Population and Sampling Distributions

Yalin Yang

2020-04-23

Contents

Population and Sampling Distributions

Demonstrates numerical integration to get

- the distribution function from the density function
- calculate the expectation
- calculate the variance

Example: **The exponential distribution** with x >= 0 and parameter λ Density <- $\lambda * exp(-\lambda * x)$ Distribution <- $1 - exp(-\lambda * x)$ Expectation <- $1/\lambda$ Variance <- $1/\lambda^2$

Estimator lambda <- $\bar{x}$

Riemann summands

- Plots the Rieman summands into an existing plot
- Calculate the midpoint Riemann sum
- start value <- a
- end value <- b
- number of summands <- n

```
IntBoxes <- function(IntFunc,a,b,n,plotIt=TRUE){
  intgrnd <- match.fun(IntFunc)
  integrand <- function(x) intgrnd(x)
  xleft <- seq(a,b-((b-a)/n),by=(b-a)/n)
```

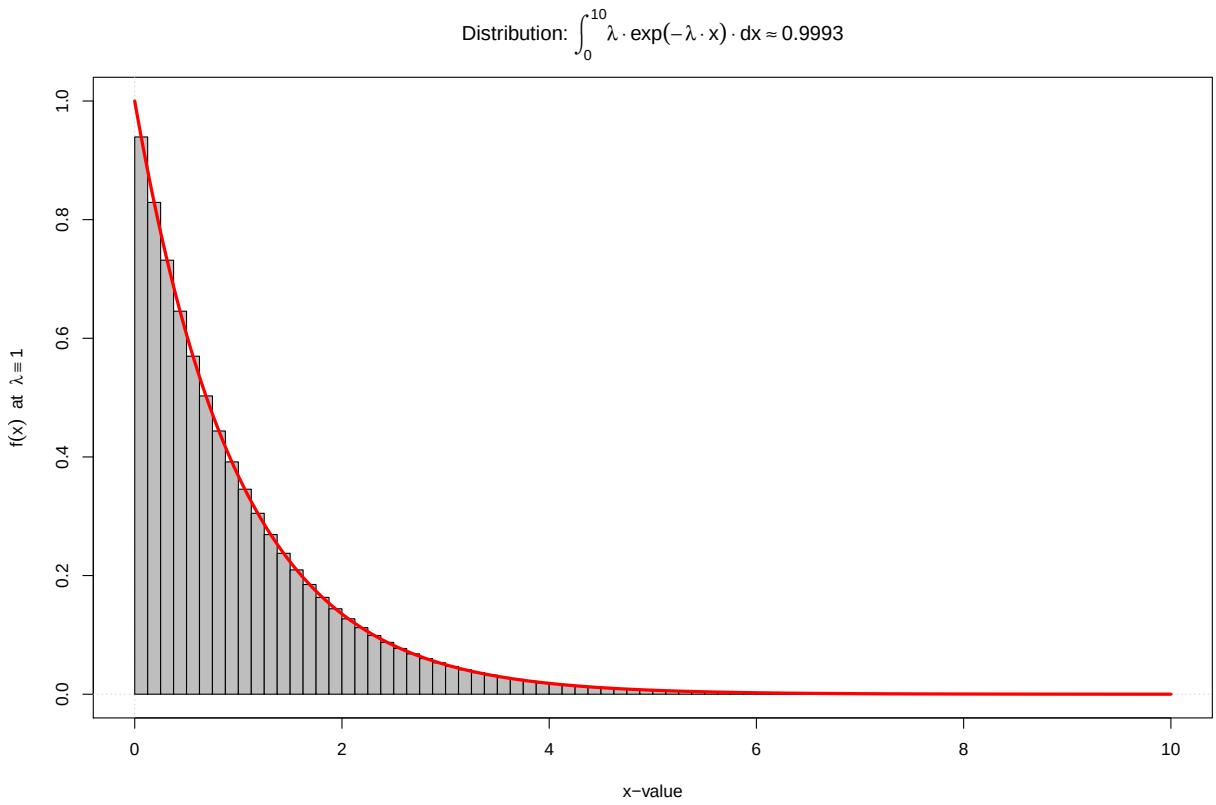
```
  xright <- seq(a+((b-a)/n),b,by=(b-a)/n)
  ybottom <- rep(0,n-1)
  ytop <- integrand(seq(a+((b-a)/(2*n)),b-((b-a)/(2*n)),by=(b-a)/n))
  if (plotIt) rect(xleft,ybottom,xright,ytop,col="grey")      # plot summands
  RieSum <- (b-a)/n*sum(ytop)
  return(RieSum)
} #end::IntBox
```

Simulate density function, expectation, and variance.

```
## Parameters
  nBoxes <- 80       # Number of Summands for the Rieman sum
  lambda <- 1        # Define lambda as a global variable
  xMin <- 0          # Lower integration bound
  xMax <- 10         # Upper integraion bound. Set xMax larger for smaller lambdas
  x <- seq(xMin,xMax,length.out=500) # Sequence of x values for plot

  ## Define function to be evaluated
  ExpDens <- function(x) {                      # density
    ifelse(x >= 0,lambda*exp(-lambda*x),0)
  }
  ExpDensExpect <- function(x) {                # expected value
    ifelse(x >= 0,x * ExpDens(x),0)
  }

  ExpDensVar <- function(x) {                    # variance
    ifelse(x >= 0,(x-1/lambda)^2 * ExpDens(x),0)
  }
```

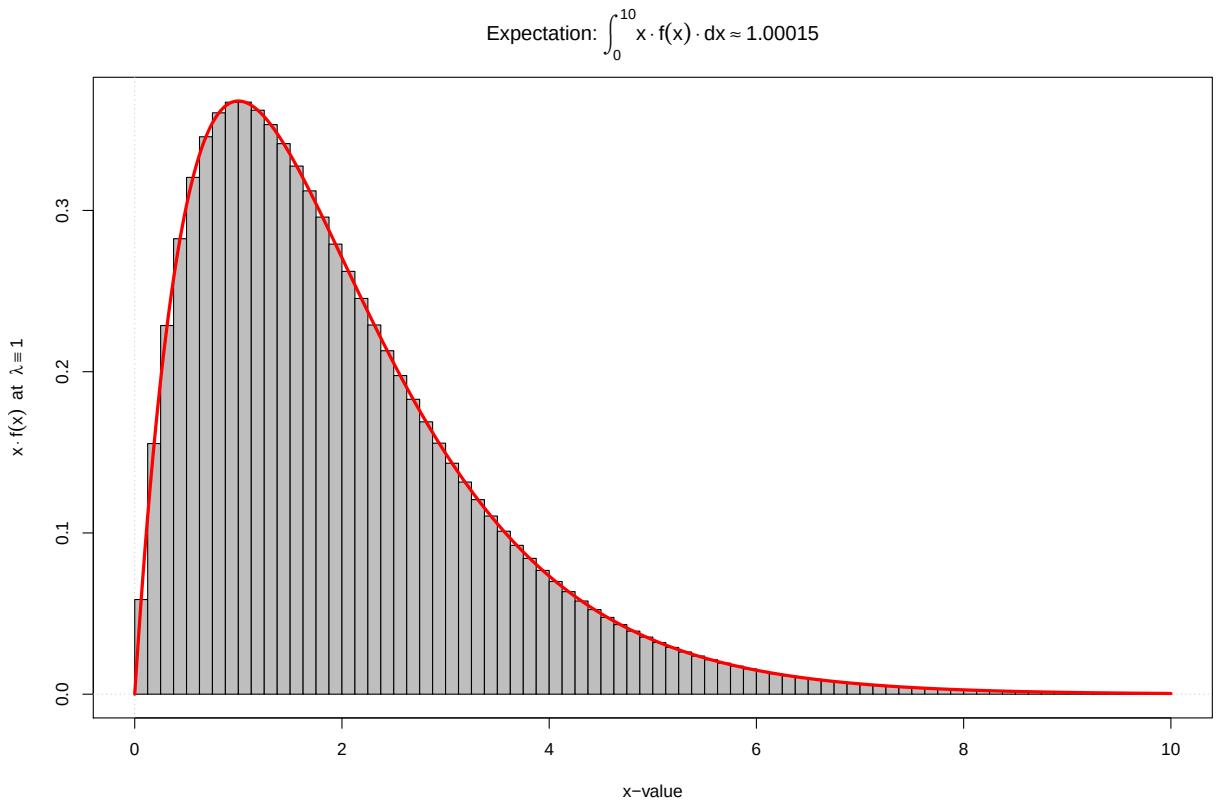
Plot distribution function with a specific range

```
xCut <- xMax
plot(x,ExpDens(x),type="n",xlab="x-value",
     ylab=bquote(paste(f(x),"  at  ", lambda %==% .(lambda))))
abline(v=0,lty="dotted",col="grey"); abline(h=0,lty="dotted",col="grey")
ExpDistrib <- IntBoxes(ExpDens,xMin, xCut, nBoxes)
lines(x,ExpDens(x),type="l",col="red",lwd=3)
title(main=bquote(paste("Distribution: ",integral(lambda%.%exp(-lambda%.%x)%.%dx, .(xMin), .(xMax)) %~~%
```



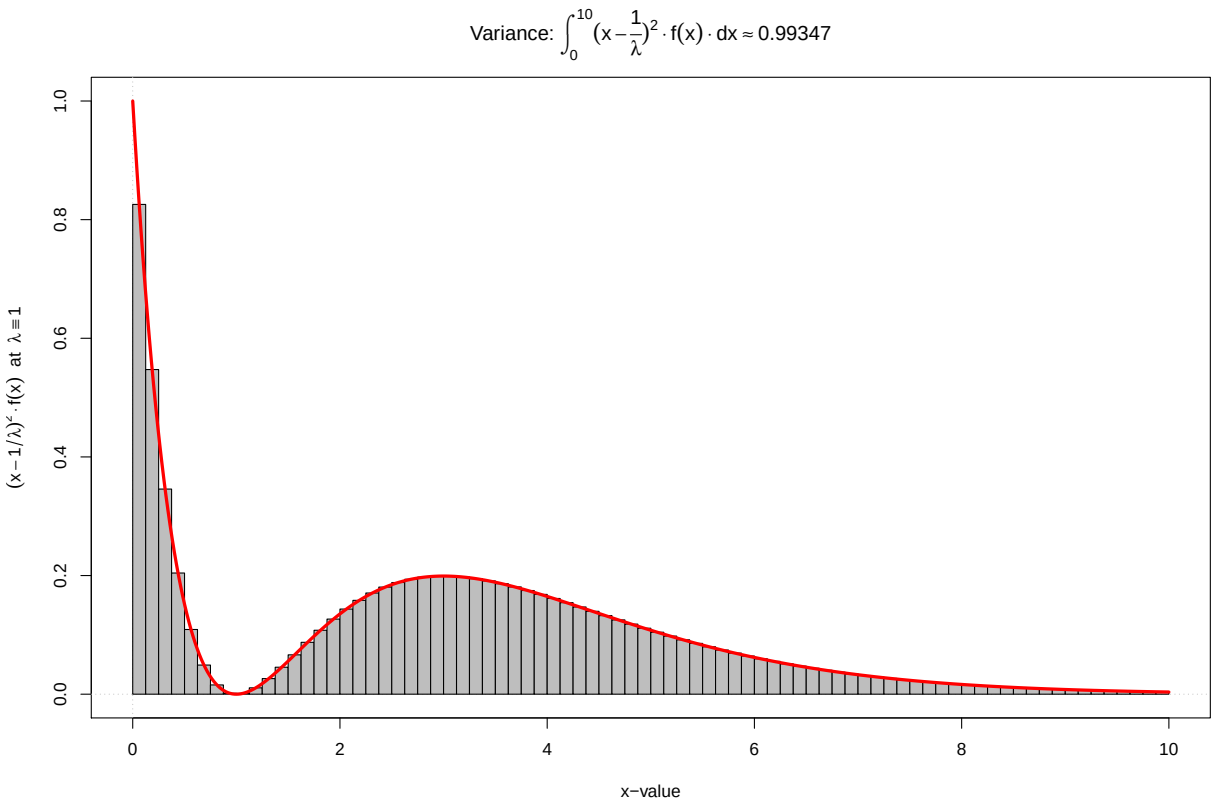
Expected value

```
plot(x,ExpDensExpect(x),type="n", xlab="x-value",
ylab=bquote(paste(x%.%f(x),"  at  ", lambda %==% .(lambda))))
abline(v=0,lty="dotted",col="grey"); abline(h=0,lty="dotted",col="grey")
EstExpect <- IntBoxes(ExpDensExpect,xMin,xMax,nBoxes)
lines(x,ExpDensExpect(x),type="l",col="red",lwd=3)
title(main=bquote(paste("Expectation: ",integral(x%.%f(x)%.%dx, .(xMin), .(xMax)) %~~% .(round(EstExpec
```



Variance

```
plot(x,ExpDensVar(x),type="n",xlab="x-value",
     ylab=bquote(paste((x-1/lambda)^2%.%f(x),"  at  ", lambda %==% .(lambda))))
abline(v=0,lty="dotted",col="grey"); abline(h=0,lty="dotted",col="grey")
EstVar <- IntBoxes(ExpDensVar,xMin,xMax,nBoxes)
lines(x,ExpDensVar(x),type="l",col="red",lwd=3)
title(main=bquote(paste("Variance: ",integral((x-over(1,lambda))^2%.%f(x)%.%dx, .(xMin), .(xMax)) %~~%
```



Distributions

Simulate and plot three main distributions

Initialize variables

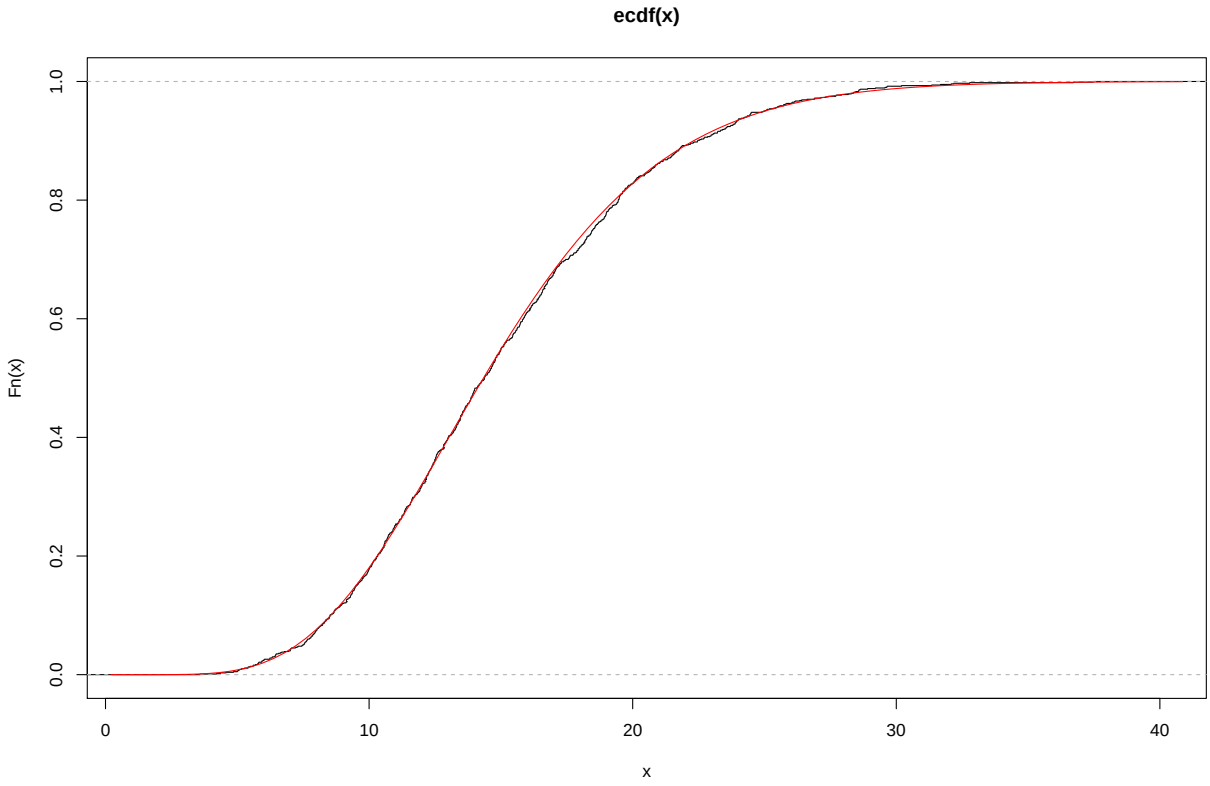
```
xr <- function(x){
  seq(from=min(x)-((max(x)-min(x))/10),
      to=max(x)+((max(x)-min(x))/10),
      length.out=n/10)
}
## Set degrees of freedom for numerator (df1) and denominator (df2)
df1 <- 15
df2 <- 20

## Set number of simulated random variables
n <- 1000
```

chi^2-distribution

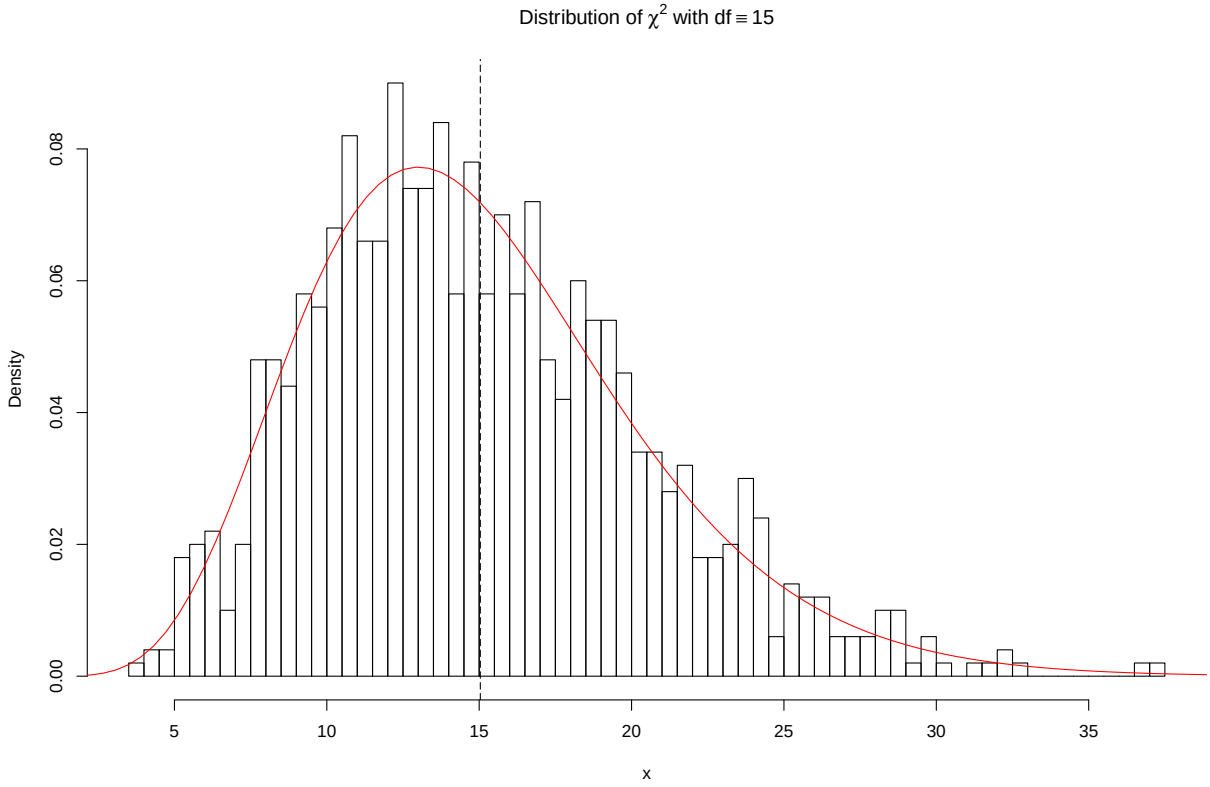
```
x <- rep(NA, n)             # Initialize vector of random variables
for (i in 1:n){
  x[i] <- sum(rnorm(df1)^2)
}
ks.test(x,"pchisq", df1, alternative="two.sided")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.018696, p-value = 0.8757
## alternative hypothesis: two-sided
```

```
plot(ecdf(x)); lines(xr(x), pchisq(xr(x), df1), col="red")
```




```
hist(x, breaks= n/20, freq=FALSE,
     main=bquote(paste("Distribution of ", chi^2," with ", df%==% .(df1))))
lines(xr(x), dchisq(xr(x), df1), col="red")
abline(v=mean(x), lty=5)
```

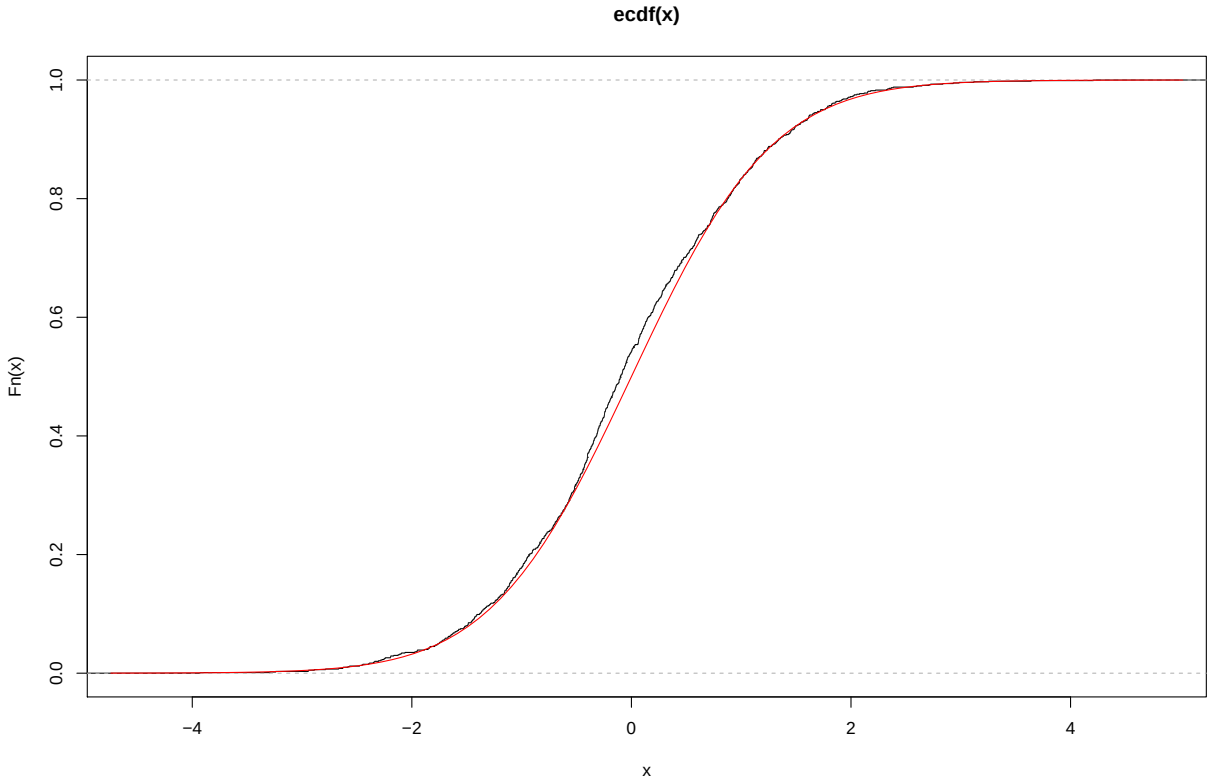


t-distribution

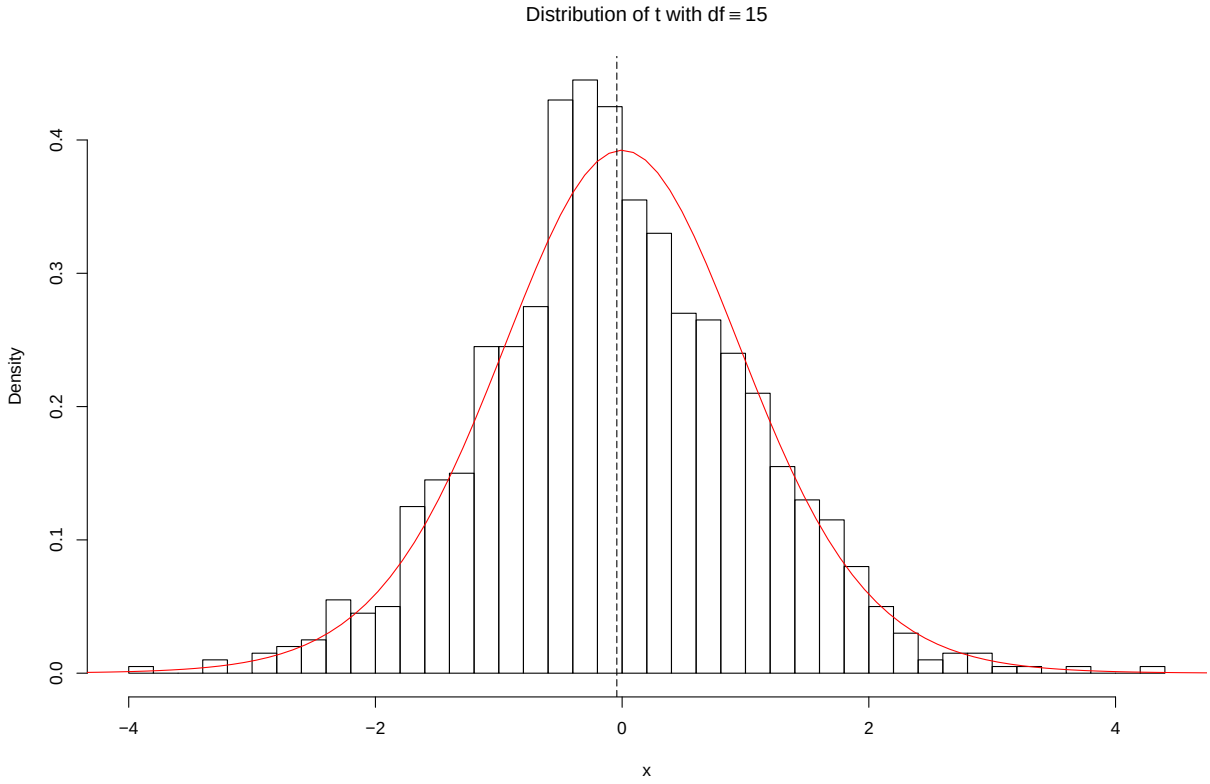
```
x <- rep(NA, n)             # Initialize vector of random variables
for (i in 1:n){
  x[i] <- rnorm(1)/sqrt(sum(rnorm(df1)^2)/df1)
}
ks.test(x,"pt", df1, alternative="two.sided")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.046311, p-value = 0.02743
## alternative hypothesis: two-sided
```

```
plot(ecdf(x)); lines(xr(x), pt(xr(x), df1), col="red")
```




```
hist(x, breaks= n/20, freq=FALSE,
     main=bquote(paste("Distribution of t with ", df%==% .(df1))))
lines(xr(x), dt(xr(x), df1), col="red")
abline(v=mean(x), lty=5)
```

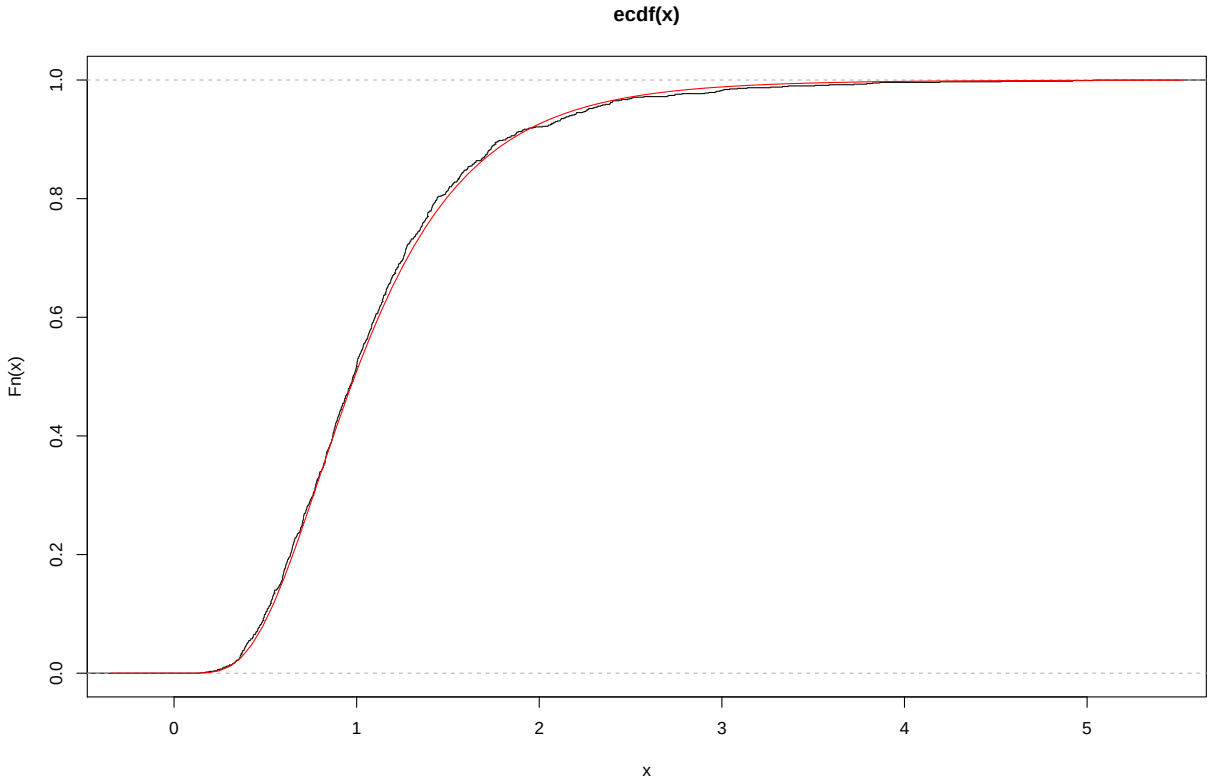


F-distribution

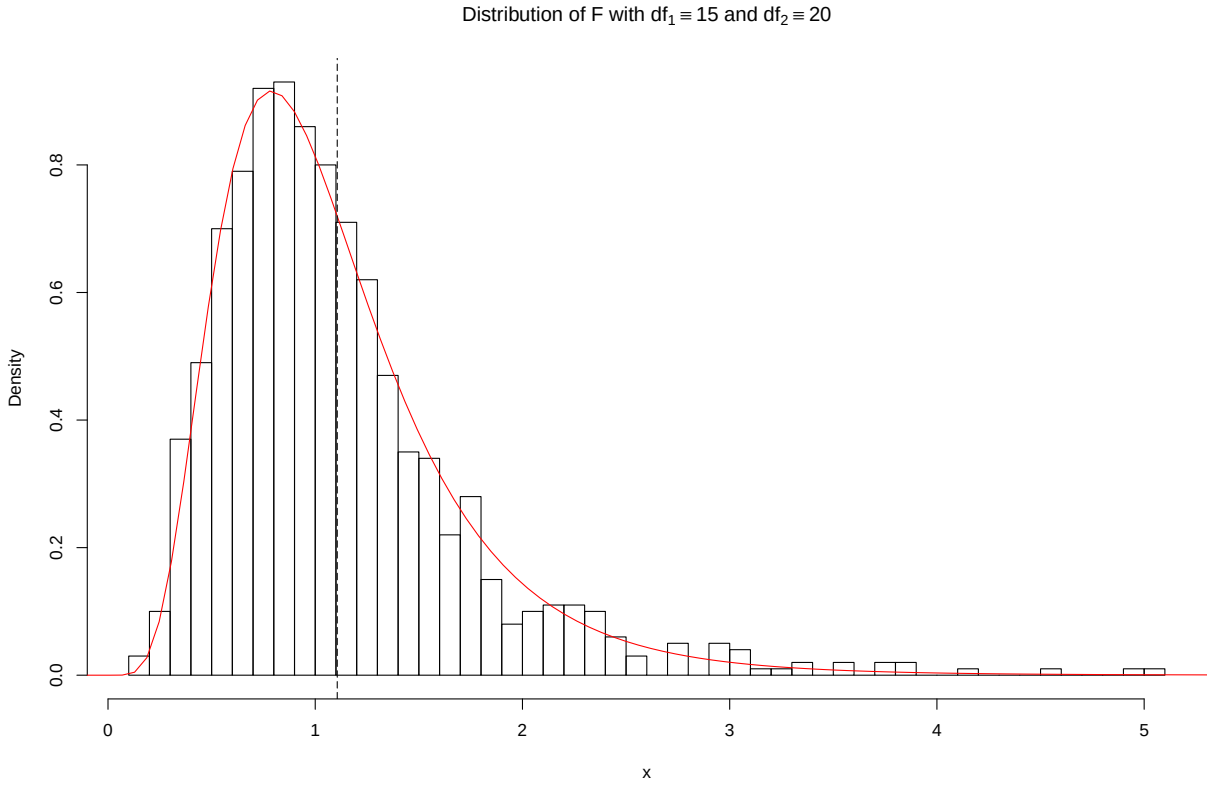
```
x <- rep(NA, n)             # Initialize vector of random variables
for (i in 1:n){
  x[i] <- (sum(rnorm(df1)^2)/df1)/(sum(rnorm(df2)^2)/df2)
}
ks.test(x,"pf", df1, df2, alternative="two.sided")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.022221, p-value = 0.7068
## alternative hypothesis: two-sided
```

```
plot(ecdf(x)); lines(xr(x), pf(xr(x), df1, df2), col="red")
```




```
hist(x, breaks= n/20, freq=FALSE,
     main=bquote(paste("Distribution of F with ", df[1]%==% .(df1)," and ", df[2]%==% .(df2))))
lines(xr(x), df(xr(x), df1, df2), col="red")
abline(v=mean(x), lty=5)
```


Distribution of F with $df_1 \equiv 15$ and $df_2 \equiv 20$
Density
0.0
0.2
0.4
0.6
0.8
0
1
2
3
4
5
x