/100` plot the power of the test against `sig.sq`. Add a horizontal dotted line to indicate the level.

Solution

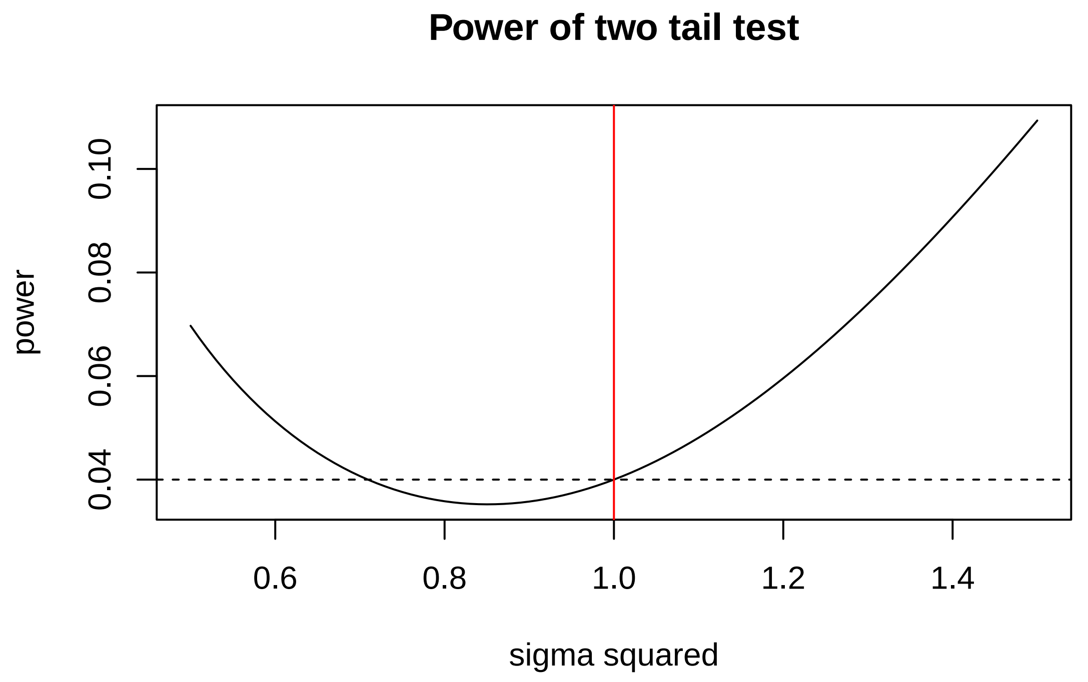
We plot the power of the test for various values of σ^2. To do this, recall that the probability of rejecting is $P(Y < a) + P(Y > b)$ for a and b. Since $Y \sim \sigma^2\chi^2_{n-1} \iff \frac{Y}{\sigma^2} \sim \chi^2_{n-1}$ we have that:

$$\begin{aligned} P(Y < a) &= P\left(\frac{Y}{\sigma^2} < \frac{a}{\sigma^2}\right) \\ &= P\left(\chi^2_4 < \frac{a}{\sigma^2}\right) \end{aligned}$$

And that

$$\begin{aligned} P(Y > b) &= P\left(\frac{Y}{\sigma^2} > \frac{b}{\sigma^2}\right) \\ &= P\left(\chi^2_4 > \frac{b}{\sigma^2}\right) \end{aligned}$$

```r
# sigma values
sig.sq = (50:150)/100
# power of rejecting: P(chi^2 < a/sigma^2) + P(chi^2 > b/sigma^2)
power.rej = pchisq(q = a/sig.sq, df = n - 1, lower.tail = TRUE) + pchisq(q = b
    /sig.sq, df = n - 1, lower.tail = FALSE)
# plot power against theta
plot(sig.sq, power.rej, type = "l", main = "Power of two tail test", xlab = "
    sigma squared", ylab = "power")
# add a horizontal dotted line at 0.04
abline(h = 0.04, lty = 2)
# add sigma_0^2 too
abline(v = 1, col = "red")
```



Note that the red veritcal line represents the value of $\sigma^2 = \sigma_0^2 = 1$

2. In Tutorial week 7 we also saw that the UMPU test rejects for large values of

$$S^2 - \log\left(S^2\right)$$

which is equivalent to rejecting for small values of the statistic

$$T = (n-1)\log Y - Y;$$

to see this, write $\log\left(S^2\right) = \log Y - \log(n-1)$, multiply through by $n-1$ and ignore the $(n-1)\log(n-1)$ term.

If the test is to have level α, we reject for $Y \le c$ or $Y \ge d$ where:

$$P_0(Y \le c) + P_0(Y \ge d) = \alpha \tag{2}$$

And

$$(n-1)\log(c) - c = (n-1)\log(d) - d \tag{3}$$

(a) Write a function of the form

```
fn=function(c,alpha,n) {
  ...
}
```

which

- computes the appropiate `d` so that `c` and `d` satisfy (2)
- then computes and outputs the difference between the left-hand side and right-hand side in (3).

Solution

For this question, what you're essentially doing is solving for d in terms of c from equation (2) and then forming an equation **in terms of** c for equation (3) via substitution. This sets us up to **solve the equation** in part (b)

```
fn = function(c, alpha, n) {
    # find the value of d which satisfies the relationship of equation (2).
      Note that d is a
    # function of c when written like this!  P0(Y ≤ c) + P0(Y ≥ d) = alpha
      P0(Y ≥ d) = alpha -
    # P0(Y ≤ c) P0(Y < d) = 1 - alpha + P0(Y ≤ c) P0(Y ≤ d) = 1 - alpha +
      P0(Y ≤ c) [since Y is
    # continuous]
    d = qchisq(1 - alpha + pchisq(c, df = n - 1), df = n - 1)

    # sub it into equation (3) and return it as our equation
    equation = (n - 1) * log(c) - c - (n - 1) * log(d) + d
    return(equation)
}
```

(b) Use the R function `uniroot()` to find the root (in `c`) of the equation `fn(c,0.04,5)=0`. In your code you will need a command along the lines of

```
uniroot(fn,lower=0,upper=...,alpha=0.04,n=5)
```

Consult the week 7 exercise for some hints as to how to choose the `upper=...`. When you have worked out the right commands, wrap it all in a function of the form

```
norm.var.umpu=function(alpha,n) {
  ...
}
```

which returns a list containing elements `$c` and `$d`.

Solution

We note that c can be no bigger than the lower α-quantile of the χ^2_{n-1} distribution. hence the `upper` parameter will be the lower α-quantile "minus a little bit of wiggle room" represented by the variable `eps`

```
eps = 1e-08
alpha = 0.04
n = 5
uniroot(f = fn, lower = 0, upper = qchisq(alpha, df = n - 1) - eps, alpha =
    alpha, n = n)
```

```
$root
[1] 0.5395026

$f.root
[1] 1.269793e-07

$iter
[1] 7

$init.it
[1] NA

$estim.prec
[1] 9.838951e-05
```

We now make a function solving for c and d by wrapping all the functions we have made in the following function. Note that up until and including this point, the main idea of this function is very simple– all we did was solve for d from equation (2), then sub that into equatin (3) to get an equation in c. We then solved that and finally we re-substitute the value of c into equation (2) to solve for d and output the answer in a list.

```
norm.var.umpu = function(alpha, n) {
    # wiggle room
    eps = 1e-08
    # calculate c
    c = uniroot(f = fn, lower = 0, upper = qchisq(alpha, df = n - 1) - eps,
      alpha = alpha, n = n)$root
    # calculate d. Recall: P0(Y ≤ d) = 1 - alpha + P0(Y ≤ c) [since Y is
      continuous]
    d = qchisq(1 - alpha + pchisq(c, df = n - 1), df = n - 1)
    return(list(c = c, d = d))
}
norm.var.umpu(0.04, 5)
```

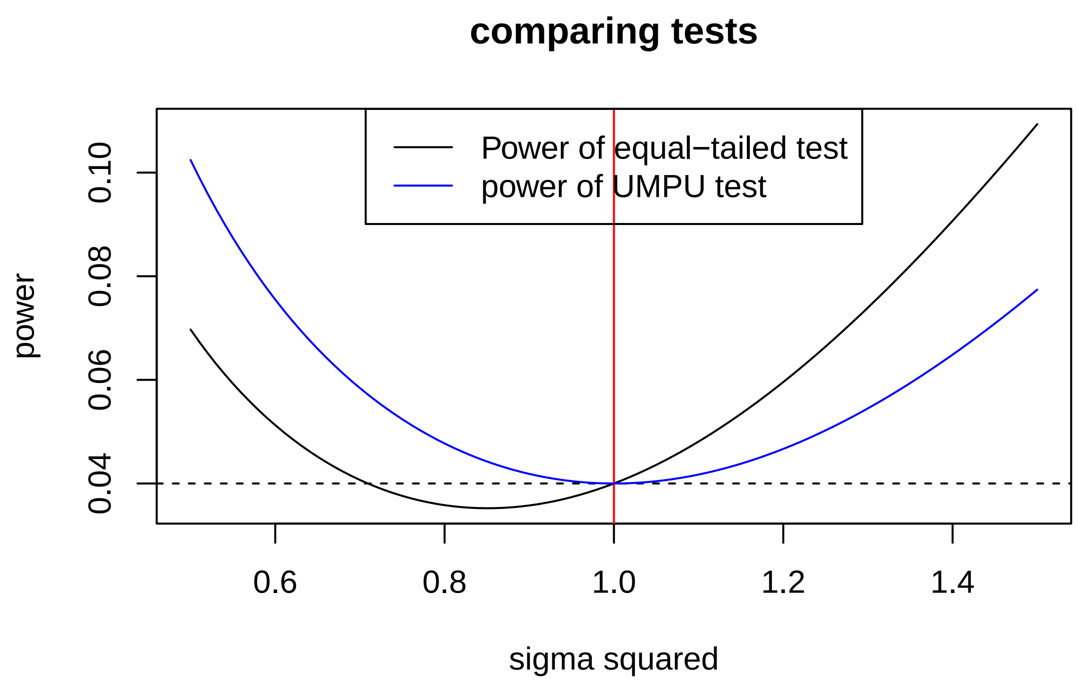
```
$c
[1] 0.5395026

$d
[1] 13.38425
```

(c) Recreate your plot from part (b) of the previous question and add to it the power function of the UMPU test.

Solution

```
# significane level
alpha = 0.04
# degrees of freedom
n = 5
# sigma squared values
sig.sq = (50:150)/100

# for the two tailed test (use a and b from before):
power.rej = pchisq(q = a/sig.sq, df = n - 1, lower.tail = TRUE) + pchisq(q = b
    /sig.sq, df = n - 1, lower.tail = FALSE)
```

```
plot(sig.sq, power.rej, type = "l", main = "comparing tests", xlab = "sigma squared", ylab = "power")
abline(h = 0.04, lty = 2)
abline(v = 1, col = "red")

# for the UMPU test: 1. find the values of c and d according to alpha = 0.04 and n = 5
umpu = norm.var.umpu(alpha, n)
# 2. find the power of the umpu test for each theta value:
umpu.power = pchisq(umpu$c/sig.sq, df = n - 1) + pchisq(umpu$d/sig.sq, df = n - 1, lower.tail = FALSE)
# 3. plot
lines(sig.sq, umpu.power, col = "blue")
# legend
legend("top", legend = c("Power of equal-tailed test", "power of UMPU test"),
    col = c("black", "blue"),
     lty = c(1, 1))
```



3. The GLRT test of (1) above uses the statistic

$$L_n = \ell\left(\overline{X}, \frac{Y}{n}; \boldsymbol{X}\right) - \ell\left(\overline{X}, 1; \boldsymbol{X}\right) = -\frac{n}{2}\log\left(\frac{Y}{n}\right) - \frac{n}{2} + \frac{Y}{2}$$

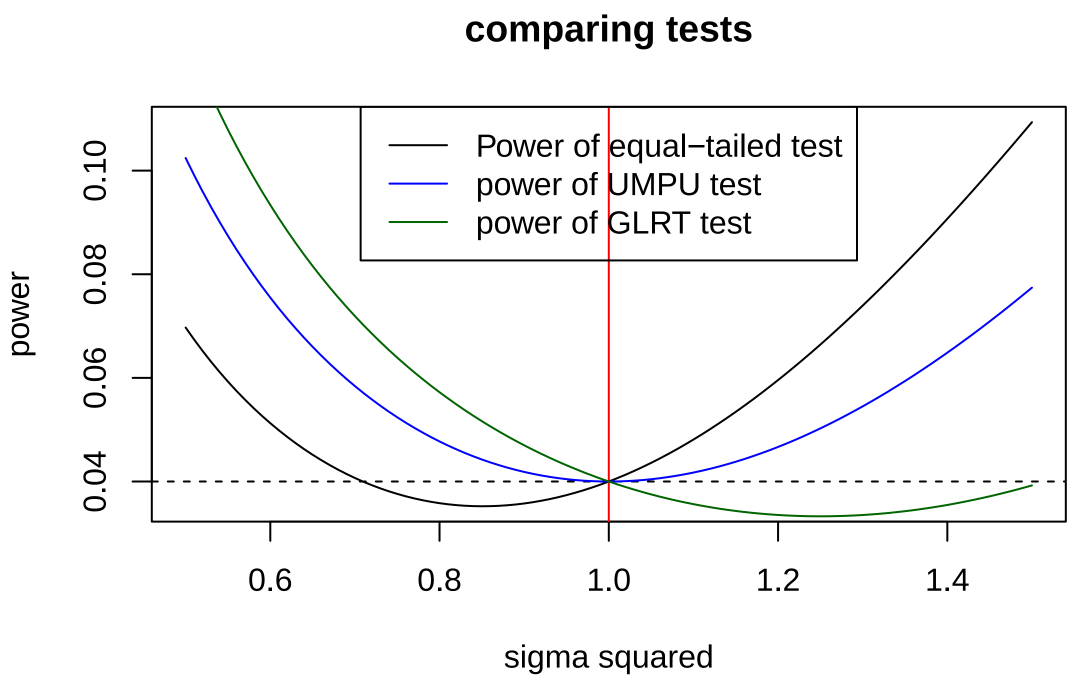
which is an increasing function of $Y - n\log Y$ (as opposed to the UMPU which rejects for large $Y - n\log Y$). Adapt your code for the previous question to compute the power of the exact GLRT, recreate your earlier plot and add a power curve to it so it shows all 3 power curves on the 1 graph. Add an informative heading, legend, etc.. Comment on the main differences between the 3 tests.

Solution

Quite simply we just repeat the steps for the new equation, analgous to equation (3):

$$-\frac{n}{2}\log\left(\frac{c}{2}\right)-\frac{n}{2}+\frac{c}{2}=-\frac{n}{2}\log\left(\frac{d}{2}\right)-\frac{n}{2}+\frac{d}{2} \tag{4}$$

```r
# function for generalized linear ratio test (equation in c to be solved)
fn.glrt = function(c, alpha, n) {
    d = qchisq(1 - alpha + pchisq(c, df = n - 1), df = n - 1)
    equation = -(n/2) * log(c/n) + c/2 + (n/2) * log(d/n) - d/2
    return(equation)
}
# calculate c and d
norm.var.glrt = function(alpha, n) {
    # wiggle room
    eps = 1e-08
    # calculate c
    c = uniroot(f = fn.glrt, lower = 0, upper = qchisq(alpha, df = n - 1) - eps,
        alpha = alpha, n = n)$root
    # calculate d. Recall: P0(Y ≤ d) = 1 - alpha + P0(Y ≤ c) [since Y is
        continuous]
    d = qchisq(1 - alpha + pchisq(c, df = n - 1), df = n - 1)
    return(list(c = c, d = d))
}

# plot everything
alpha = 0.04
n = 5
sig.sq = (50:150)/100

# for the two tailed test (use a and b from before):
power.rej = pchisq(q = a/sig.sq, df = n - 1, lower.tail = TRUE) + pchisq(q = b/
    sig.sq, df = n - 1, lower.tail = FALSE)
plot(sig.sq, power.rej, type = "l", main = "comparing tests", xlab = "sigma
    squared", ylab = "power")
abline(h = 0.04, lty = 2)
abline(v = 1, col = "red")

# for the UMPU test:
umpu = norm.var.umpu(alpha, n)
umpu.power = pchisq(umpu$c/sig.sq, df = n - 1) + pchisq(umpu$d/sig.sq, df = n - 1,
     lower.tail = FALSE)
lines(sig.sq, umpu.power, col = "blue")

# for the glrt:
glrt = norm.var.glrt(alpha, n)
glrt.power = pchisq(glrt$c/sig.sq, df = n - 1) + pchisq(glrt$d/sig.sq, df = n - 1,
     lower.tail = FALSE)
lines(sig.sq, glrt.power, col = "Darkgreen")

# legend
legend("top", legend = c("Power of equal-tailed test", "power of UMPU test", "
    power of GLRT test"), col = c("black",
     "blue", "Darkgreen"), lty = c(1, 1, 1))
```



And so what we see is that:[1]

- For values greater than σ_0 the power of the equal tailed test **exceeds** the other two tests. For values less than σ_0 the power of the equal tailed test is lower than the other two tests
- The UMPU test is the only unbiased test in the sense that there **are not** any values σ for which the power is lower than that of σ_0. The further you get away from the null, the higher the probability of rejecting (which is good!)
- For values greater than σ_0 the power of the GLRT is **lower** the other two tests. For values less than σ_0 the power of the GLRT **exceeds** the other two tests.

It seems to me that if you were to do a single sided test, either pick the equal tailed test or the GLRT depending on the alternative (so the power is greater) whereas with a two sided test, stick with the UMPU test.

4. As a final step, recreate your last plot but use an extended range for the parameter:
`sig.sq=(1:400)/100`.

Solution

```
# plot everything
alpha = 0.04
n = 5
sig.sq = (1:400)/100

# for the two tailed test (use a and b from before):
power.rej = pchisq(q = a/sig.sq, df = n - 1, lower.tail = TRUE) + pchisq(q = b/
    sig.sq, df = n - 1, lower.tail = FALSE)
```

[1] These comments take into account the graph from Q4 too

```
plot(sig.sq, power.rej, type = "l", main = "comparing tests", xlab = "sigma
    squared", ylab = "power")
abline(h = 0.04, lty = 2)
abline(v = 1, col = "red")

# for the UMPU test:
umpu = norm.var.umpu(alpha, n)
umpu.power = pchisq(umpu$c/sig.sq, df = n - 1) + pchisq(umpu$d/sig.sq, df = n - 1,
     lower.tail = FALSE)
lines(sig.sq, umpu.power, col = "blue")

# for the glrt:
glrt = norm.var.glrt(alpha, n)
glrt.power = pchisq(glrt$c/sig.sq, df = n - 1) + pchisq(glrt$d/sig.sq, df = n - 1,
     lower.tail = FALSE)
lines(sig.sq, glrt.power, col = "Darkgreen")

# legend
legend("top", legend = c("Power of equal-tailed test", "power of UMPU test", "
    power of GLRT test"), col = c("black",
     "blue", "Darkgreen"), lty = c(1, 1, 1))
```