Math 3070 § 1. Treibergs | **f Simulation Example: Simulating p-Values of Two Sample Variance Test.** | Name: Example June 26, 2011

The t-test is fairly robust with regard to actual distribution of data. But the f-test is much less robust. To explore the dependence on distributions we simulate data from various distributions. We plot the histogram to appreciate the sampling distribution of the p-value for these tests.

We select random samples from various distributions. If the samples are normal $X_1, X_2, \ldots, X_{n_1} \sim \mathrm{N}(\mu_1, \sigma_1)$; $Y_1, Y_2, \ldots, X_{n_2} \sim \mathrm{N}(\mu_2, \sigma_2)$ from a normal distribution, to test the hypothesis $H_0 : \sigma_1 = \sigma_2$ vs. the alternative $H_a : \sigma_1 \neq \sigma_2$, one computes the f statistic,

$$F = \frac{\mathrm{var}(X)}{\mathrm{var}(Y)}$$

which is also a random variable which is distributed according to the f-distribution with $(n_1 - 1, n_2 - 1)$ degrees of freedom. In particular, any function of this is also a random variable, for example, the p-value of this two-tailed test is

$$P = \begin{cases} 2\mathtt{pf}(\mathtt{F}, \mathtt{n_1} - 1, \mathtt{n_2} - 1, \mathtt{lower.tail} = \mathtt{FALSE}), & \text{if } f \geq 1; \\ 2\mathtt{pf}(\mathtt{F}, \mathtt{n_1} - 1, \mathtt{n_2} - 1), & \text{if } f < 1. \end{cases}$$

where $F(x) = \mathrm{P}(f \leq x)$ is the cdf for f with $(n_1 - 1, n_2 - 1)$ degrees of freedom. The p-value is computed when the canned test is run

$$\mathtt{var.test(X, Y)\$p.value}$$

If the background distributions are both normal with $\sigma_1 = \sigma_2$, then the type I errors occur when P is small. The probability of a type I error is $\mathrm{P}(P \leq \alpha)$ for a significance level α test, namely, that the test shows that the mean is significantly above μ_0 (*i.e.,* we reject H_0), even though the sample was drawn from data satisfying the null hypothesis $X_i \sim \mathrm{N}(\mu_0, \sigma)$. It turns out that in ths case, the p-value is a uniform rv in $[0, 1]$ when $\sigma_1 = \sigma_2$, with an argument like the one given in the "Soporific Example," where the p-value of the on-sample, one-sided t-test is discussed.

I ran examples with $\mu_0 = 0$, $\sigma = 1$, samples of size $n_1 = 10$ and $n_2 = 7$ with $n = 10,000$ trials for various distributions. In our histograms the bar from 0 to .05 is drawn red. For example, when $\sigma_1 = \sigma_2$ and X, Y are normaql, the $P \sim \mathrm{U}(0, 1)$, the bars have nearly the same height and type I errors occurred 488 times or 4.88% of the time.

If one of the distributions is normal and the other one is one of the distributions exponential, t with $df = 4$, t with $df = 20$, or uniform, then the chances of a type one error increases. the worst was when one distribution is heavy tailed, t with $df = 4$, vs. one that is light-tailed, uniform. Curiously, however, if both distributions are uniform, then the type I error went down!

One more point is in order. Since we are testing the type I errors for different distributions, we need to make sure that the distributions all have unit variance. In the case of the normal distribution, we specify the mean and standard deviation, so the cdf and normal sample may be obtained by

$$\mathtt{dnorm(x, mu, 1)}; \qquad \mathtt{rnorm(10, mu, 1)}.$$

For the exponential distribution, the mean and standard deviations are both $1/\lambda$, so that we specify $\lambda = 1$ to get unit mean and standard deviation. The cdf and random sample may be obtained by

$$\mathtt{dexp(x, 1); rexp(10, 1)}.$$

For the uniform distribution $U(a, b)$ supported on the interval $[a, b]$, the mean and variance are

$$\mu = \frac{a + b}{2}; \qquad \sigma^2 = \frac{(b - a)^2}{12}.$$

To obtain $\mu = \sigma = 1$, we choose $a = 1 - \sqrt{3}$ and $b = 1 + \sqrt{3}$. The cdf and random sample may be obtained by

$$\texttt{dunif}(\texttt{x}, 1 - \texttt{sqrt}(3), 1 + \texttt{sqrt}(3)); \qquad \texttt{runif}(10, 1 - \texttt{sqrt}(3), 1 + \texttt{sqrt}(3)).$$

Finally, the standard t distribution $T \sim T(df = \nu)$ has mean zero but NOT unit variance. In fact, its variance for $\nu > 2$ is

$$\sigma^2 = \frac{\nu}{\nu - 2}$$

Thus, the standard cdf and standard random numbers have to be rescaled to get unit variance. For four degrees of freedom,

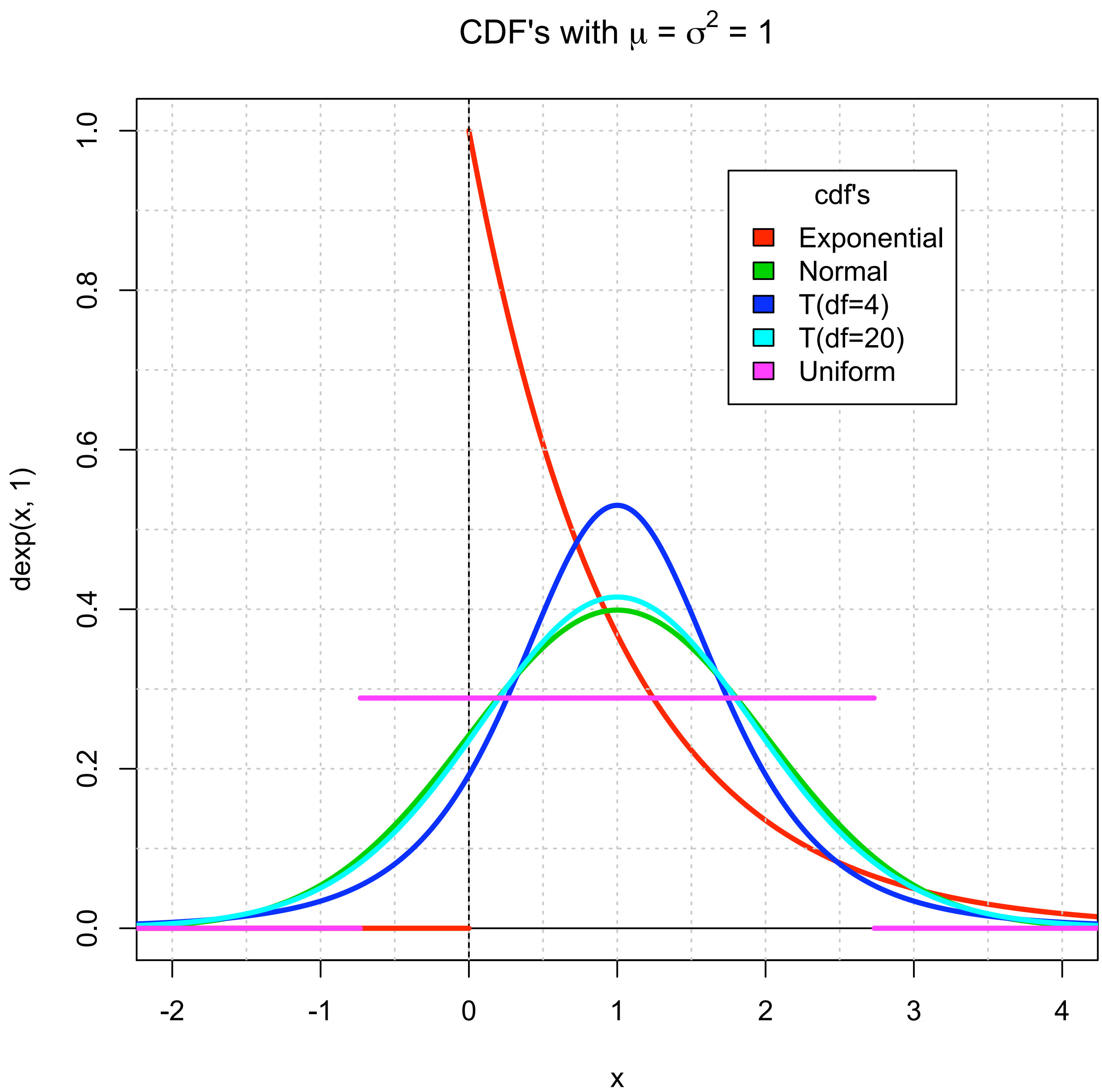
```
c <- sqrt(4/(4-2))
c * dt(c * x, 4);              rt(10, 4)/c.
```

We start our **R** study by deconstructing the two sample variance test.

R Session:

```
R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.31 (5538) powerpc-apple-darwin8.11.1]

[Workspace restored from /Users/andrejstreibergs/.RData]

> ################## P-VALUE FROM CANNED VAR TEST #########################
> x<- rnorm(10,2,1)
> y <- rnorm(7,3,1)
> v <-var.test(x,y)
> v

F test to compare two variances

data:  x and y
F = 1.758, num df = 9, denom df = 6,
p-value = 0.5068
alternative hypothesis: true ratio of variances is not equal to 1
```

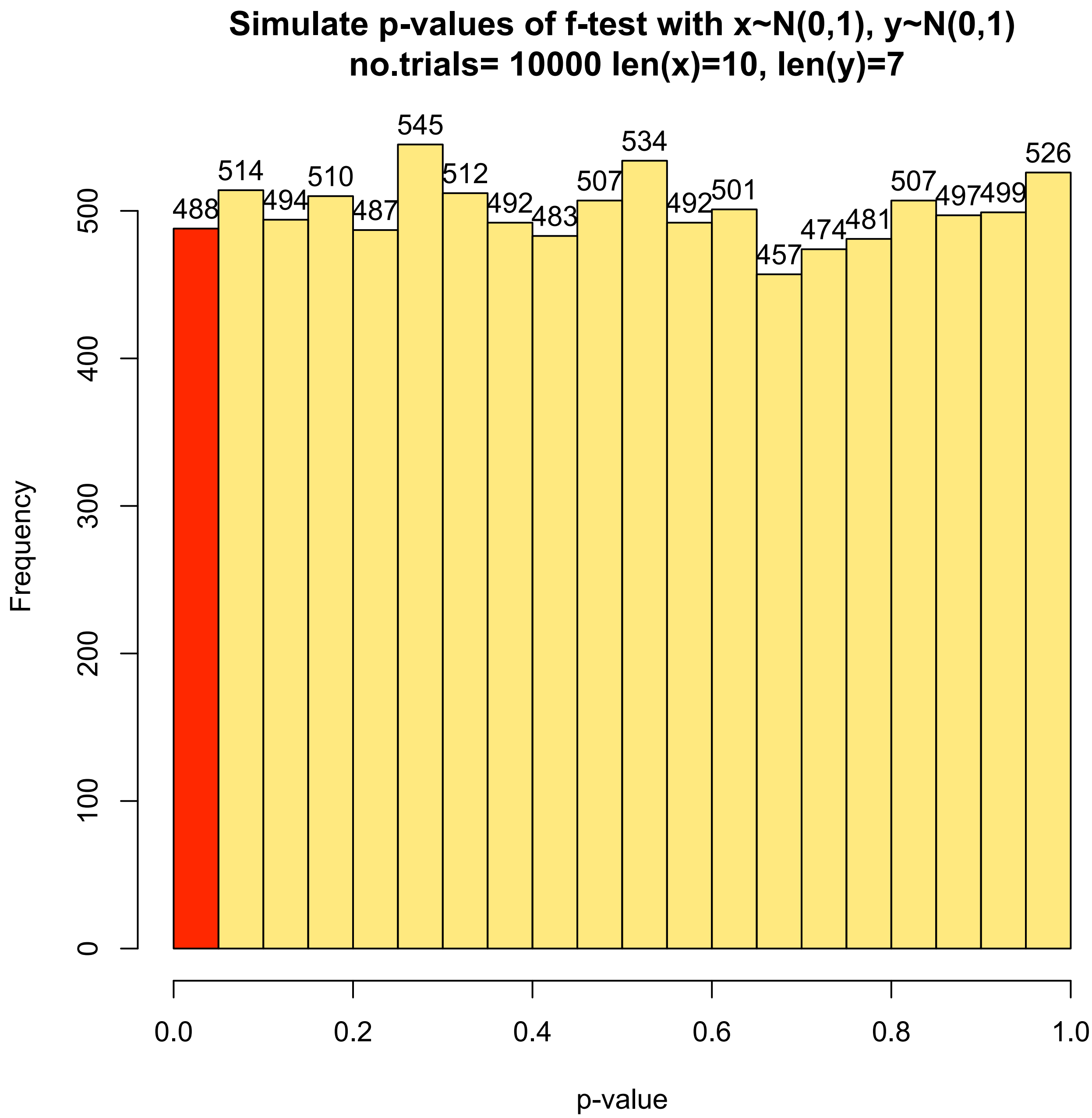
```
95 percent confidence interval:
 0.3182826 7.5940894
sample estimates:
ratio of variances
          1.758004

> # To extract the  p-value from the list
> v$p.value
[1] 0.5067721

> ################ P-VALUE BY HAND ########################################
> vx <- var(x)
> vy <- var(y)
> f <- vx/vy; f
[1] 1.758004
> 2*pf(f,nx-1,ny-1,lower.tail=F)
[1] 0.5067721

> ################ PLOT THE CDF'S OF THE DISTRIBUTIONS USED ###############
>
>
> x <- seq(0,4.3,1/77)
> plot(x, dexp(x,1), type="l", col=2, lwd=3,
+ main = expression(paste("CDF's with ", mu, " = ", sigma^2, " = 1")),
+ ylim = 0:1, xlim=c(-2,4))
> abline(h = 1:10/10, col=8, lty=3); abline(v = 0)
> abline(v = -4:8/2, col=8, lty=3); abline(h = 0)
> lines(c(-2.5,0), c(0,0), col=2, lwd=3)
> x <- seq(-2.5,4.5,1/55)
> lines(x, dnorm(x,1,1), col=3, lwd=3)
> c <- sqrt(4/(4-2))
> c
[1] 1.414214

> lines(x, c*dt(c*(x-1),4), col=4, lwd=3)
> c20 <- sqrt(20/(20-2))
> lines(x, c20*dt(c20*(x-1),20), col=5, lwd=3)
> a <- 1-sqrt(3); b <- 1+sqrt(3)
> lines(c(-2.5,a), c(0,0), col=6, lwd=3)
> lines(c(b,4.5), c(0,0), col=6, lwd=3)
> h <- 1/(b-a)
> lines(c(a,b), c(h,h), col=6, lwd=3)
> legend(1.75,.95, legend = c("Exponential","Normal","T(df=4)","T(df=20)",
+ "Uniform"), fill=2:6, bg="white", title="cdf's")
>
```

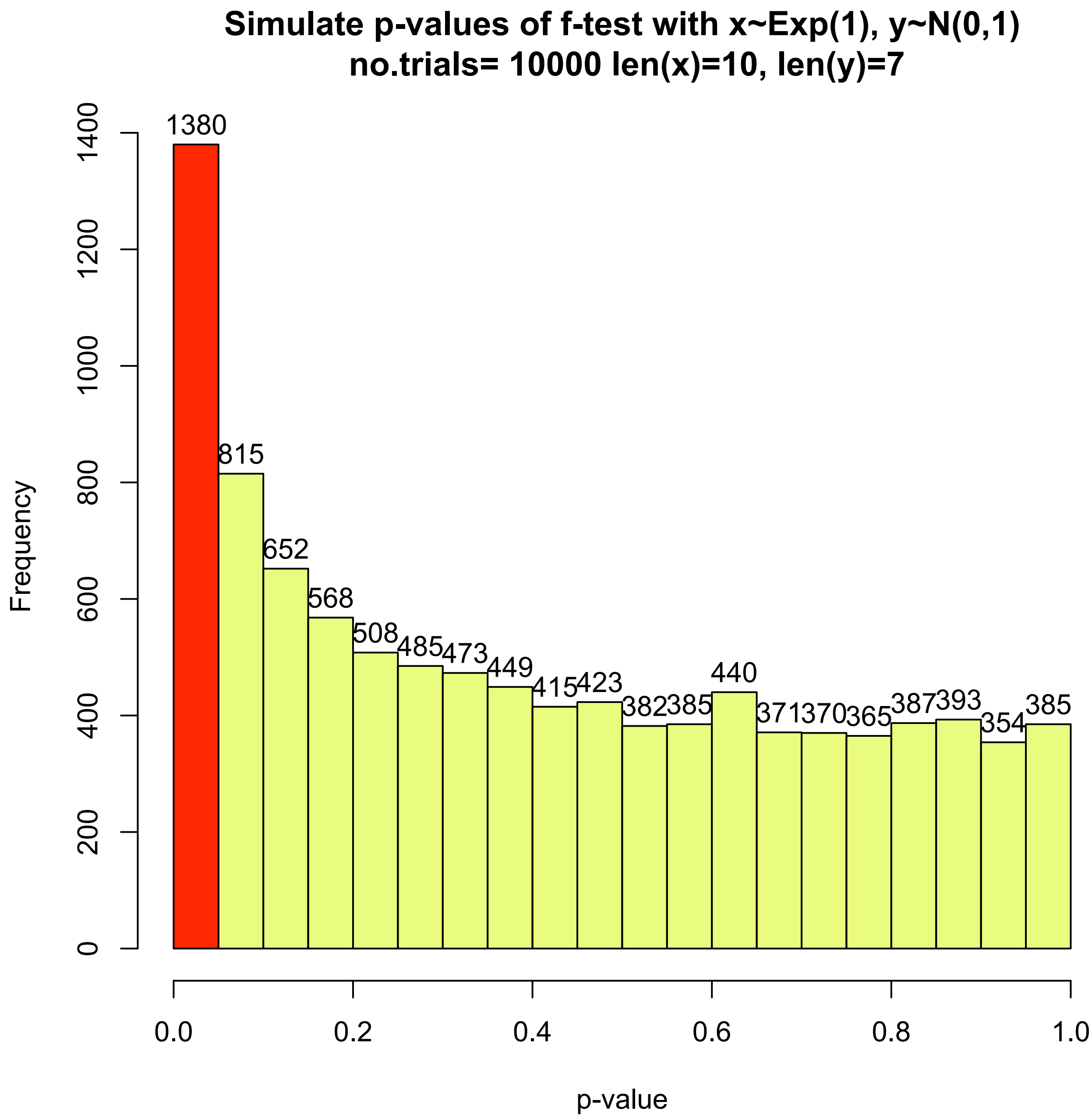

CDF's with $\mu = \sigma^2 = 1$
cdf's
Exponential
Normal
T(df=4)
T(df=20)
Uniform
dexp(x, 1)
0.0
0.2
0.4
0.6
0.8
1.0
-2
-1
0
1
2
3
4
x

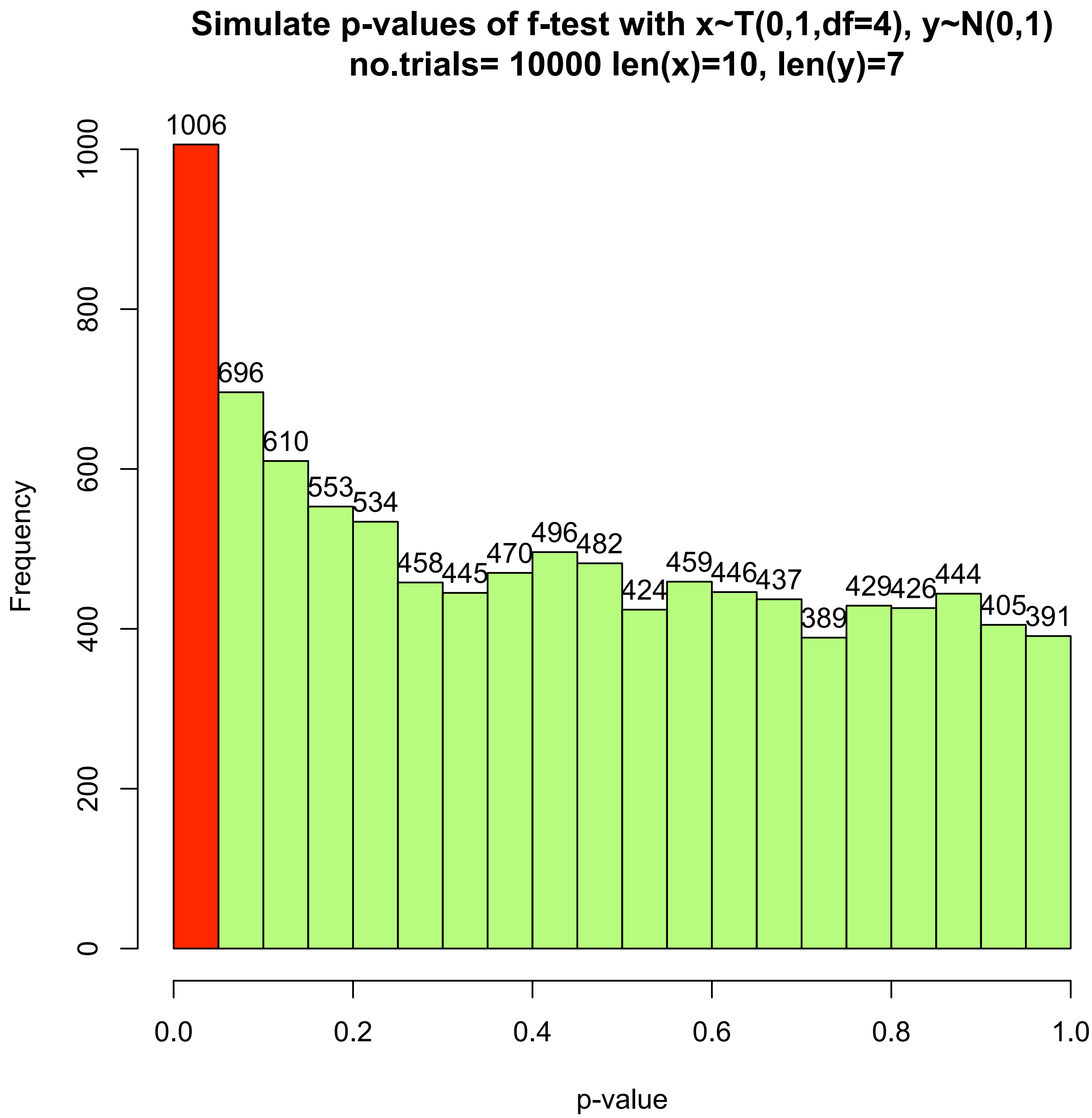
```
> #################### SIMULATE P-VALUES OF 2-SAMPLE VAR TEST ###############
>
> n <- 10000
> br <- seq(0,1,.05)
>
> #            NORMAL - NORMAL
>
> cl <- c(2,rep(rainbow(15, alpha=.5)[3], 19))
> mn <- paste("Simulate p-values of f-test with x~N(0,1), y~N(0,1)\n",
+ " no.trials=", n, "len(x)=10, len(y)=7")
> hist( replicate(n, var.test(rnorm(10),rnorm(7))$p.value), breaks=br, col=cl,
+ main=mn, xlab="p-value", labels=TRUE)
> # M3074fSim1.pdf
>
>
> #            EXPONENTIAL - NORMAL
>
> mn <- paste("Simulate p-values of f-test with x~Exp(1), y~N(0,1)\n",
+ "no.trials=", n, "len(x)=10, len(y)=7")
> cl <- c(2,rep(rainbow(15,alpha=.5)[4],19))
> hist(replicate(n,var.test(rexp(10,1), rnorm(7))$p.value), breaks=br, col=cl,
+ main=mn, xlab="p-value", labels=TRUE)
> # M3074fSim2.pdf
>
>
> #            T(df=4) - NORMAL
>
> mn <- paste("Simulate p-values of f-test with x~T(0,1,df=4), y~N(0,1)\n",
+ "no.trials=", n, "len(x)=10, len(y)=7")
> cl <- c(2, rep(rainbow(15,alpha=.5)[5],19))
> hist(replicate(n,var.test(rt(10,4)/c,rnorm(7))$p.value), breaks=br, col=cl,
+ main=mn, xlab="p-value", labels=TRUE)
> # M3074fSim3.pdf
>
>
> #            T(df=20) - NORMAL
>
> mn <- paste("Simulate p-values of f-test with x~T(0,1,df=20), y~N(0,1)\n",
+ " no.trials=", n, "len(x)=10, len(y)=7")
> cl <- c(2,rep(rainbow(15,alpha=.5)[6],19))
> hist(replicate(n, var.test(rt(10,20)/c20, rnorm(7))$p.value), breaks=br, col=cl,
+ main=mn, xlab="p-value", labels=TRUE)
> # M3074fSim4.pdf
```

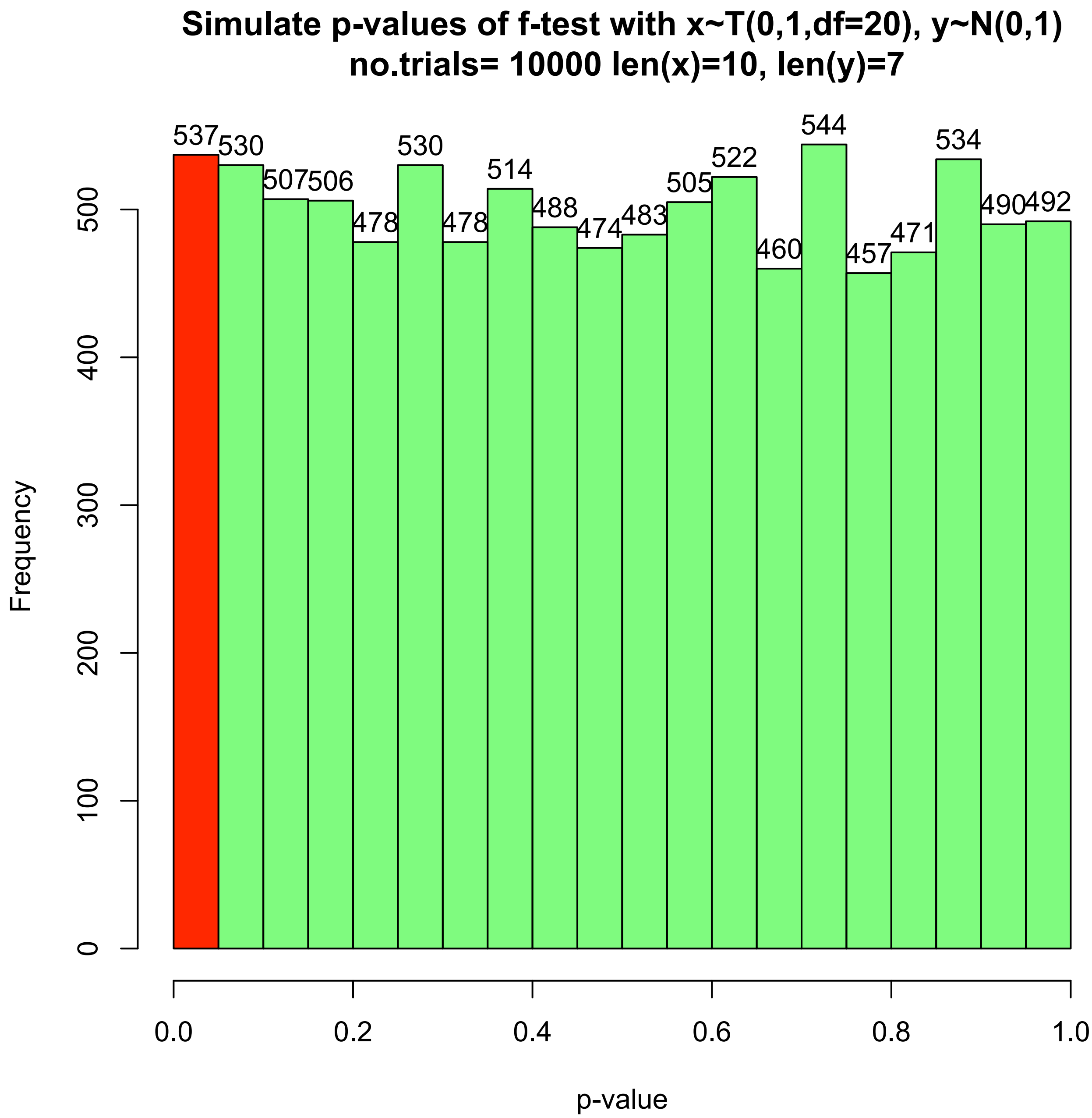
```
> #            UNIFORM - NORMAL
>
> mn <- paste("Simulate p-values of f-test with x~U(-sqrt(3),sqrt(3)), y~N(0,1)\n",
+ " no.trials=", n, "len(x)=10, len(y)=7")
> cl <- c(2, rep(rainbow(15, alpha=.5)[7],19))
> hist(replicate(n, var.test(runif(10,a,b), rnorm(7))$p.value), breaks=br, col=cl,
+ main=mn, xlab="p-value", labels=TRUE)
> # M3074fSim5.pdf
>
>
> #            EXPONENTIAL - EXPONENTIAL
>
> cl <- c(2,rep(rainbow(15,alpha=.5)[8],19))
> mn <- paste("Simulate p-values of f-test with x~Exp(1), y~Exp(1)\n",
+ " no.trials=", n,"len(x)=10, len(y)=7")
> hist(replicate(n,var.test(rexp(10,1), rexp(7,1))$p.value), breaks=br, col=cl,
+ main=mn, xlab="p-value", labels=TRUE)
> # M3074fSim6.pdf
>
>
> #            T(df=4) - T(df=4)
>
> cl <- c(2,rep(rainbow(15,alpha=.5)[9],19))
> mn <- paste("Simulate p-values of f-test with x~T(df=4), y~T(df=4)\n",
+ " no.trials=", n, "len(x)=10, len(y)=7")
> hist(replicate(n, var.test(rt(10,4), rt(7,4))$p.value), breaks=br, col=cl,
+ main=mn, xlab="p-value", labels=TRUE)
> # M3074fSim7.pdf
>
>
> #            T(df=20) - T(df=20)
>
> cl <- c(2,rep(rainbow(15,alpha=.5)[10],19))
> mn <- paste("Simulate p-values of f-test with x~T(df=20), y~T(df=20)\n",
+ "no.trials=",  n, "len(x)=10, len(y)=7")
> hist(replicate(n,var.test(rt(10,20), rt(7,20))$p.value), breaks=br, col=cl,
+ main=mn, xlab="p-value", labels=TRUE)
> # M3074fSim8.pdf
>
>
> #            UNIFORM - UNIFORM
>
> cl <- c(2,rep(rainbow(15,alpha=.5)[11],19))
> mn <- paste("Simulate p-values of f-test with x~U(0,1), y~U(0,1)\n",
+ " no.trials=", n, "len(x)=10, len(y)=7")
> hist(replicate(n,var.test(runif(10), runif(7))$p.value), breaks=br, col=cl,
+ main=mn, xlab="p-value", labels=TRUE)
> # M3074fSim9.pdf
```

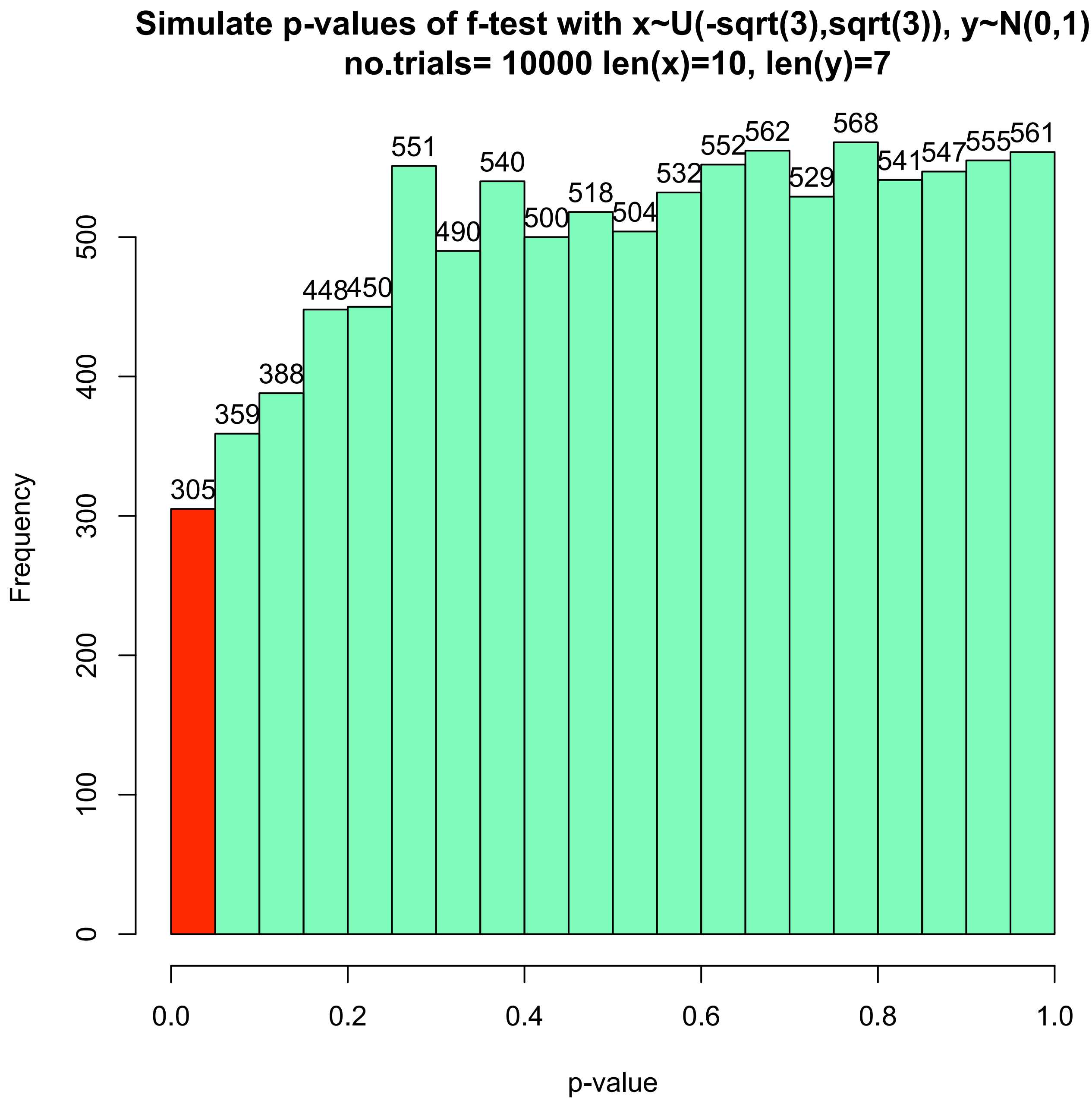
```
> #           UNIFORM - T(df=4)
>
> cl <- c(2,rep(rainbow(15,alpha=.5)[12],19))
> mn <- paste("Simulate p-values of f-test with x~U(-sqrt(3),sqrt(3)), y~T(0,1,df=4)\n",
+ " no.trials=", n,"len(x)=10, len(y)=7")
> hist(replicate(n,var.test(runif(10,a,b), rt(7,4)/c)$p.value), breaks=br, col=cl,
+ main=mn, xlab="p-value", labels=TRUE)
> # M3074fSim10.pdf
>
>
> #           UNIFORM - T(df=20)
>
> cl <- c(2,rep(rainbow(15,alpha=.5)[13],19))
> mn <- paste("Simulate p-values of f-test with x~U(-sqrt(3),sqrt(3)), y~T(0,1,df=20)\n",
+ " no.trials=", n, "len(x)=10, len(y)=7")
> hist(replicate(n,var.test(runif(10,a,b), rt(7,20)/c20)$p.value), breaks=br, col=cl,
+ main=mn, xlab="p-value", labels=TRUE)
> # M3074fSim11.pdf
>
```

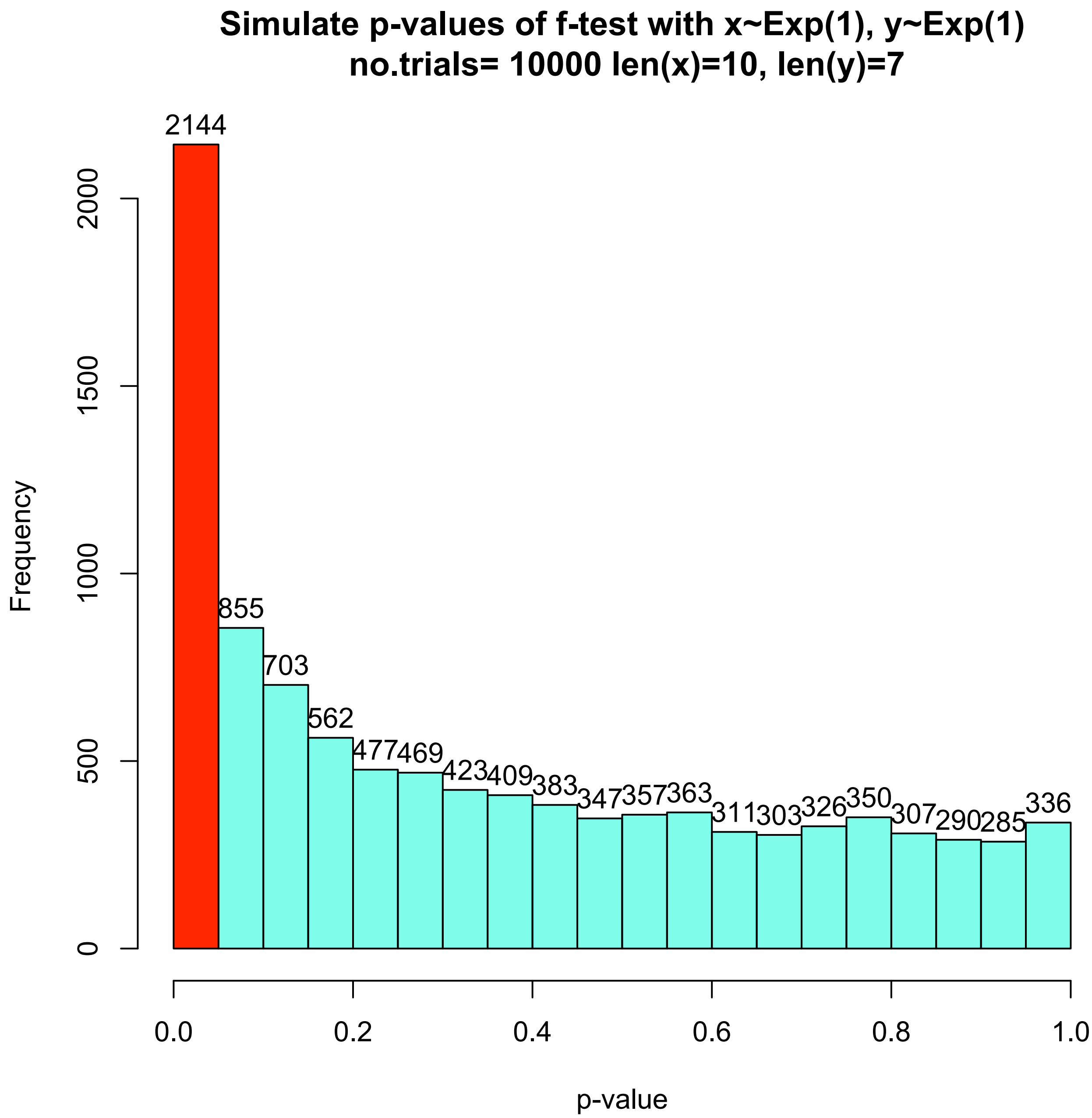

Simulate p-values of f-test with x~N(0,1), y~N(0,1)
no.trials= 10000 len(x)=10, len(y)=7
488
514
494
510
487
545
512
492
483
507
534
492
501
457
474
481
507
497
499
526
Frequency
0
100
200
300
400
500
0.0
0.2
0.4
0.6
0.8
1.0
p-value

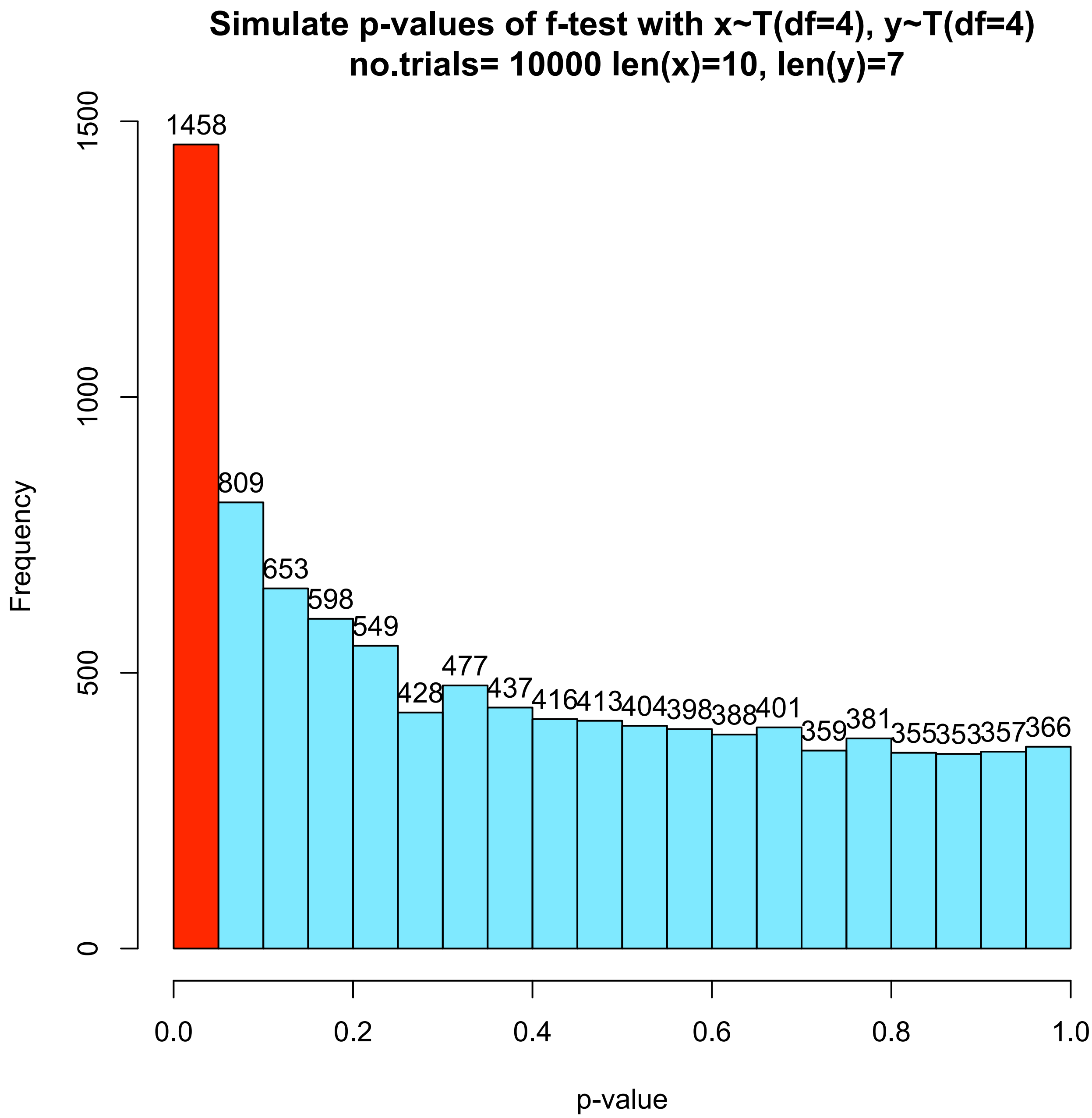

Simulate p-values of f-test with x~Exp(1), y~N(0,1)
no.trials= 10000 len(x)=10, len(y)=7
Frequency
p-value
0
200
400
600
800
1000
1200
1400
0.0
0.2
0.4
0.6
0.8
1.0
1380
815
652
568
508
485
473
449
415
423
382
385
440
371
370
365
387
393
354
385

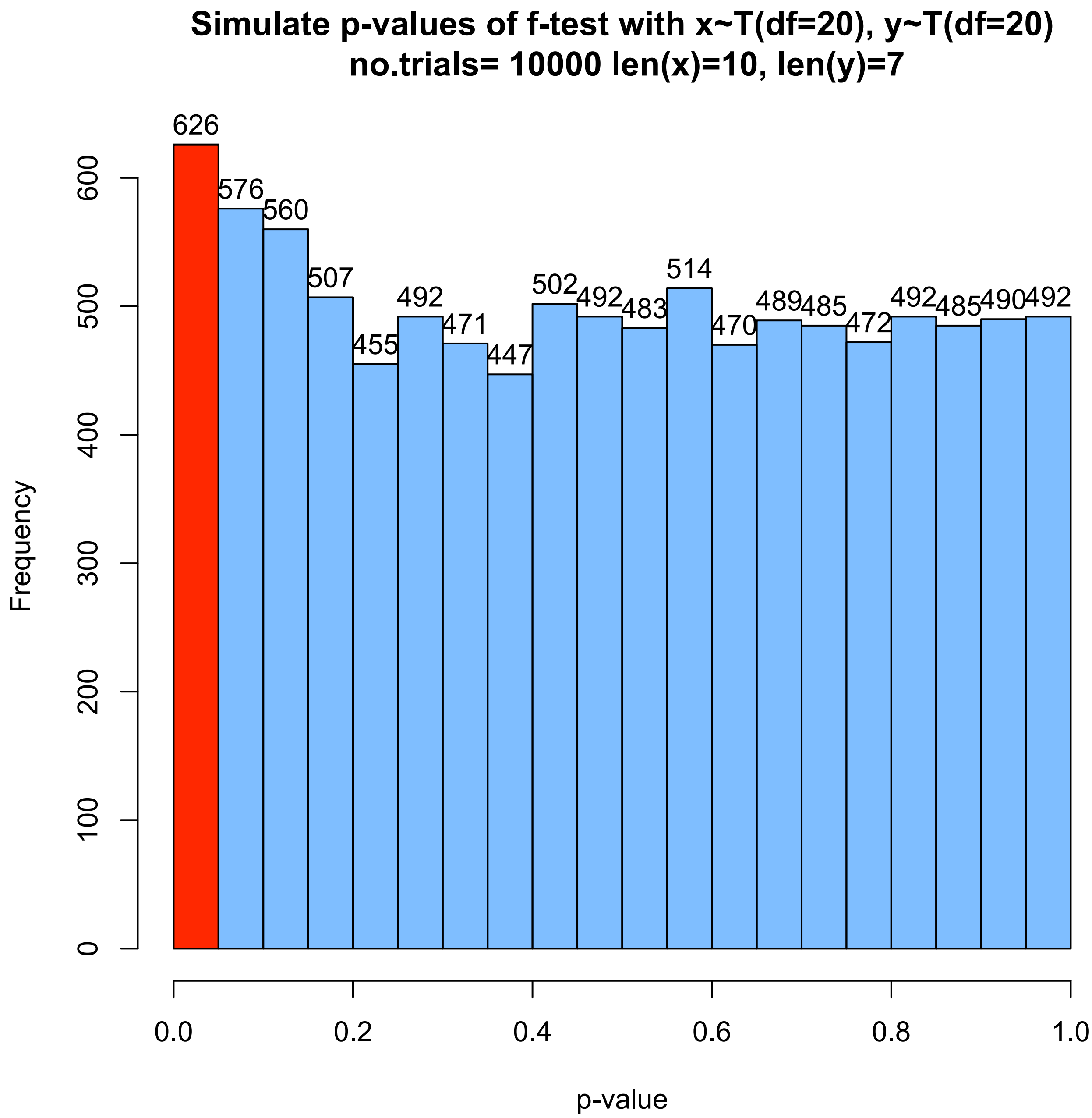

Simulate p-values of f-test with x~T(0,1,df=4), y~N(0,1)
no.trials= 10000 len(x)=10, len(y)=7
Frequency
p-value
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0
1006
696
610
553
534
458
445
470
496
482
424
459
446
437
389
429
426
444
405
391

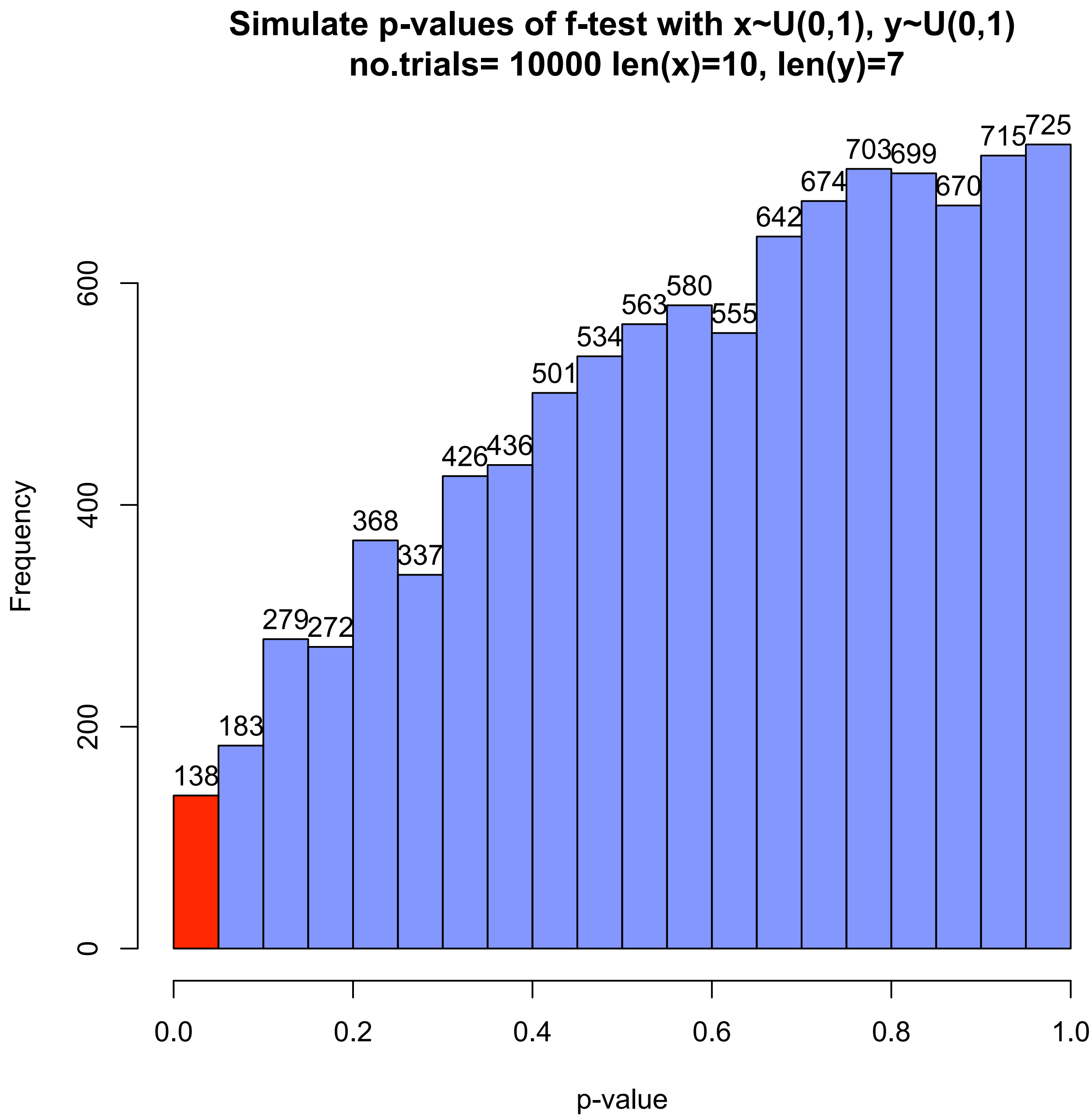

Simulate p-values of f-test with x~T(0,1,df=20), y~N(0,1)
no.trials= 10000 len(x)=10, len(y)=7
537
530
507
506
478
530
478
514
488
474
483
505
522
460
544
457
471
534
490
492
Frequency
0
100
200
300
400
500
0.0
0.2
0.4
0.6
0.8
1.0
p-value

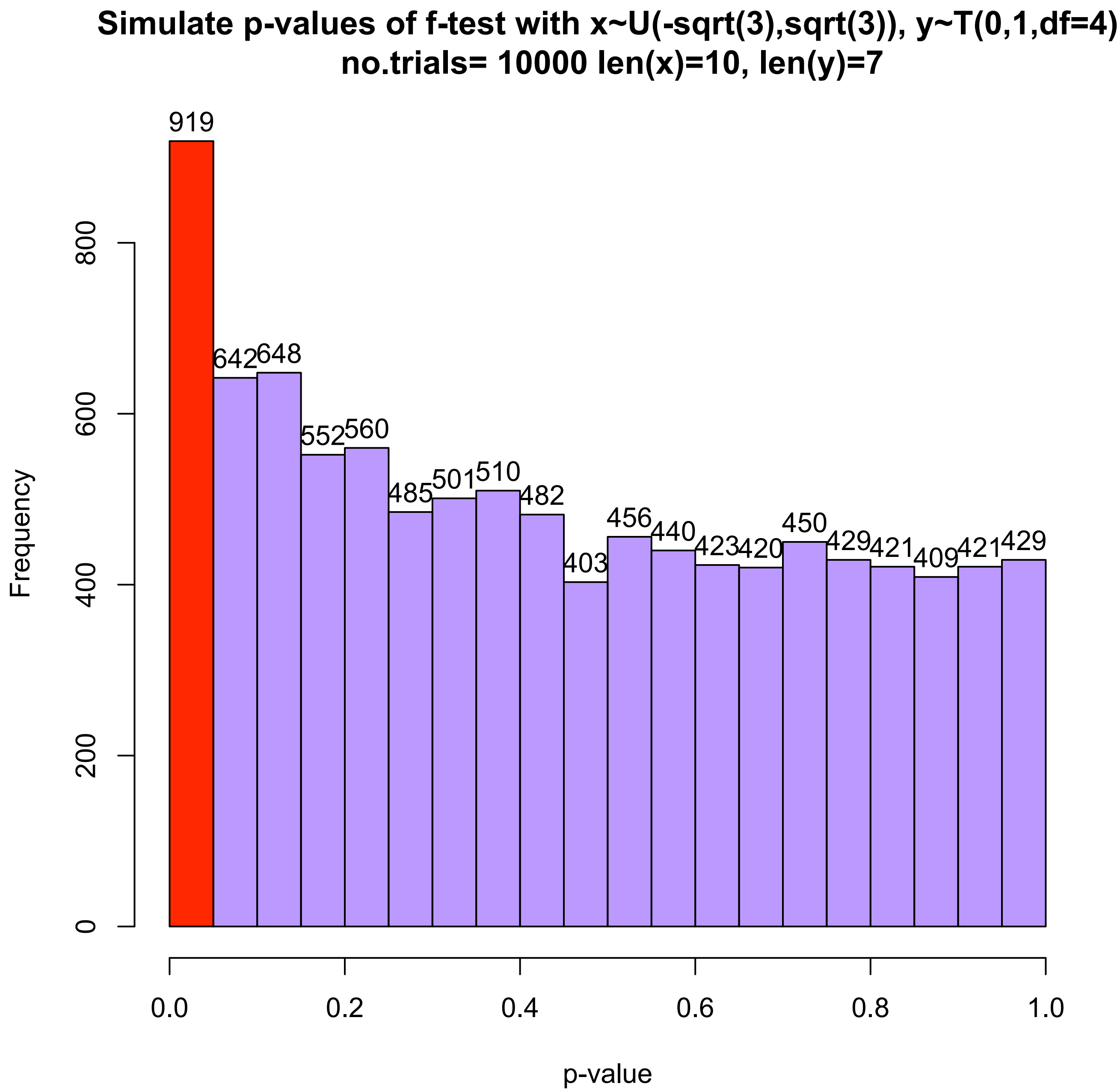

Simulate p-values of f-test with x~U(-sqrt(3),sqrt(3)), y~N(0,1)
no.trials= 10000 len(x)=10, len(y)=7
Frequency
p-value
0
100
200
300
400
500
0.0
0.2
0.4
0.6
0.8
1.0
305
359
388
448
450
551
490
540
500
518
504
532
552
562
529
568
541
547
555
561

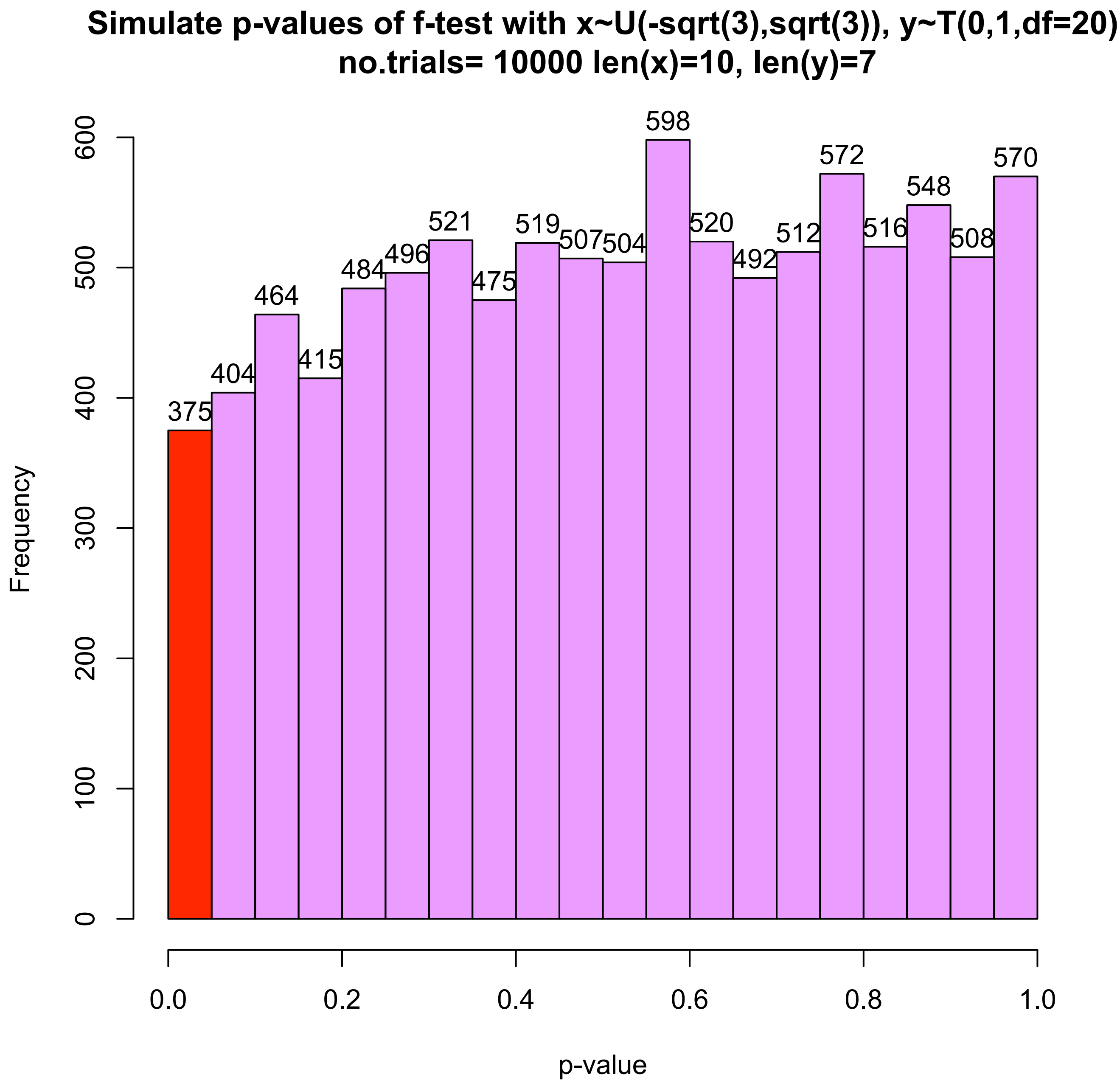

Simulate p-values of f-test with x~Exp(1), y~Exp(1)
no.trials= 10000 len(x)=10, len(y)=7
Frequency
p-value
0
500
1000
1500
2000
0.0
0.2
0.4
0.6
0.8
1.0
2144
855
703
562
477
469
423
409
383
347
357
363
311
303
326
350
307
290
285
336


Simulate p-values of f-test with x~T(df=4), y~T(df=4)
no.trials= 10000 len(x)=10, len(y)=7
Frequency
p-value
0
500
1000
1500
0.0
0.2
0.4
0.6
0.8
1.0
1458
809
653
598
549
428
477
437
416
413
404
398
388
401
359
381
355
353
357
366


Simulate p-values of f-test with x~T(df=20), y~T(df=20)
no.trials= 10000 len(x)=10, len(y)=7
Frequency
p-value
0
100
200
300
400
500
600
0.0
0.2
0.4
0.6
0.8
1.0
626
576
560
507
455
492
471
447
502
492
483
514
470
489
485
472
492
485
490
492


Simulate p-values of f-test with x~U(0,1), y~U(0,1)
no.trials= 10000 len(x)=10, len(y)=7
Frequency
0
200
400
600
138
183
279
272
368
337
426
436
501
534
563
580
555
642
674
703
699
670
715
725
0.0
0.2
0.4
0.6
0.8
1.0
p-value


Simulate p-values of f-test with x~U(-sqrt(3),sqrt(3)), y~T(0,1,df=4)
no.trials= 10000 len(x)=10, len(y)=7
919
642
648
552
560
485
501
510
482
403
456
440
423
420
450
429
421
409
421
429
0
200
400
600
800
Frequency
0.0
0.2
0.4
0.6
0.8
1.0
p-value


Simulate p-values of f-test with x~U(-sqrt(3),sqrt(3)), y~T(0,1,df=20)
no.trials= 10000 len(x)=10, len(y)=7
Frequency
p-value
0
100
200
300
400
500
600
0.0
0.2
0.4
0.6
0.8
1.0
375
404
464
415
484
496
521
475
519
507
504
598
520
492
512
572
516
548
508
570