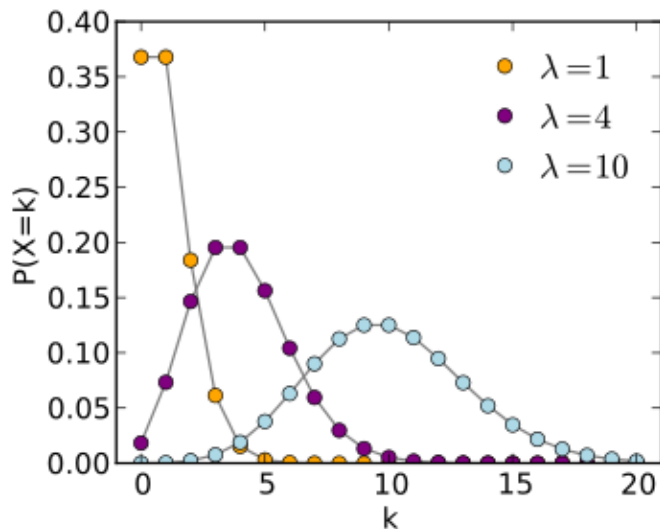


R language and data analysis: Distribution

Qiang Shen

Jan.2, 2018

Distribution



Distribution

Distribution	Abbreviation	Distribution	Abbreviation
Beta	beta	Logistic	logis
Binomial	binom	Multinomial	multinom
Cauchy	cauchy	Negative binomial	nbinom
Chi-squared (noncentral)	chisq	Normal	norm
Exponential	exp	Poisson	pois
F	f	Wilcoxon Signed Rank	signrank
Gamma	gamma	T	t
Geometric	geom	Uniform	unif
Hypergeometric	hyper	Weibull	weibull
Lognormal	lnorm	Wilcoxon Rank Sum	wilcox

Four function for probability distribution

- d = density function
- p = distribution function
- q = quantile function
- r = random function

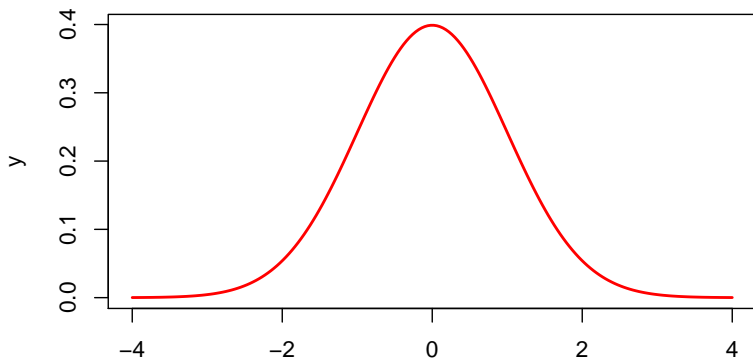
Normal distribution: pdf

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$$

Standard normal distribution: equation

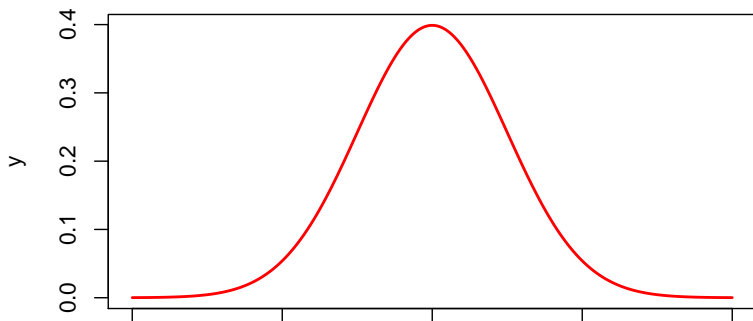
```
x=seq(-4,4,length=200)
y=1/sqrt(2*pi)*exp(-x^2/2)
plot(x,y,type="l",lwd=2,col="red")
```



Standard normal distribution:dnorm function

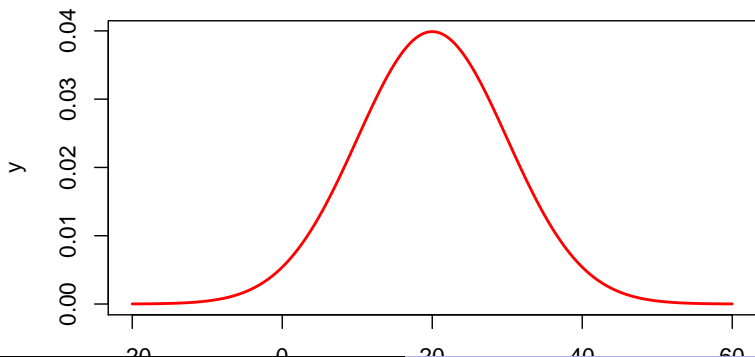
- dnorm

```
x=seq(-4,4,length=200)  
y=dnorm(x,mean=0,sd=1)  
plot(x,y,type="l",lwd=2,col="red")
```



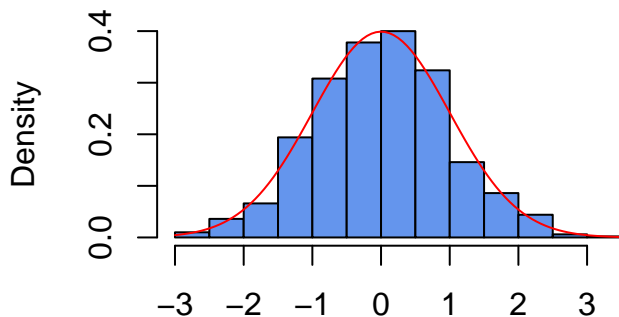
Normal distribution:dnorm function

```
mean=20;sd=10  
x=seq(mean-4*sd,mean+4*sd,length=200)  
y=dnorm(x,mean,sd)  
plot(x,y,type="l",lwd=2,col="red")
```



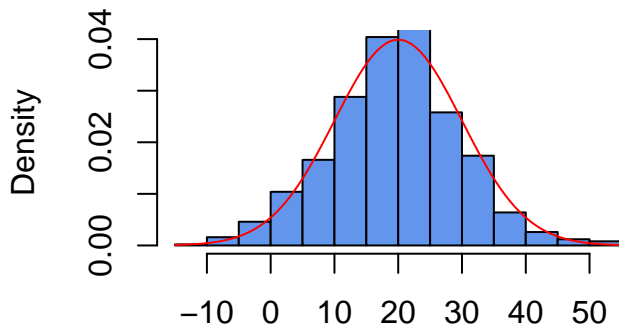
rnorm

```
x=seq(-4,4,length=200)
hist(rnorm(1000),freq=F,col="cornflowerblue",ylim=c(0,0.4),
curve(dnorm(x), add=TRUE, col="red"))
```



rmnorm

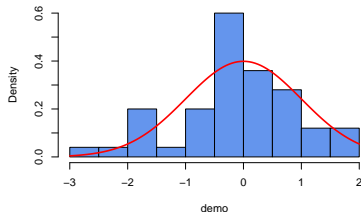
```
x=seq(-4,4,length=200)  
hist(rnorm(1000,mean=20,sd=10),freq=F,col="cornflowerblue",  
curve(dnorm(x,20,10), add=TRUE, col="red"))
```



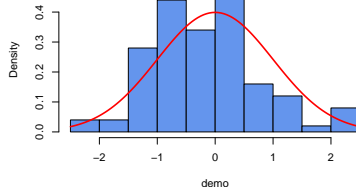
normal distribution

- sample size

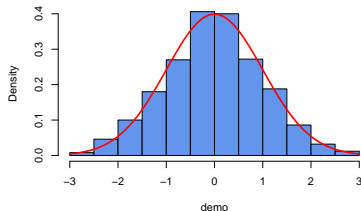
histogram with 50 sample



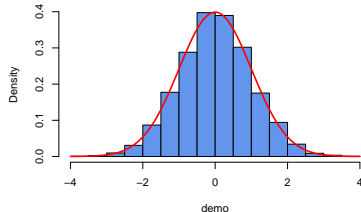
histogram with 100 sample



histogram with 1000 sample



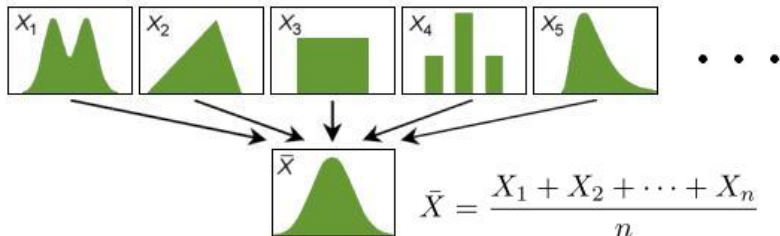
histogram with 10000 sample



normal distribution

```
par(mfrow=c(2,2))
func<-function(x){
  demo<-rnorm(x,mean=0,sd=1)
  hist(demo, freq=FALSE,breaks=70,col="cornflowerblue",
       main=paste("histogram with",x,'sample'))
  curve(dnorm(x), add=TRUE, col="red", lwd=2)
}
sapply(c(50,100,1000,10000),func)
```

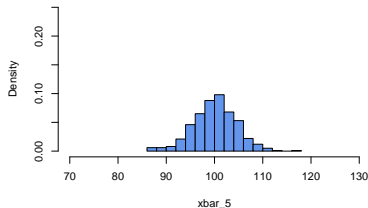
central limit theorem (CLT)



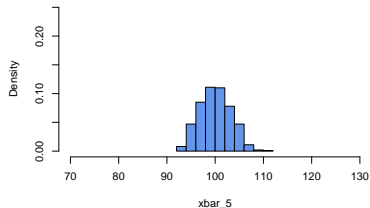
Sample from normal distribution

500 times sampling from a normal population with mean 100 and sd 20.

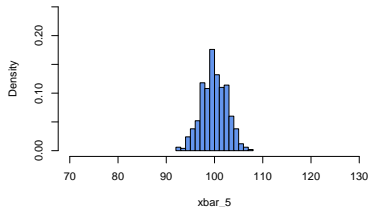
sample size: 5



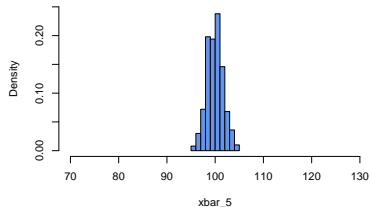
sample size: 10



sample size: 15



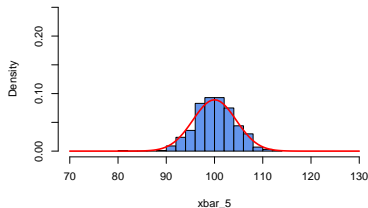
sample size: 30



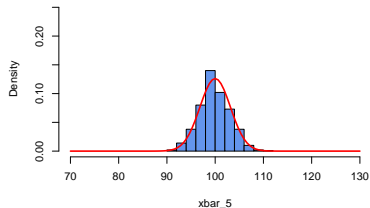
Sample from normal distribution

500 times sampling from a normal population with mean 100 and sd 20.

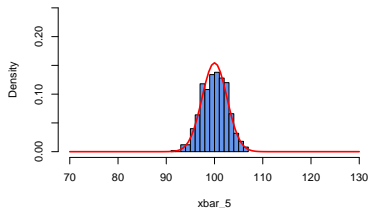
sample size: 5



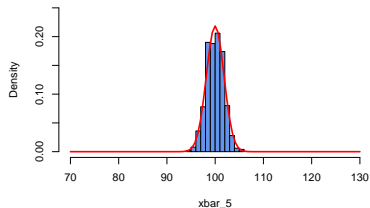
sample size: 10



sample size: 15



sample size: 30



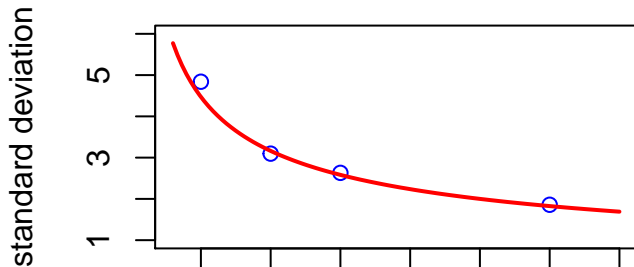
Sample from normal distribution

500 times sampling from a population with mean 100 and sd 20.

```
par(mfrow=c(2,2))
mu=100; sigma=10
clt<-function(n){
  xbar=rep(NA,500)
  for (i in 1:500) {
    xbar[i]=mean(rnorm(n,mean=mu,sd=sigma))
  }
  hist(xbar,prob=TRUE,breaks=12,xlim=c(70,130),
       ylim=c(0,0.25),col='cornflowerblue',main=paste(
         "sample size:",n))
  curve(dnorm(x,mu,sigma/sqrt(n)), add=TRUE,
        col="red", lwd=2)
  return(xbar)
}
result<-sapply(c(5,10,15,30),clt)
```


standard error

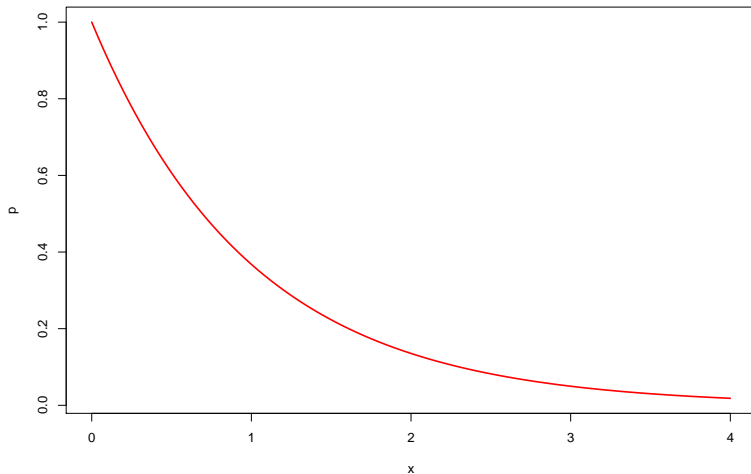
```
par(mfrow=c(1,1));results<-read.csv('result.csv');ssizes=c  
plot(ssizes,apply(result,2,sd),col='blue',  
xlab="sample size",ylab="standard deviation",  
xlim=c(3,35),ylim=c(1,6));x=seq(2,32,length=200)  
curve(10/sqrt(x),add=TRUE,type="l",lwd=2,col="red")
```



exponential distribution

$$f(x) = \lambda e^{-\lambda x}$$

exponential distribution

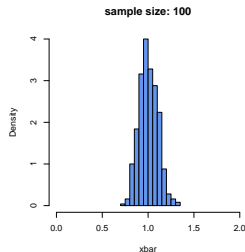
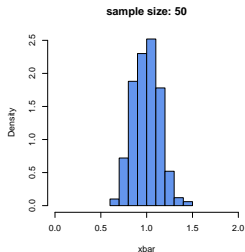
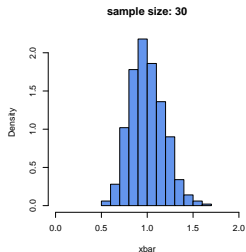
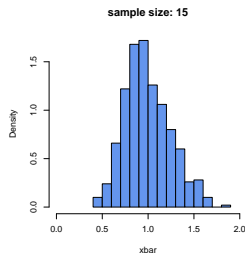
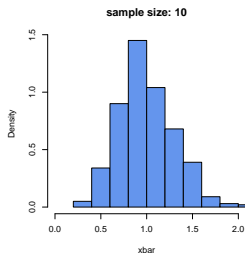
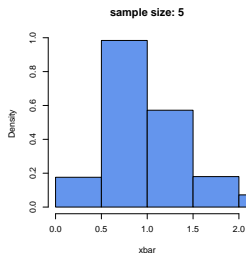


exponential distribution

```
par(mfrow=c(1,1))  
curve(dexp(x,rate=1),0,4,lwd=2,col="red",ylab="p")
```

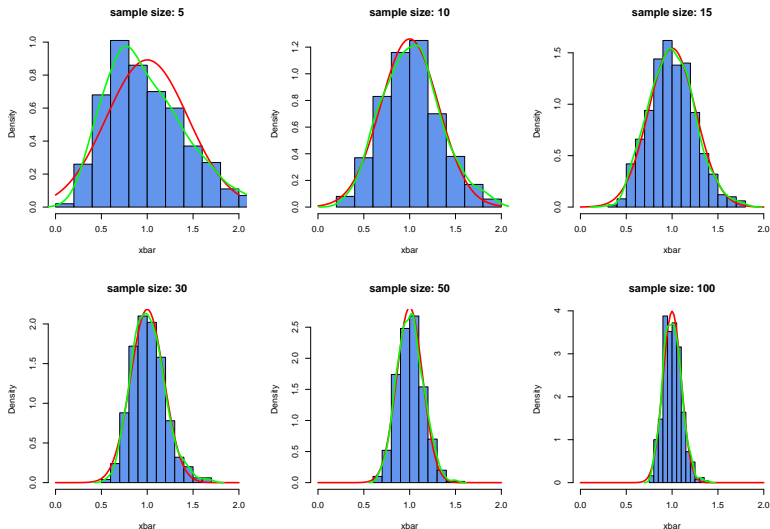
expotential distribution sampling

500 times sampling from expotential distribution with $\lambda = 1$



exponential distribution sampling

500 times sampling from exponential distribution with $\lambda = 1$

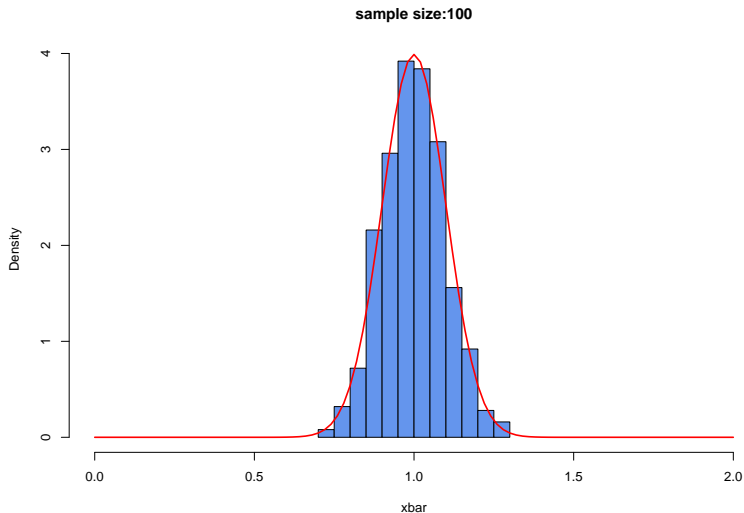


expotential distribution sampling

500 times sampling from expotential distribution with $\lambda = 1$

```
par(mfrow=c(2,3))
clt_e<-function(n,lambda=1){
  xbar=rep(0,500);rate=1
  for (i in 1:500) {
    xbar[i]=mean(rexp(n,rate=1))
  }
  hist(xbar,prob=TRUE,breaks=12,xlim=c(0,2),
       main=paste("sample size:",n),col='cornflowerblue')
  curve(dnorm(x,rate,rate/sqrt(n)), add=TRUE,
        col="red", lwd=2)
  return(xbar)
}
result<-sapply(c(5,10,15,30,50,100),clt_e)
```

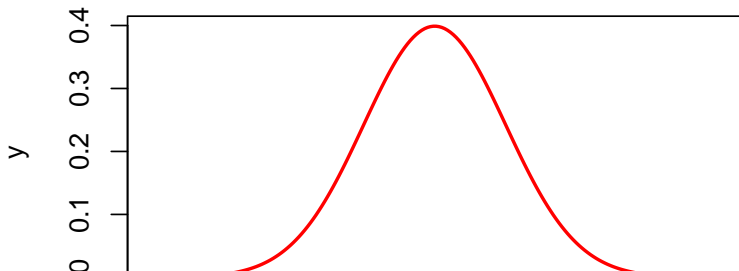
sample size = 100



probability density function

- dnorm

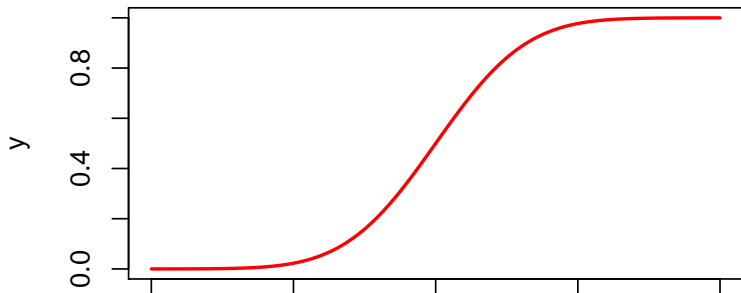
```
# with dnorm  
x=seq(-4,4,length=200)  
y=dnorm(x,mean=0,sd=1)  
plot(x,y,type="l",lwd=2,col="red")
```



cumulative density function

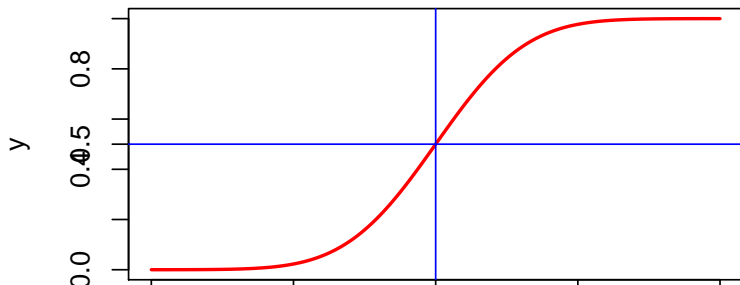
- pnorm

```
x=seq(-4,4,length=200)
y=pnorm(x,mean=0,sd=1)
plot(x,y,type="l",lwd=2,col="red")
```



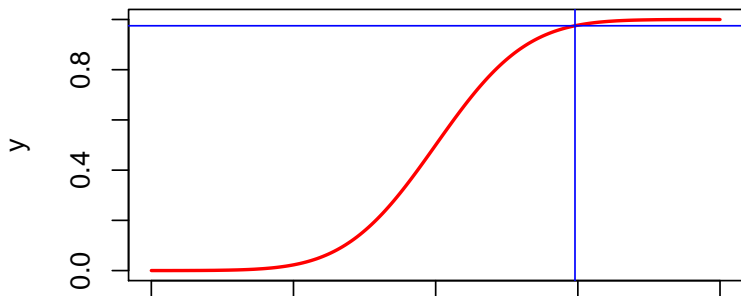
cdf:pnorm

```
x=seq(-4,4,length=200)
y=pnorm(x,mean=0,sd=1)
plot(x,y,type="l",lwd=2,col="red")
axis(2, at=0.5);abline(v=0, col="blue")
abline(h=0.5,col='blue')
```

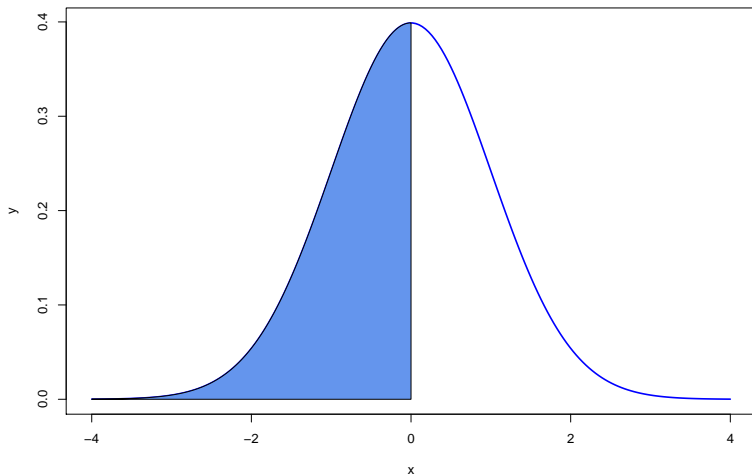


pnorm

```
x=seq(-4,4,length=200)
y=pnorm(x,mean=0,sd=1)
plot(x,y,type="l",lwd=2,col="red")
abline(v=1.96, col="blue");abline(h=0.975,col='blue')
```

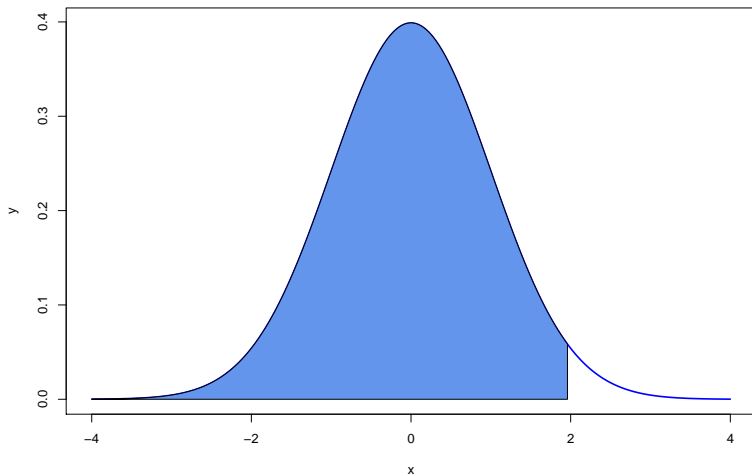


pdf vs. cdf

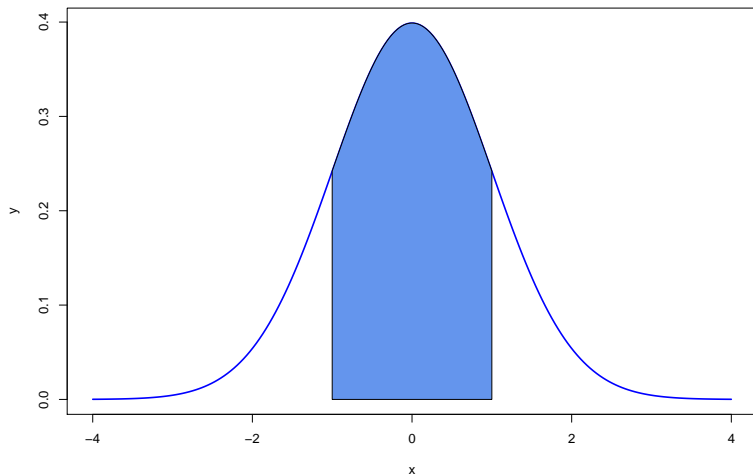


`pnorm(0)`

pnorm: 1.96



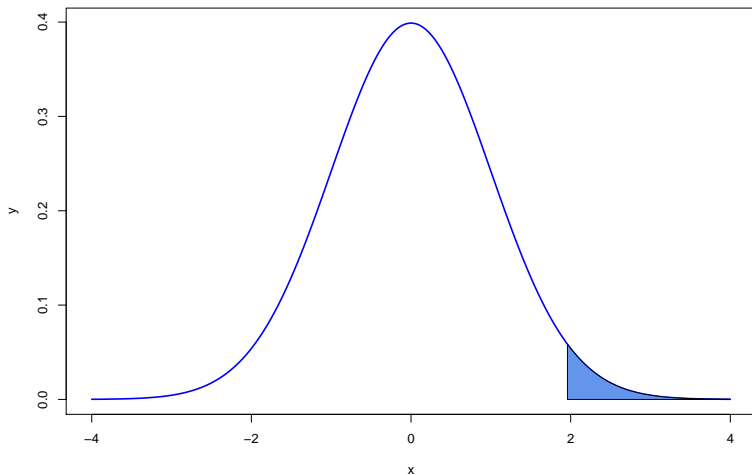
`pnorm(1.96)`



```
pnorm(1)-pnorm(-1)
```

```
[1] 0.6826895
```


dnorm



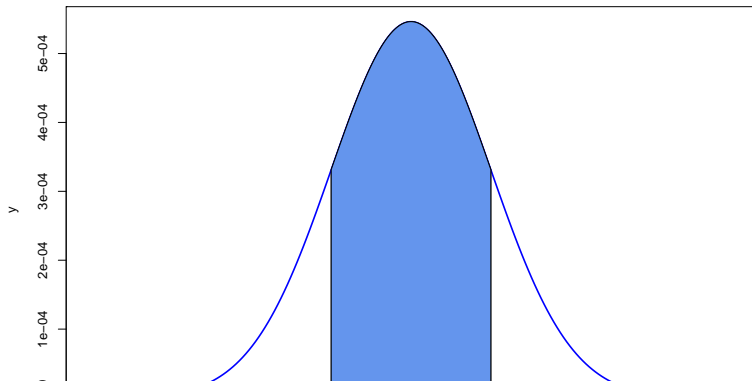
```
pnorm(-1.96)
```

```
[1] 0.0249979
```

```
1-pnorm(1.96)
```

```
[1] 0.0249979
```

example

$$\begin{aligned} \mu &= 51800 \\ sd &= \frac{4000}{\sqrt{30}} \\ \text{range} & 51300 \ 51800 \end{aligned}$$


example

```
mu=51800;variance<-4000;s=30  
sigma=variance/sqrt(s)# sd of sample mean  
z1<-(52300-mu)/sigma  
z2<-(51300-mu)/sigma  
pnorm(z1)-pnorm(z2)
```

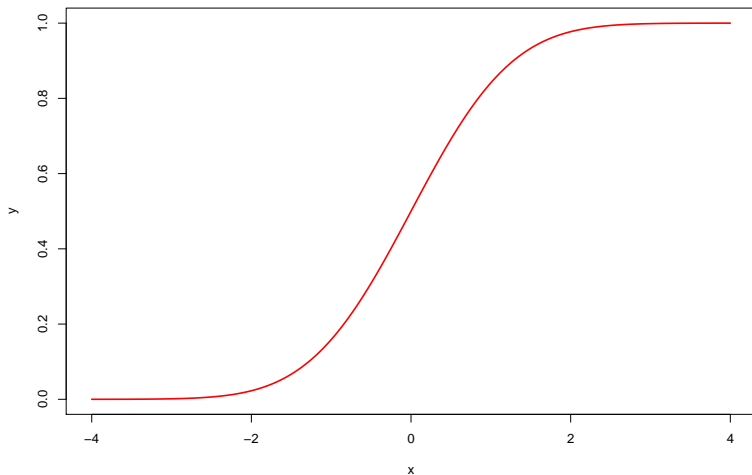
```
[1] 0.5064372
```

example:method 2

```
pnorm(mu+500,mu,sigma)-pnorm(mu-500,mu,sigma)
```

```
[1] 0.5064372
```

pnorm



qnorm

#95%

```
qnorm(0.975)
```

```
[1] 1.959964
```

#pvalue2sided

```
2*pnorm(-1.96)
```

```
[1] 0.04999579
```

#99%

```
qnorm(0.995)
```

```
[1] 2.575829
```

#90%

```
qnorm(0.95)
```

four function for probability distribution

- d = density function
- p = distribution function
- q = quantile function
- r = random function