Ratio of means

Suppose that $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$ are from Normal distributions $N(\mu_X, 1)$ and $N(\mu_Y, 1)$ and we are interested in $\mu_X/\mu_Y$. An example might be cost-effectiveness calculations where $X$ represents the cost of a treatment and $Y$ the benefit.

An obvious estimator of $r = \mu_X/\mu_Y$ is $R = \bar{X}/\bar{Y}$, but we don’t have a simple formula for the sampling distribution.

There is a clever trick where we write the hypothesis $r = r_0$ as $\mu_X - r_0\mu_Y = 0$.

\[
\begin{align*}
\bar{X} & \sim N(\mu_X, 1/n) \\
\bar{Y} & \sim N(\mu_Y, 1/n) \\
\bar{X} - r_0\bar{Y} & \sim N\left(\mu_X - r_0\mu_Y, \frac{1 + r_0^2}{n}\right)
\end{align*}
\]
We can now test $r = r_0$ by comparing $\bar{X} - r_0\bar{Y}$ to $N\left(0, \frac{1+r_0^2}{n}\right)$: we reject the hypothesis if

$$\frac{\bar{X} - r_0\bar{Y}}{\sqrt{\frac{1+r_0^2}{n}}}$$

is large (compared to its standard Normal sampling distribution). For a 5% level test, ‘large’ means larger than 1.96.

We can solve

$$\frac{\bar{X} - r_0\bar{Y}}{\sqrt{\frac{1+r_0^2}{n}}} = \pm 1.96$$

since it is just a quadratic equation.
The solutions to $ax^2 + bx + c = 0$ are

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

`quad.solve <- function(a,b,c){
  discr <- b*b-4*a*c
  if (discr<0)
    return(NULL)
  if (discr==0)
    return(-b/(2*a))
  (-b+c(-1,1)*sqrt(discr))/(2*a)
}`
One weird possibility is obvious: what if $b^2 < 4ac$ so there are no (real) solutions?

Another one is less obvious: do we know that the hypothesis is rejected outside the interval, or might it be rejected inside the interval?

We can pick a point in the middle of the interval and check whether the test rejects or not.
fieller.interval <- function(xbar, ybar, n, alpha=0.05){
    one.96 <- abs(qnorm(alpha/2))
    endpoints <- quadsolve(ybar^2-one.96^2/n,
                            -2*xbar*ybar,
                            xbar^2-one.96^2/n)
    if (length(endpoints)<2)
        return(c(-Inf, Inf, NA, NA))
    midpoint<-mean(endpoints)
    if (abs(xbar-midpoint*ybar) < sqrt((1+midpoint^2)/n))
        c(endpoints, NA, NA)
    else
        c(-Inf, sort(endpoints), Inf)
}
one.interval <- function(n,mu.x=1,mu.y=1){
  x<-rnorm(n, mean=mu.x)
  y<-rnorm(n, mean=mu.y)
  fieller.interval(mean(x), mean(y), n)
}
lots <- replicate(100, one.interval(20))
plot(1, type="n",xlim=c(1,100), ylim=c(-10, 10),
     xlab="",ylab="Ratio")
lots[lots == Inf] <- 20
lots[lots == -Inf] <- -20

segments(1:100,lots[,1],1:100,lots[,2])
segments(1:100,lots[,3],1:100,lots[,4])
segments(1:100,lots[,2],1:100,lots[,3],col="grey80")
abline(h=0.5,col="red")
We could get complex solutions to our quadratic equation: we just need to tell R that the discriminant should be treated as a complex number.

cquadsolve <- function(a,b,c){
  discr <- as.complex(b*b-4*a*c)
  (-b+c(-1,1)*sqrt(discr))/(2*a)
}

giving

> cquadsolve(1,0,1)
[1] 0-1i 0+1i
> quadsolve(1,0,1)
NULL

Complex numbers aren’t very useful in statistics, except for Fourier analysis of time series.
The first line of the file has an extra line break, so it’s easiest to download and edit it rather than reading it directly

```r
inflamm<-read.table("inflamm.txt")
```

a. Provide suitable statistics for the distribution of times to censoring for observations of death. In particular, consider whether you can estimate the minimum time of follow-up for these patients

The ”survival” package has built-in censored data functions

```r
library(survival)
> survfit(Surv(ttodth,1-death),data=inflamm)
```
Data analysis

Call: survfit(formula = Surv(ttodth, 1 - death), data = inflamm)

    n  events median  0.95LCL  0.95UCL
5000   3879   2733     2726     2747

> plot(survfit(Surv(ttodth,1-death),data=inflamm))
> with(inflamm, by(ttodth,death,summary))

INDICES: 0

    Min. 1st Qu.  Median     Mean 3rd Qu.     Max.
    1480   2630   2726     2604   2834     2942

--------------------------------------------------------

INDICES: 1

    Min. 1st Qu.  Median     Mean 3rd Qu.     Max.
       5    934    1609     1554   2236     2912

> 1480/365

[1]  4.054795
Data analysis

Histogram of ttodth[death == 0]/365

Frequency

ttodth[death == 0]/365
Data analysis
Data analysis

Of course, we have survival analysis code that we wrote earlier

```r
source("medsurv.R")
km<-with(subset(inflamm, crp<=2), kaplanmeier(ttodth,death))
plot(km$time,km$surv,ylim=c(0,1), type="s")
km<-with(subset(inflamm, crp>2), kaplanmeier(ttodth,death))
lines(km$time,km$surv,ylim=c(0,1), type="s",lty=2)
legend("bottomleft",lty=1:2, legend=c("CRP<=2","CRP>2"))
```
Data analysis
2. We are interested in estimating the probability of a patient dying from any cause in the years following accrual to the study.
   a. Provide suitable descriptive statistics for the distribution of times to death from any cause for all patients in the study.
   
   b. Produce a plot of survival curves stratified by the groups defined by whether the C-reactive protein (CRP) value was higher than 2 mg/l or not. Produce a table of estimates of the 90th, 80th, and 75th percentiles of the survival distribution by CRP strata. Also include in that table the estimated probabilities of surviving for 3, 5, and 8 years for each stratum. Are the estimates suggestive that CRP level is associated with mortality? Give descriptive statistics supporting your answer.
   
   c. Repeat part b using thresholds of 3 mg/l and 5 mg/l for CRP.
Data analysis

> with(subset(inflamm, crp<=2), mediansurv(ttodth,death,quantile=0.9))
[1] 1750
> with(subset(inflamm, crp<=2), mediansurv(ttodth,death,quantile=0.9))
[1] 1750
> with(subset(inflamm, crp<=2), mediansurv(ttodth,death,quantile=0.75))
[1] NA
Warning messages:
1: no non-missing arguments to min; returning Inf
2: no non-missing arguments to min; returning Inf
3: NAs introduced by coercion
> with(subset(inflamm, crp>2), mediansurv(ttodth,death,quantile=0.9))
[1] 1067
> with(subset(inflamm, crp>2), mediansurv(ttodth,death,quantile=0.9))
[1] 1067
> with(subset(inflamm, crp>2), mediansurv(ttodth,death,quantile=0.75))
[1] 2430
> survat<-function(time,event, at.time){
+ km<-kaplanmeier(time,event)
+ km$surv[min(which(km$time>at.time))]
+ }
> with(subset(inflamm, crp<=2), survat(ttodth,death, at.time=365*3))
  1100
0.952895
Data analysis

> with(subset(inflamm, crp<=2), survat(ttodth,death, at.time=365*5))
  1828
0.8909532
> with(subset(inflamm, crp<=2), survat(ttodth,death, at.time=365*8))
  2922
0.776836
> with(subset(inflamm, crp>2), survat(ttodth,death, at.time=365*3))
  1097
0.8971215
> with(subset(inflamm, crp>2), survat(ttodth,death, at.time=365*5))
  1826
0.8190185
> with(subset(inflamm, crp>2), survat(ttodth,death, at.time=365*8))
  2922
0.678866
Data analysis

Now he wants us to do this all over again. We might cut and paste all that code twice, or put it in a function

```r
analyse.crpstratum<-function( threshold){
kmb<-with(subset(inflamm, crp<=threshold), kaplanmeier(ttodth,death))
plot(km$time,km$surv,ylim=c(0,1), type="s")
kml<-with(subset(inflamm, crp>threshold), kaplanmeier(ttodth,death))
lines(km$time,km$surv,ylim=c(0,1), type="s",lty=2)
legend("bottomleft",lty=1:2,
       legend=(paste(c("CRP<=","CRP>"),threshold)))
below<-subset(inflamm, crp<=threshold)
qbelow <- with(below, sapply(c(0.9,0.8,0.75), mediansurv,
     time=ttodth, event=death))
pbelow <-with(below, sapply(c(3,5,8)*365,survat,
     time=ttodth, event=death))
above<-subset(inflamm, crp>threshold)
```
Data analysis

```r
qabove <- with(above, sapply(c(0.9, 0.8, 0.75),
    mediansurv, time=ttodth, event=death))
pabove <- with(above, sapply(c(3, 5, 8)*365,
    survat, time=ttodth, event=death))
list(qbelow=qbelow, pbelow=round(pbelow, 2),
    qabove=qabove, pabove=round(pabove, 2),
    threshold=threshold)
}

> analyse.crpstratum(2) ##to check
> analyse.crpstratum(3)
$qbelow
[1] 1659  2647   NA

$pbelow
1097 1828 2922
Data analysis

0.95 0.88 0.77

$qabove
[1] 940 1813 2245

$pabove
1105 1826 2922
0.88 0.80 0.65

$threshold
[1] 3

Warning messages:
1: no non-missing arguments to min; returning Inf
2: no non-missing arguments to min; returning Inf
3: NAs introduced by coercion
Data analysis

> analyse.crpstratum(5)
$qbelow
[1] 1613 2590 NA

$pbelow
1097 1828 2922
0.94 0.88 0.76

$qabove
[1] 934 1769 2155

$pabove
1105 1826 2922
0.88 0.79 0.62

$threshold
Data analysis

[1] 5

Warning messages:
1: no non-missing arguments to min; returning Inf
2: no non-missing arguments to min; returning Inf
3: NAs introduced by coercion
Data analysis

The graph shows the relationship between km$time and km$surv for two groups: CRP <= 3 (solid line) and CRP > 3 (dashed line). The graph indicates a decreasing trend in km$surv over time for both groups, with the CRP <= 3 group generally having a higher survival rate compared to the CRP > 3 group.
Data analysis
Data analysis

Another interesting graph is a scatterplot of CRP by survival time, colored by censoring

> plot(log(crp)~ttodth,col=ifelse(death==1,"red","gray"),
   pch=19,data=inflamm)
Data analysis
Data analysis

For the sensitivity and specificity we read ahead and notice that three cutpoints are used, so we start off by writing a function

```r
diagnostics<-function(threshold){
    has.crp<-subset(inflamm, !is.na(crp))
    prevpos<-with(has.crp, mean(ttodth<3*365 & death==1))
    prevevent<-with(has.crp, mean(crp>threshold))
    sens<-with(has.crp, mean(ttodth<3*365 & death==1 & crp>threshold)/mean(ttodth<3*365 & death==1))
    spec<-with(has.crp, mean(!(ttodth<3*365 & death==1) & !(crp>threshold))/mean(!(ttodth<3*365 & death==1)))
    ppv<-with(has.crp, mean(ttodth<3*365 & death==1 & crp>threshold)/mean(crp>threshold))
    npv<-with(has.crp, mean(!(ttodth<3*365 & death==1) & !(crp>threshold))/mean(!(crp>threshold)))

    list(prevpos=prevpos, prevent=prevevent,
         sens=sens, spec=spec,
         ppv=ppv, npv=npv)
}
```

Now run the function at the three thresholds
> diagnostics(threshold=2)
$prevpos
[1] 0.06770728
$prevent
[1] 0.380296
$sens
[1] 0.5718563
$spec
[1] 0.633616
$ppv
[1] 0.1018124
$npv
[1] 0.9532221

> diagnostics(threshold=3)
$prevpos
Data analysis

[1] 0.06770728
$prevent
[1] 0.2381918
$sens
[1] 0.4191617
$spec
[1] 0.774951
$ppv
[1] 0.1191489
$npv
[1] 0.9483768

> diagnostics(threshold=5)
$prevpos
[1] 0.06770728
$prevent
We might also look at the ROC curve, since we programmed that earlier.

```r
ROC <- function(test, disease){
  cutpoints <- c(-Inf, sort(unique(test)), Inf)
  sensitivity<-sapply(cutpoints,
    function(result) mean(test>result & disease)/mean(disease))
  specificity<-sapply(cutpoints,
    function(result) mean(disease | result) / mean(disease))
}```
Data analysis

function(result) mean(test<=result & !disease)/mean(!disease))
  return(list(sens=sensitivity, spec=specificity, cutpoints=cutpoints))
}

crproc<-with(has.crp, ROC(crp, ttodth<3*365 & death==1))
plot(1-crproc$spec,crproc$sens,xlab="1 - specificity",
     ylab="Sensitivity",type="l")

marks<-match(c(2,3,5), crproc$cutpoints)
points(1-crproc$spec[marks], crproc$sens[marks], col="red",pch=19)
Data analysis