F Distribution: An investigation on the ratio of variances

Eamonn O'Brien

31 July, 2016

1 Introduction

Consider two sample variances that are calculated from random samples from different normal populations.

If we need to perfom a significance test to determine whether the underlying variances are in fact equal; that is, we want to test the hypothesis H_0: $\sigma_1^2 = \sigma_2^2$ versus $H_1 : \sigma_1^2$!= σ_2^2 we will proceed basing the significance test on the relative magnitudes of the sample variances (s_1^2, s_2^2). It is prefereable to base the test on the ratio of the sample variances (s_1^2 / s_2^2) rather than on the difference between the sample variances (s_1^2- s_2^2).

The ratio of two such variances is called an F ratio and the F ratio has a standard distribution called an F distribution. The shape of this distribution depends on the sample sizes of the two groups more generally on the degrees of freedom of the two variance estimates. The variance ratio follows an F distribution under the null hypothesis that $\sigma_1^2 = \sigma_2^2$ and is indexed by the two parameters termed the numerator and denominator degrees of freedom, respectively. If the sizes of the first and second samples are n1 and n2 respectively, then the variance ratio follows an F distribution with n1-1 (numerator df) and n2-1 (denominator df), which is called an $F_{(n-1),(n-2)}$ distribution. If the two normal populations have different standard deviations, the F distribution is scaled by their ratio. However if the two groups really have the same population standard deviations, the distribution does not involve any unknown parameters.

First using simulation we explore the case where the two normal populations have different standard deviations. Later we show how to perform the F test and demonstrate caution is required when using bootstrap and simulations with small sample sizes to test equality of variances. More extensive testing is needed but the need for caution is demonstrated.

1.1 Population and sample size

```
s1 <- 1.0                                   # true standard deviation population 1
s2 <- 1.1                                   # true standard deviation population 2
ratio <- s1^2 / s2^2                        # true ratio of population variances

n1 <- 3                                     # (small) sample size from population 1
n2 <- 4                                     # (small) sample size from population 2

mu <- 0                                     # Common mean, not important
n.sim <- 10^5                               # Number of simulations
```

1.2 Simulate many samples from populations

```
z <- matrix(rnorm(n1 * n.sim, mean = mu, sd = s1), ncol = n.sim)
y <- matrix(rnorm(n2 * n.sim, mean = mu, sd = s2), ncol = n.sim)

z[1:n1, 1:5]  # examine the data
```

```
           [,1]       [,2]       [,3]       [,4]       [,5]
[1,] -0.5604756 0.07050839  0.4609162 -0.4456620  0.4007715
[2,] -0.2301775 0.12928774 -1.2650612  1.2240818  0.1106827
[3,]  1.5587083 1.71506499 -0.6868529  0.3598138 -0.5558411
```

```
y[1:n2, 1:5]  # examine the data
```

```
            [,1]       [,2]       [,3]       [,4]       [,5]
[1,] -0.08034955 -1.4478277 -0.5582026 -0.7114229  0.2057159
[2,]  0.78241702  0.8387769 -0.4408669  0.3067864  0.3633381
[3,]  0.09088873  0.9939460  1.0857952  0.1329912 -1.3299782
[4,] -1.22998392  1.8451593 -1.3451161 -1.3958137  2.1122425
```

1.3 Calculate the variance for each sample and calculate the ratio of the variances

```
num <- apply(z, 2, var)
den <- apply(y, 2, var)

head(num)  # examine the data
```

```
[1] 1.3000250 0.8704518 0.7717828 0.6972991 0.2405855 3.6373788
```

```
head(den)  # examine the data
```

```
[1] 0.6973351 1.9830024 1.0327852 0.6237060 1.9827430 2.8720180
```

```
Fsim <- num/den

head(Fsim)  # examine the ratio of the data
```

```
[1] 1.8642758 0.4389565 0.7472830 1.1179933 0.1213397 1.2664889
```

1.4 Plot the distribution of the ratios and overlay the F distribution that is dictated by the sample sizes

```
hist((Fsim), probability=TRUE, breaks=75, col = rainbow(75),
     main=paste("F distribution s1^2 / s2^2 \ntruth sd(s1)=",s1, "sd(s2)=",s2,sep=" "))

# and multiply s2^2/s1^2 just to get height of curve correct
curve( df(x,  n1-1, n2-1)*1/ratio,
       add=TRUE, from=(min(Fsim)),
       to= (max(Fsim)), col="black", lwd=2)

abline(v=(qf(0.975,n1-1,n2-1)), col='blue')
abline(v=(qf(0.025,n1-1,n2-1)), col='blue')
```

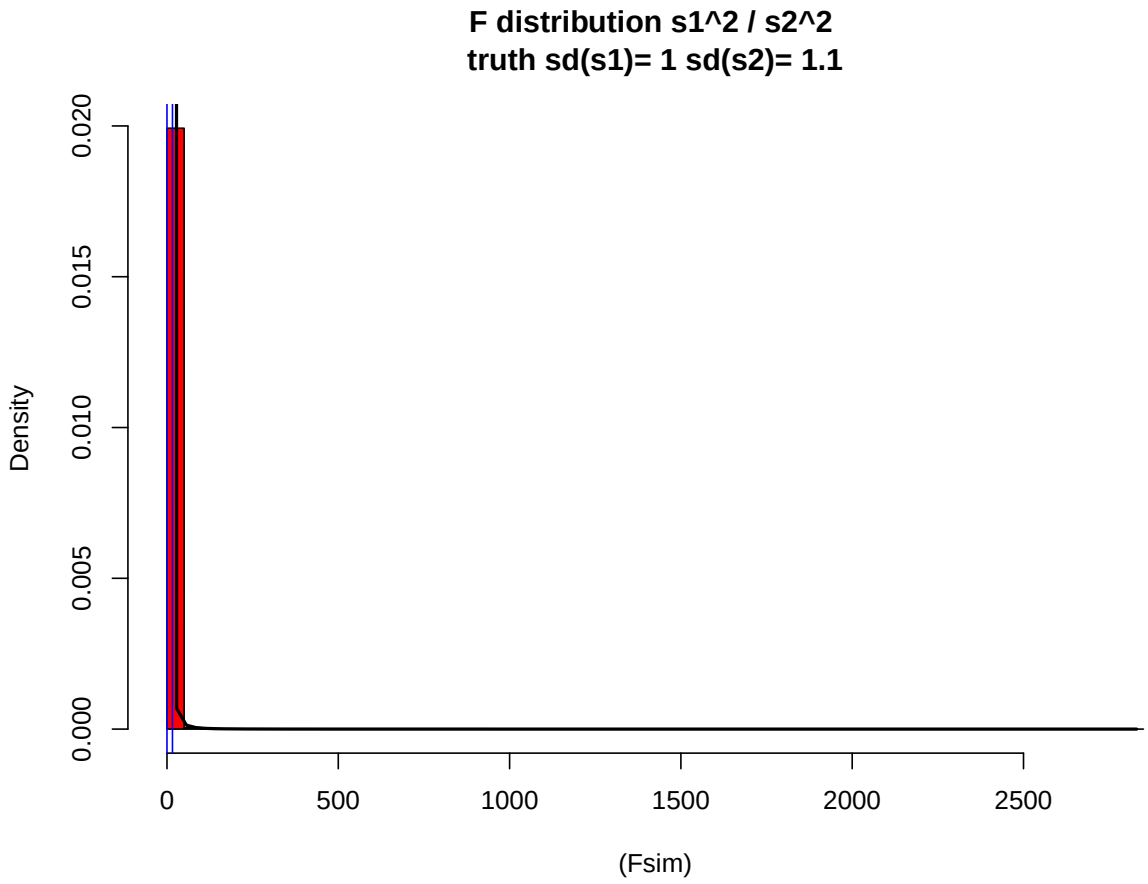

Figure 1: The small sample sizes result in a very skewed F distribution, so it is hard to tell if the theoretical curve fits the data. The blue lines are the 95% percentiles for testing equality of variances.

1.5 Plot the distribution of the ratios and overlay the F distribution that is dictated by the sample sizes, this time the F distribution is scaled by the true SDs. This is a toy example as we never have the luxury of knowing the true population SDs

```
hist((Fsim ), probability=TRUE, breaks=75, col = rainbow(75),
    main=paste("F distribution scaled by s1^2 / s2^2 \ntruth sd(s1)=",s1, "sd(s2)=",s2,sep=" "))

# scale the f distribution by the ratio of the true variances
# and multiply s2^2/s1^2 just to get height of curve correct
curve(df(x/(ratio), n1-1, n2-1)* 1/(ratio),
      add=TRUE, from=(min(Fsim)),
      to= (max(Fsim)), col="black", lwd=2)

abline(v= (qf(0.975,n1-1,n2-1)*ratio), col='blue')
abline(v= (qf(0.025,n1-1,n2-1)*ratio), col='blue')
```

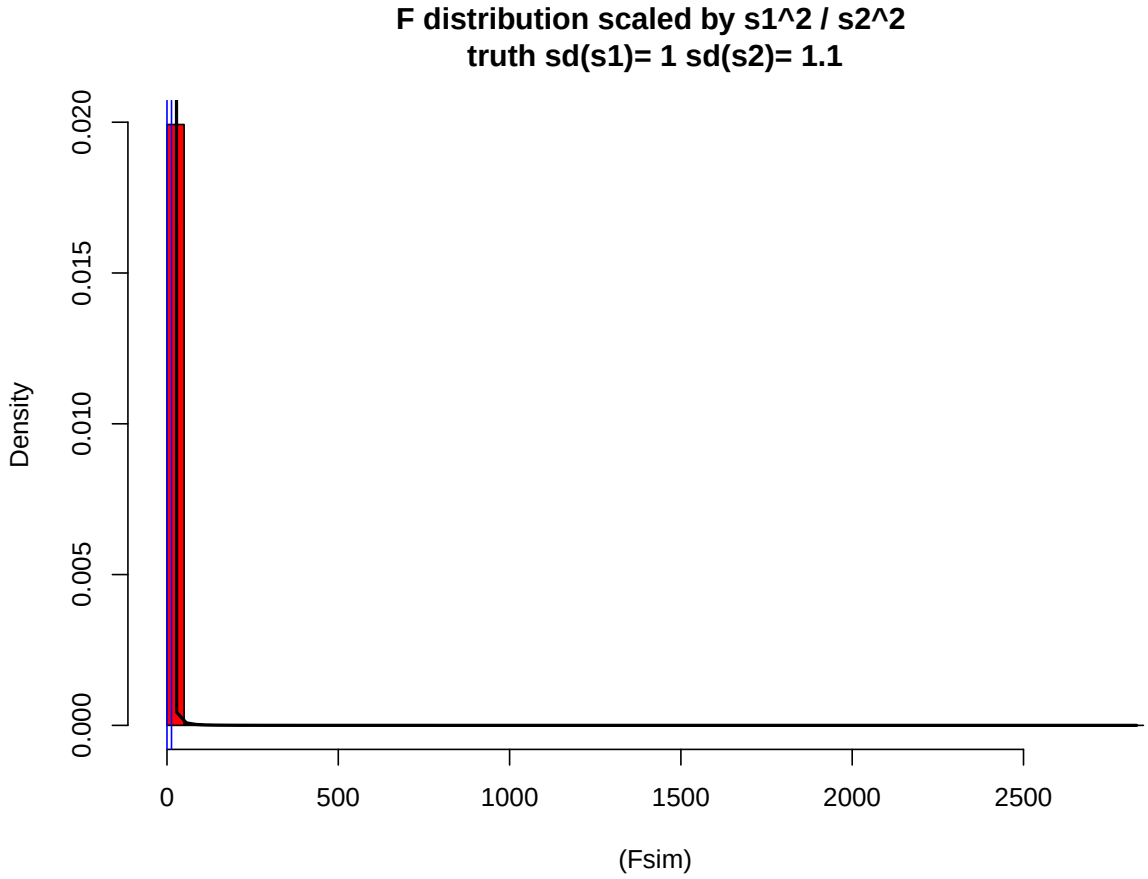

Figure 2: The small sample size results in a very skewed F distribution, so it is hard to tell if the scaled theoretical curve fits the data. The blue lines are the 95% percentiles for testing equality of variances.

1.6 On the log scale plot the distribution of the ratios and overlay the F distribution that is dictated by the sample sizes

```
hist(log(Fsim), probability=TRUE, breaks=75, col = rainbow(75), # log scale
      main=paste("F distribution s1^2 / s2^2 \ntruth sd(s1)=",s1, "sd(s2)=",s2,sep=" "))

# and multiply s2^2/s1^2 just to get height of curve correct
curve( df(exp(x),  n1-1, n2-1)*exp(x)*(1/ratio)  ,
       add=TRUE, from=log(min(Fsim)),
       to= log(max(Fsim)), col="black", lwd=2)

abline(v=log(qf(0.975,n1-1,n2-1)), col='blue')
abline(v=log(qf(0.025,n1-1,n2-1)), col='blue')
```

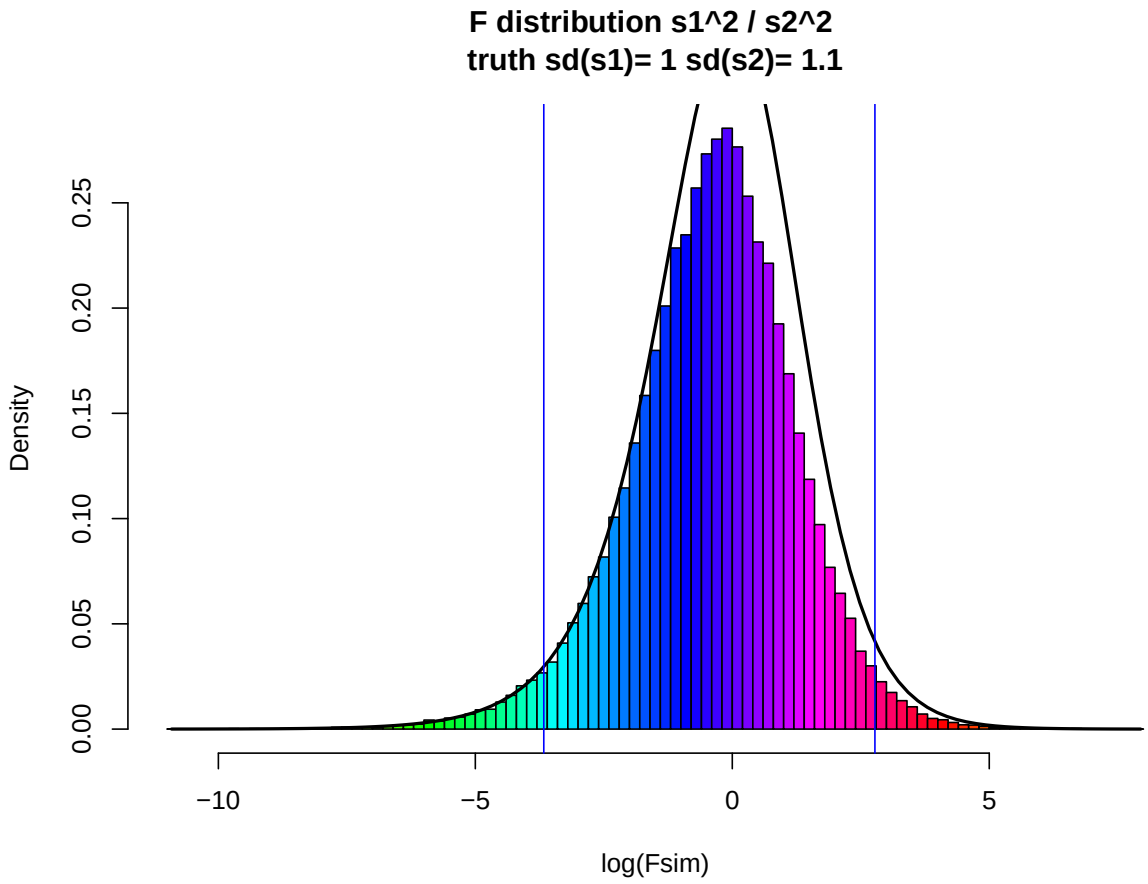

Figure 3: On the log scale the comparison is made clearer and it is apparent the theoretical curve is not a good fit for the data. The blue lines are the 95% percentiles for testing equality of variances.

1.7 On the log scale plot the distribution of the ratios and overlay the F distribution that is dictated by the sample sizes, this time the F distribution is scaled by the true SDs. This is a toy example as we never have the luxury of knowing the true population SDs.

```
hist(log(Fsim ), probability=TRUE, breaks=75, col = rainbow(75),   # log scale
     main=paste("F distribution scaled by s1^2 / s2^2 \ntruth sd(s1)=",s1, "sd(s2)=",s2,sep=" "))

# scale the f distribution by the ratio of the true variances
# and multiply s2^2/s1^2 just to get height of curve correct
curve(df(exp(x)/(ratio), n1-1, n2-1)* exp(x)*(1/(ratio)),
      add=TRUE, from=log(min(Fsim)),
      to= log(max(Fsim)), col="black", lwd=2)

abline(v=log(qf(0.975,n1-1,n2-1) ), col='blue');
abline(v=log(qf(0.975,n1-1,n2-1)*ratio), col='red')
abline(v=log(qf(0.025,n1-1,n2-1) ), col='blue');
abline(v=log(qf(0.025,n1-1,n2-1)*ratio), col='red')
```

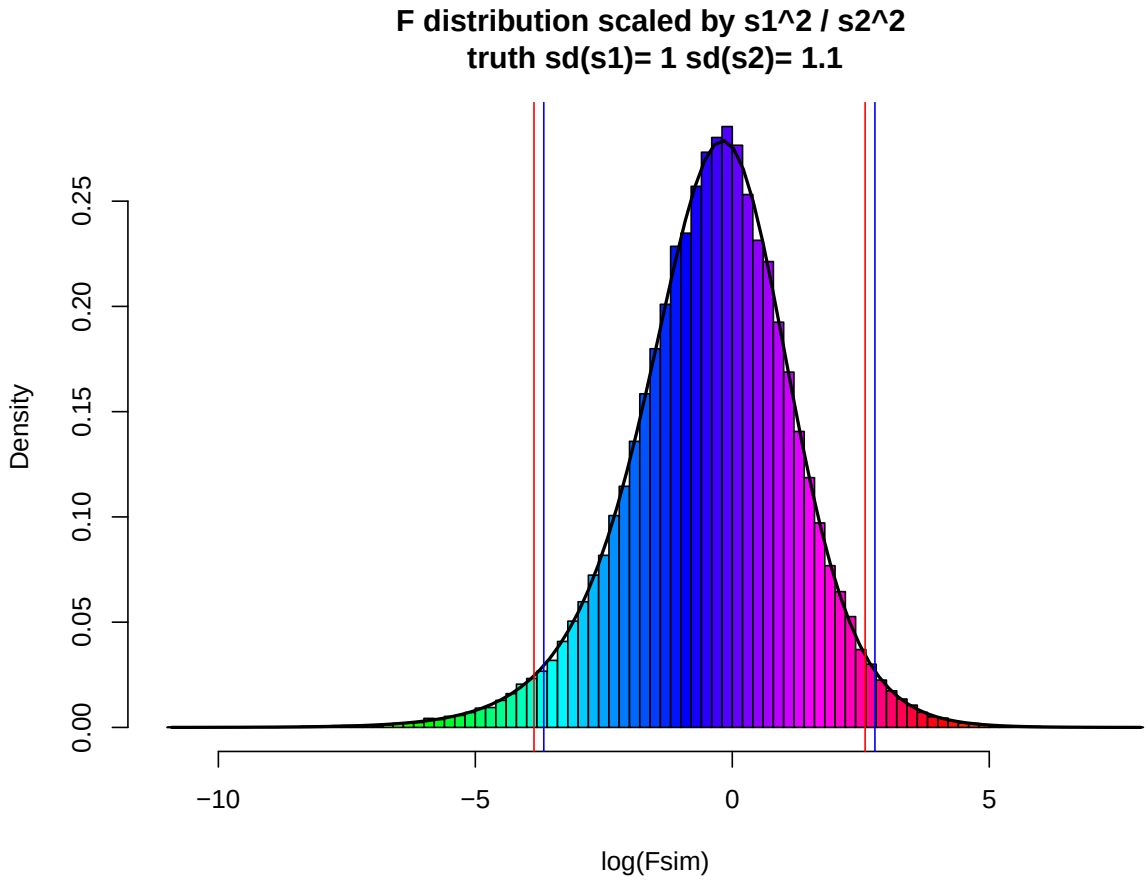

Figure 4: On the log scale the comparison is made clearer and it is apparent the scaled theoretical curve is a good fit for the data, as it should be. The blue lines are the 95% percentiles for testing equality of variances. The red lines are the 95% percentiles for testing the population variance null of s1^2 / S2^2 = 1.0/1.1

1.8 What proportion of results are in the tails of the distributions? First the unscaled. This should not be ~2.5% in each as we know the population variances are not equal

```
low1 <- qf(0.025,n1-1,n2-1)
upp1 <- qf(0.975,n1-1,n2-1)

length(Fsim[Fsim>upp1]) / n.sim
```

[1] 0.01911

```
length(Fsim[Fsim<low1]) / n.sim
```

[1] 0.02979

1.9 Now the scaled. This should be ~2.5% in each as we know the population variances are not equal and have scaled accordingly

```
low2 <- qf(0.025,n1-1,n2-1) * ratio
upp2 <- qf(0.975,n1-1,n2-1) * ratio

length(Fsim[Fsim>upp2]) / n.sim
```

[1] 0.02506

```
length(Fsim[Fsim<low2]) / n.sim
```

[1] 0.025

2 Move on to look at testing equality of variance. Here is a function to calculate the F test p value, the actual samples from the population must be entered into the function.

```
# enter each sample
variance.ratio<-function (x, y) {

  if (var(x) > var(y)) {

    vr <- var(x)/var(y)
    df1 <- length(x)-1
    df2 <- length(y)-1

    } else {

    vr <- var(y)/var(x)
    df1 <- length(y)-1
    df2 <- length(x)-1
    }

    2*(1-pf(vr,df1,df2))
  }
```

2.1 Function to calculate F test p value and ratio confidence interval, the variance and df for each sample are required.

```
# enter each variance and each degrees of freedom
var.rat <- function (v1, df1, v2, df2) {
  V.x <- v1
  DF.x <- df1
  V.y <- v2
  DF.y <- df2
  ratio <- 1
  conf.level <- 0.95
  ESTIMATE <- V.x/V.y
  STATISTIC <- ESTIMATE/ratio
  PARAMETER <- c( DF.x,  DF.y)
  PVAL <- pf(STATISTIC, DF.x, DF.y)
  PVAL <- 2 * min(PVAL, 1 - PVAL)
  BETA <- (1 - conf.level)/2
  CINT <- c(ESTIMATE/qf(1 - BETA, DF.x, DF.y),
            ESTIMATE/qf(BETA, DF.x, DF.y))
  c(ESTIMATE, CINT, PVAL)
}
```

2.2 Let's perform the F test manually, create some data, note very small sample sizes

```
s1 <- 10:12 ; s2 <- 13:16
n1 <- length(s1) ; n2 <- length(s2)
```

2.3 Manual F test, and options to calculate the ratio confidence limits. The symmetry properties of the F distribution make it possible to derive the lower percentage points of any F distribution from the corresponding upper percentage points of an F distribution with the degrees of freedom reversed.

```
(vr <- var(s1)/var(s2))     # ratio of variances
```

[1] 0.6

```
vr*qf(0.025, n2-1, n1-1)    # lower
```

[1] 0.03739691

```
vr*qf(0.975, n2-1, n1-1)    # upper
```

[1] 23.4993

```
vr/qf(0.975, n1-1, n2-1)    # lower
```

[1] 0.03739691

```
vr/qf(0.025, n1-1, n2-1)    # upper
```

[1] 23.4993

2.4 F test using functions

```
      # base R function, requires the actual samples s1 and s2 in our example
      var.test(s1, s2)
```

```
    F test to compare two variances

data:  s1 and s2
F = 0.6, num df = 2, denom df = 3, p-value = 0.7926
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
  0.03739691 23.49929674
sample estimates:
ratio of variances
               0.6
```

```
      # function defined earlier, returns only the p value for test of null hypthesis var(a)=var(b)
      variance.ratio(s1,s2)
```

```
[1] 0.7926368
```

```
      # function defined earlier
      var.rat(var(s1), n1-1, var(s2), n2-1)
```

```
[1]  0.60000000  0.03739691 23.49929674  0.79263678
```

2.5 Simulation is not advisable with small samples!

```
   n.sim <- 10^4

    x<-replicate(n.sim,
                 var( rnorm(n1, 0, sd(s1))) /
                 var( rnorm(n2, 0, sd(s2)))
   )
   quantile(x, c(0.025, 0.975))
```

```
  2.5%      97.5%
```

0.01676385 9.58545142

2.6 Simulation again is not advisable with small samples!

```
     x <- matrix(rnorm(n1*n.sim, mean=0, sd=sd(s1)), ncol=n.sim)
     y <- matrix(rnorm(n2*n.sim, mean=0, sd=sd(s2)), ncol=n.sim)
     num <-  apply(x, 2, var)
     den  <- apply(y, 2, var)
     Fsim <- num / den
     quantile(Fsim , c(0.025, 0.975))
```

```
  2.5%      97.5%
```

0.01474067 9.73194214

2.7 Bootstrap is not advisable with small samples!

```
     bootstrapStat = rep(NA,n.sim)

     for (i in 1: n.sim) {
           bootstrapStat[i] = c( var(sample(s1, replace=T)) / var(sample(s2, replace=T)) )
     }

     quantile( bootstrapStat,c(0.025, .975), na.rm=T)
```

```
    2.5%    97.5%
0.000000 5.333333
```

3 Set up an example with larger sample sizes

```
    n1 <- 30
    n2 <- 40
    s1 <- rnorm(n1, 0, 1)
    s2 <- rnorm(n2, 0, 2)
    n1 <- length(s1)
    n2 <- length(s2)

    n.sim <- 10^4
```

3.1 Base var.test function

```
    var.test(s1, s2)
```

```
    F test to compare two variances

data:  s1 and s2
F = 0.20321, num df = 29, denom df = 39, p-value = 2.748e-05
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1035793 0.4130559
sample estimates:
ratio of variances
         0.2032089
```

3.2 Simulation 1

```
    x<-replicate(n.sim,
                 var( rnorm(n1, 0, sd(s1))) /
                 var( rnorm(n2, 0, sd(s2)))
    )
    quantile(x, c(0.025, 0.975))
```

```
 2.5%     97.5%
```

0.1001230 0.4033721

3.3 Simulation again with larger samples

```
x <- matrix(rnorm(n1*n.sim, mean=0, sd=sd(s1)), ncol=n.sim)
y <- matrix(rnorm(n2*n.sim, mean=0, sd=sd(s2)), ncol=n.sim)
num <-  apply(x, 2, var)
den  <- apply(y, 2, var)
Fsim <- num / den
quantile(Fsim , c(0.025, 0.975))
```

```
 2.5%     97.5%
```

0.1009032 0.4019109

3.4 Bootstrap with larger samples

```
bootstrapStat = rep(NA,n.sim)

for (i in 1: n.sim) {
      bootstrapStat[i] = c( var(sample(s1, replace=T)) / var(sample(s2, replace=T)) )
}

quantile( bootstrapStat,c(0.025, .975), na.rm=T)
```

```
      2.5%      97.5%
0.09600325 0.48338261
```

3.5 Bootstrap with larger samples, more examples of doing the same thing, namely bootstrapping

```
x <- s1
y <- s2

ratio <- function(d, i) {var(x[i]) / var(y[i])}
bb<-boot(x, ratio, R=n.sim, stype="i")
boot.ci(bb )
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 10000 bootstrap replicates

CALL :
boot.ci(boot.out = bb)

Intervals :
Level      Normal              Basic
95%   (-0.0871,  0.3667 )   (-0.1588,  0.2785 )

Level     Percentile            BCa
95%   ( 0.0773,  0.5146 )   ( 0.0631,  0.4218 )
Calculations and Intervals on Original Scale
```

3.6 Bootstrap with larger samples, more examples of doing the same thing, namely bootstrapping

```
ratio <- function(d, i) {var(rnorm( n1,0,sd(s1)) * i) / var(rnorm( n2,0,sd(s2)) * i)}
bb<-boot(x, ratio, R=n.sim, stype="i")
boot.ci(bb )
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 10000 bootstrap replicates

CALL :
boot.ci(boot.out = bb)

Intervals :
Level      Normal              Basic
95%   ( 0.3528,  0.7714 )   ( 0.2917,  0.7035 )

Level     Percentile            BCa
95%   ( 0.0797,  0.4915 )   ( 0.3106,  1.0020 )
Calculations and Intervals on Original Scale
Warning : BCa Intervals used Extreme Quantiles
Some BCa intervals may be unstable
```

3.7 Bootstrap with larger samples, more examples of doing the same thing, namely bootstrapping

```
y1 <- c(x,y)
group <- c(rep(1, each=n1),rep(2, each=n2))
xx <- as.data.frame(cbind(group, y1))

b<-NULL
b <- boot(data=x,
          statistic = function(x, i) {
            booty <- tapply( xx$y, xx$group, FUN=function(x) sample(x, length(x),TRUE))
            1/exp(diff(log(sapply(booty, var) )))  # take the difference of the log variances
          },
          R=n.sim)
boot.ci(b)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 10000 bootstrap replicates

CALL :
boot.ci(boot.out = b)

Intervals :
Level      Normal              Basic
95%   ( 0.3920,  0.7971 )   ( 0.3360,  0.7307 )

Level     Percentile            BCa
95%   ( 0.0954,  0.4901 )   ( 0.3464,  1.2161 )
Calculations and Intervals on Original Scale
Warning : BCa Intervals used Extreme Quantiles
Some BCa intervals may be unstable
```

3.8 Bootstrap with larger samples, more examples of doing the same thing, namely bootstrapping

```
b1 <- bootstrap(x, n.sim, var)
b2 <- bootstrap(y, n.sim, var)
rat <- b1$thetastar/ b2$thetastar
quantile(rat , c(.025, .975), na.rm=T)
```

```
     2.5%     97.5%
0.0950275 0.4840457
```

4 References

https://github.com/eamonn2014/programs/blob/master/F%20DISTRIBUTION.Rmd
https://www.safaribooksonline.com/library/view/the-r-book/9780470510247/ch002-sec049.html
http://www.ncss.com/wp-content/themes/ncss/pdf/Procedures/PASS/Confidence_Intervals_for_the_Ratio_of_Two_Variances_using_Variances.pdf
https://stat.ethz.ch/R-manual/R-devel/library/stats/html/var.test.html
http://stackoverflow.com/questions/18255757/is-it-possible-to-pass-samples-of-unequal-size-to-function-boot-in-r
https://cran.r-project.org/web/packages/bootstrap/bootstrap.pdf p20

5 Computing Environment

```
R version 3.2.2 (2015-08-14)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 8 x64 (build 9200)

locale:
[1] LC_COLLATE=English_United Kingdom.1252
[2] LC_CTYPE=English_United Kingdom.1252
[3] LC_MONETARY=English_United Kingdom.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United Kingdom.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods
[7] base

other attached packages:
[1] bootstrap_2015.2 boot_1.3-17      knitr_1.13

loaded via a namespace (and not attached):
 [1] magrittr_1.5    formatR_1.4     tools_3.2.2     htmltools_0.3.5
 [5] yaml_2.1.13     Rcpp_0.12.6     stringi_1.1.1   rmarkdown_1.0
 [9] stringr_1.0.0   digest_0.6.9    evaluate_0.9


[1] "~/X/"
```

This took 27.35 seconds to execute.