#rwalk() generates 500 steps of a symmetric random walk and contrasts frequency non-
#convergence with proportion convergence.

```
> rwalk
function (n=500,p=.5)
{
	heads=rbinom(n-1,1,p)
	a=2*(heads)-1
	b=diffinv(a,xi=0)
	c=1:n
	pheads=(diffinv(heads,xi=.5))/c
	op=par(mfrow=c(2,1))
	plot(c,b,type="n",main="Law of Averages",xlab="Toss Number",ylab="Excess
of Heads",lw=2,cex.lab=1.5,cex.main=2)
	lines(c,b,col="red")
	lines(c,rep(0,n),col="black")
	plot(c,pheads,type="n",xlab="Toss Number",ylab="Proportion of
Heads",lw=2,cex.lab=1.5)
	lines(c,pheads,col="blue")
	lines(c,rep(.5,n),col="black")
	par(op)
	#Sys.sleep(1)
}
```

---

rw() uses a symmetric random walk to mimic the Toronto stock market index over one
#year.

```
> rw
function (n=250,p=.5,stepsize=1)
{
	heads=rbinom(n-1,1,p)
	a=(2*(heads)-1)*stepsize
	b=diffinv(a,xi=0)+100
	c=1:n
	plot(c,b,type="n",main="Symmetric Random Walk",xlab="Step
Number",ylab="Net Gain",lw=2,cex.lab=1.5,cex.main=2,ylim=c(80,150))
	lines(c,b,col="red")
	lines(c,rep(0,n)+100,col="black")
}
```

---

risky() takes a portfolio of risky companies and shows that, if the companies are
independent, they can produce a profit quite reliably.

```
> risky
function (m=1,n=25)
#n is the number of companies
#m is the number of simulations
```

```
#if m=1 just get one portfolio expce
{
	cum=1:m
	for (i in 1:m)
	{
	return=c(-1,-.5,0,3)
	a=sample(return,n,T,c(.25,.25,.25,.25))
	if (m ==1) {plot(table(a),xlab="net returns",ylab="frequency",col="purple")
		pause()
		quartz()
		plot(diffinv(a),type="l",xlab="company
ID",col="green",ylab="cumulative net return")
		text(.75*n,max(a)-1,"Profit=",cex=2)
		text(.75*n,max(a)-2,100*mean(a),cex=2)
		text(.85*n,max(a)-2,"%",cex=2)
		pause()
		my.dotplot(a,xlab="Net Return",col="blue")
		}
	cum[i]=mean(a)
	}
	if (m >1) {my.dotplot(cum,xlab="Average Profit per $1 Invested",col="blue")}
	pause()
	graphics.off()
}
```