Making plots in R [things I wish someone told me when I started grad school]

Kirk Lohmueller
Department of Ecology and Evolutionary Biology
UCLA

September 14, 2016

In honor of “Talk Like a Pirate Day...”

What’s a pirates favorite computer language?

In honor of “Talk Like a Pirate Day...”

Rrrrrr!

In honor of “Talk Like a Pirate Day...”

But, why?

In honor of “Talk Like a Pirate Day...”

Because they get lost when they go to C

Learning objectives

Learning objectives

- Know how to make simple plots:

Learning objectives

- Know how to make simple plots:
  - Scatterplot, histogram, density plot, bar plot

Learning objectives

- Know how to make simple plots:
  - Scatterplot, histogram, density plot, bar plot
- Read “large” datasets into R

Learning objectives

- Know how to make simple plots:
    - Scatterplot, histogram, density plot, bar plot
- Read “large” datasets into R
- Perform manipulations of large datasets

Learning objectives

- Know how to make simple plots:
  - Scatterplot, histogram, density plot, bar plot
- Read “large” datasets into R
- Perform manipulations of large datasets
- Calculate some statistics in R

Learning objectives

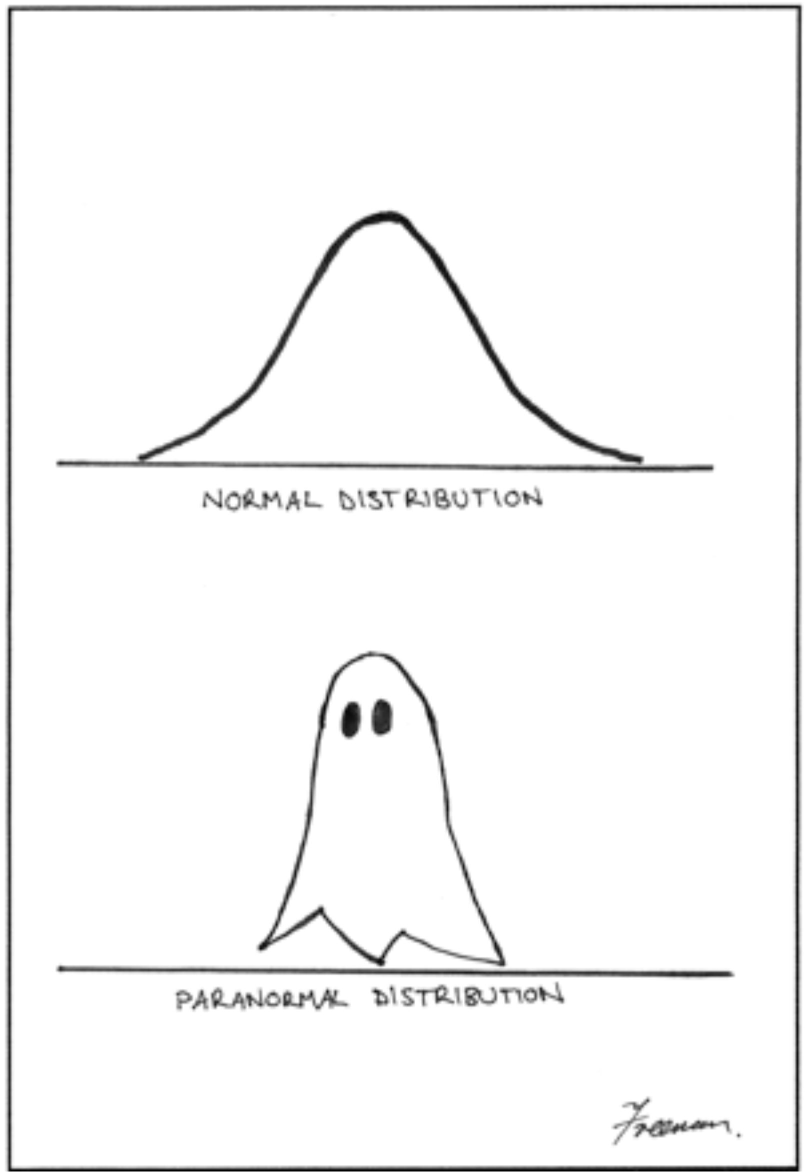
- Know how to make simple plots:
  - Scatterplot, histogram, density plot, bar plot
- Read “large” datasets into R
- Perform manipulations of large datasets
- Calculate some statistics in R
- Start to become familiar with simulation

Learning objectives

- Know how to make simple plots:
  - Scatterplot, histogram, density plot, bar plot
- Read “large” datasets into R
- Perform manipulations of large datasets
- Calculate some statistics in R
- Start to become familiar with simulation
- Begin to appreciate and enjoy Kirk’s corny humor

R can simulate data from a probability distribution

Let's look at the normal distribution:



Matthew Freeman *J Epidemiol Community Health* 2006;60:6

Simulating data from a normal distribution

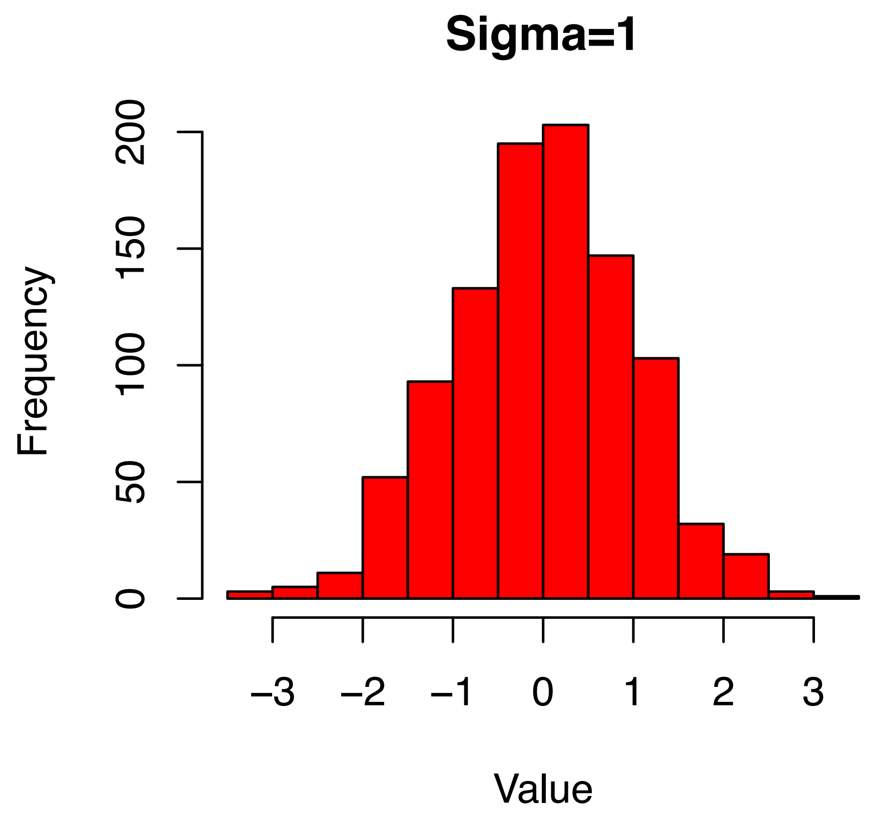
```
> #first, draw 1000 random values from a standard normal distribution (SD=1):
> s1<-rnorm(1000,mean=0,sd=1)
>
> #now do 1000 drawn from a normal distribution with SD=3.
> s3<-rnorm(1000,mean=0,sd=3)
>
> head(s1)
[1]  0.26951848 -2.43530911  1.15968499  0.09647798 -0.74425935  0.40504897
> head(s3)
[1]  3.6718664  4.8193934 -0.6078601  2.1520862  2.9089759 -3.6002362
```

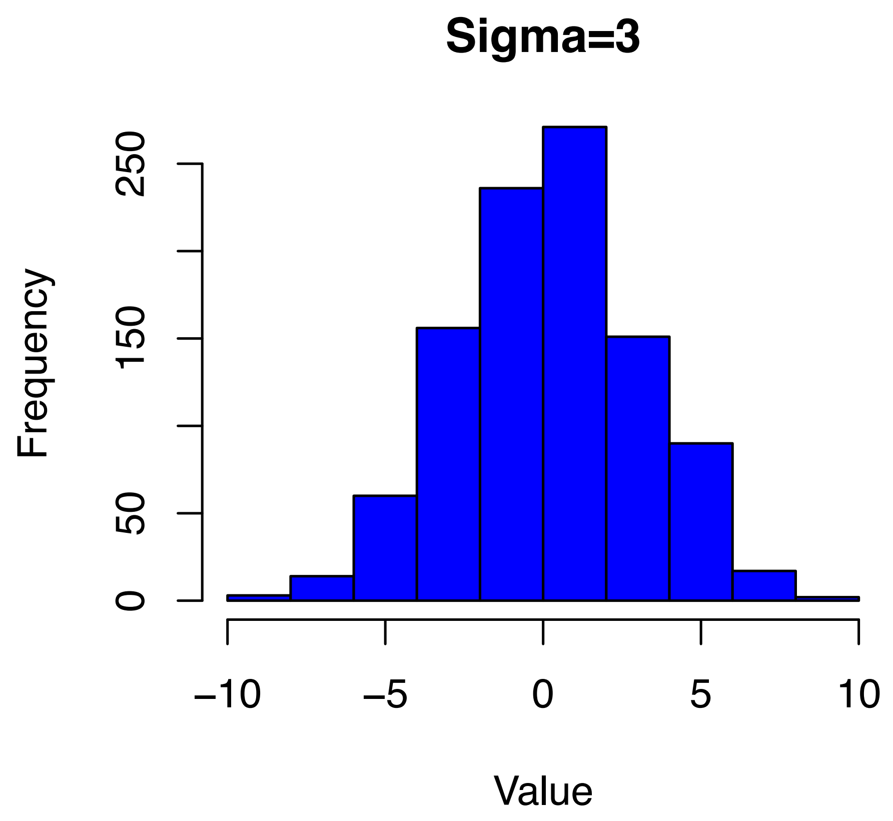
Basic histogram

Makes it in your wd if you don't give path
file = "this is the file name"

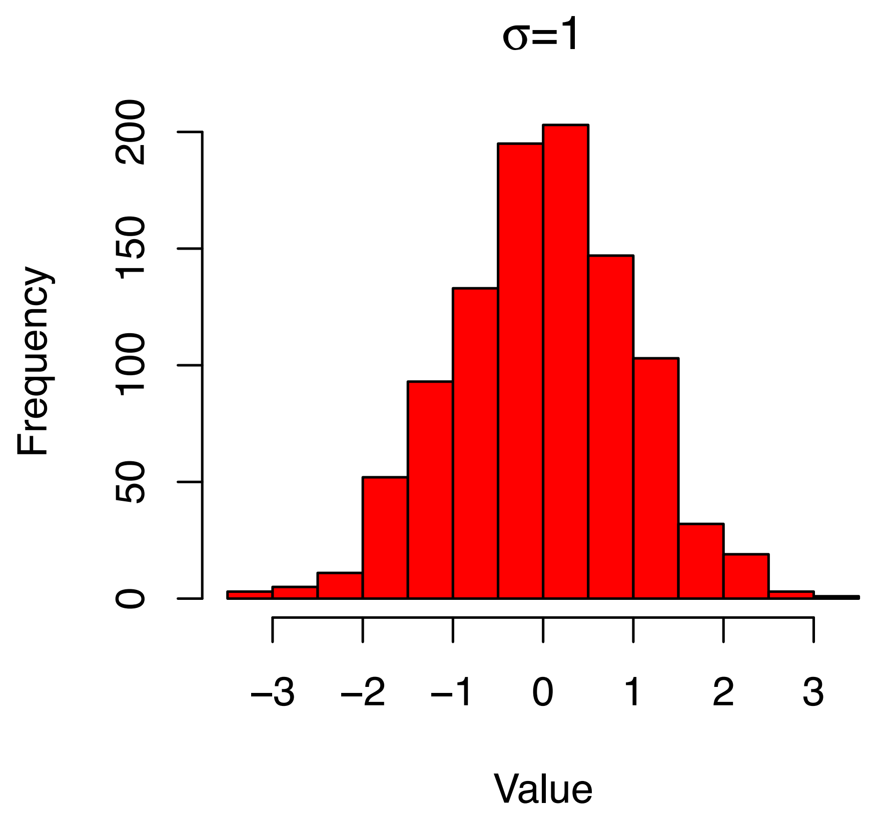
```
> #plot histograms of both on same panel and save to a file:
> pdf(file="Normal_hist.pdf", width=4,height=7);
> #open the file
>       2 rows, 1 col     bottom, left, top, right in inches, guess until it works
> par(mfrow=c(2,1), mar=c(4, 4, 3, 2)) #sets plotting area and margins
>
> hist(s1,col=2,xlab="Value",main="Sigma=1") #make first hist
>
> hist(s3,col=4,xlab="Value",main="Sigma=3") #make second hist
>
> dev.off() #shuts off current output device    Closes the pdf
quartz
     2
```

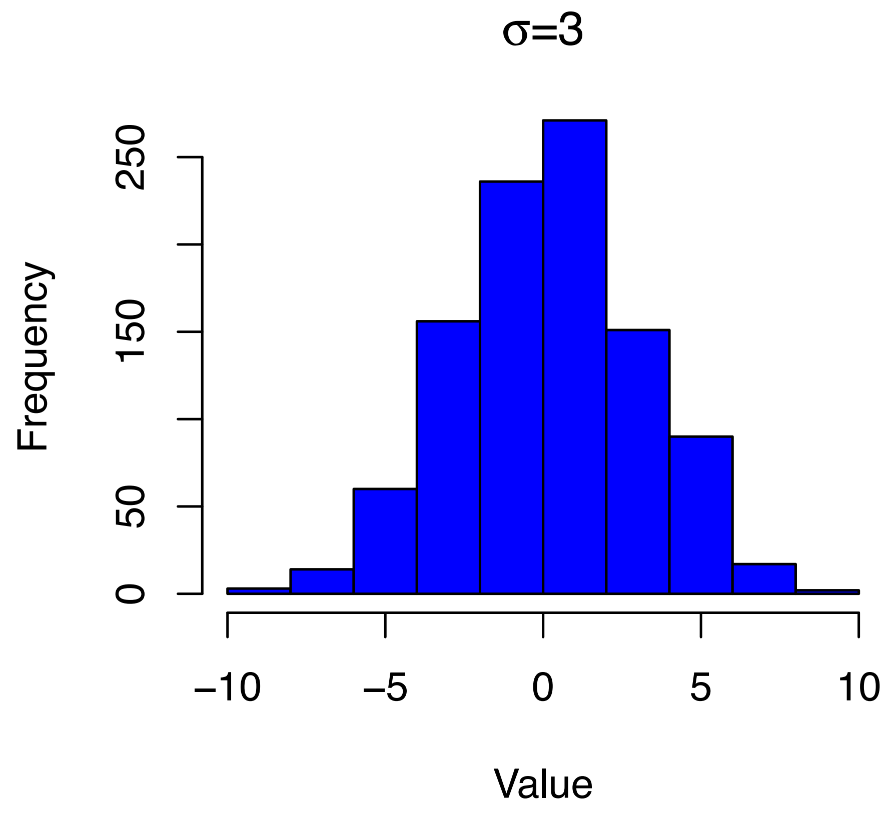
Basic histogram





Getting fancier...





How did I do that?

```
>> #plot histograms of both on same panel and save to a file:
> pdf(file="Normal_hist.fancy.pdf", width=4,height=7);
> #open the file
>
> par(mfrow=c(2,1), mar=c(4, 4, 3, 2)) #sets plotting area and
margins
>
> hist(s1,col=2,xlab="Value",main=expression(paste(sigma,"=1")))
#make first hist
>
> hist(s3,col=4,xlab="Value",main=expression(paste(sigma,"=3")))
#make second hist
>
> dev.off() #shuts off current output device
pdf
  2
```

Expression converts sigma to greek letter, nested with paste inside don’t know why paste is inside, but has to be

Smooth density plot

```
> #make smooth density plot:
>
```

size matters for the other setting ie line width

```
> pdf(file="Normal_density.pdf", width=6,height=6); #open the file
>
> par(mfrow=c(1,1), mar=c(4, 4, 3, 2)) #sets plotting area and margins
>
```

plot first density

```
> plot(density(s1),col=2,lwd=4,xlab="Value",xlim=c(-15,15),main="Normal
distribution")
>
```

this adds line to existing plot

```
> lines(density(s3),col=4,lwd=4) #add the SD=3 values
>
```

doesn’t use c( ) for x and y, sets top left corner **color for each line**

```
> legend(-15,0.4,c("sigma=1","sigma=3"),lwd=4,col=c(2,4),cex=1.5) #put a legend on
>
```

cex -> how big. 1.5 bigger than default

```
> #we can highlight the upper 10% of each distribution with a vertical line:
> abline(v=quantile(s1,0.9),lty=2,lwd=3,col=2) #puts a vertical line onto the plot
for s1
```

abline(v=location) ad a vert line there, h does horizontal

```
> abline(v=quantile(s3,0.9),lty=2,lwd=3,col=4) #puts a vertical line onto the plot
for s3
```

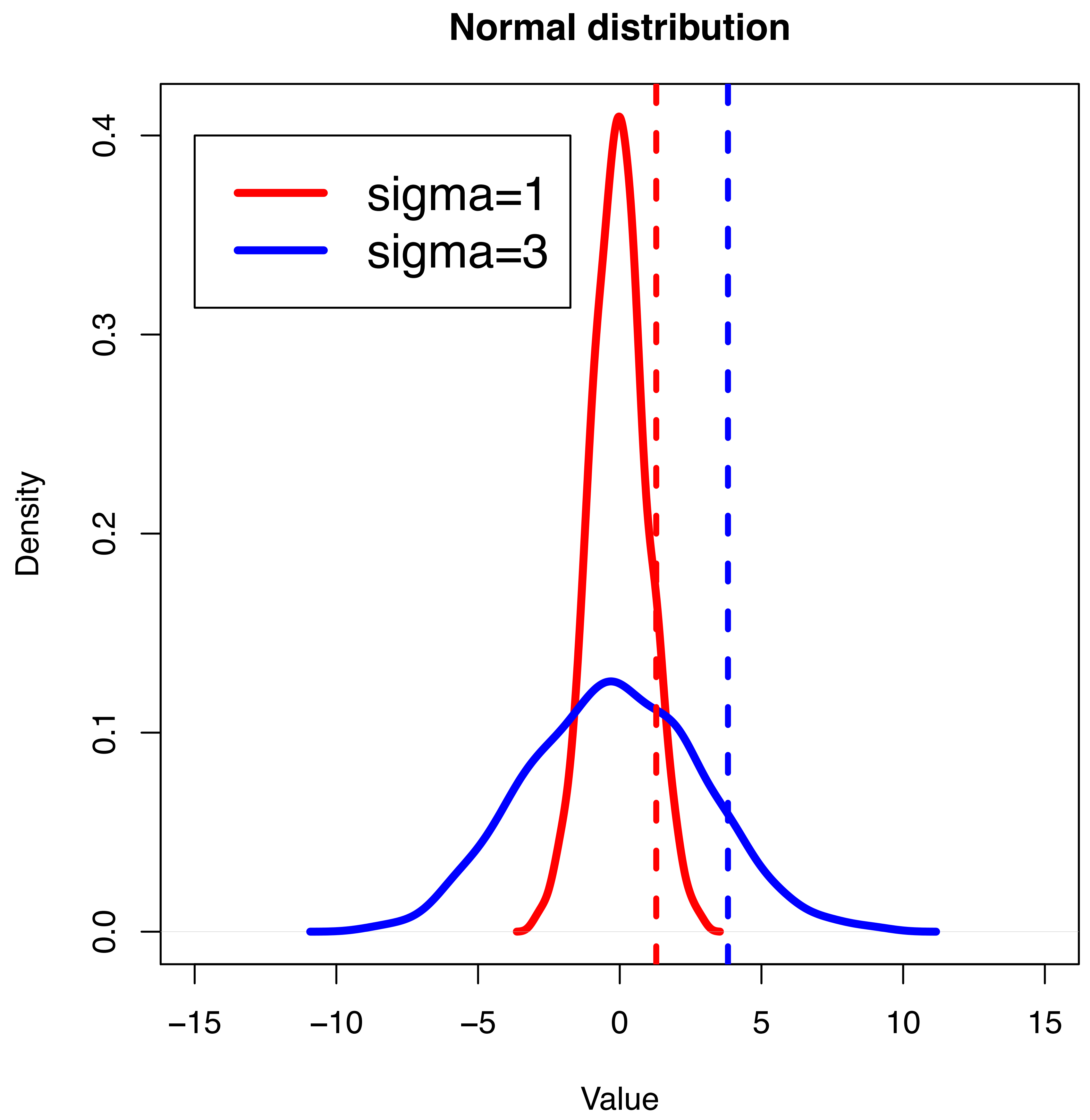
lty is line type

```
> dev.off()
quartz
     2
```

quantile(s3 , 0.9) say mark 90th percentile of s3

If you miss up legend, you can’t just remove it. Have to start over

Smooth density plot



More on “quantile”

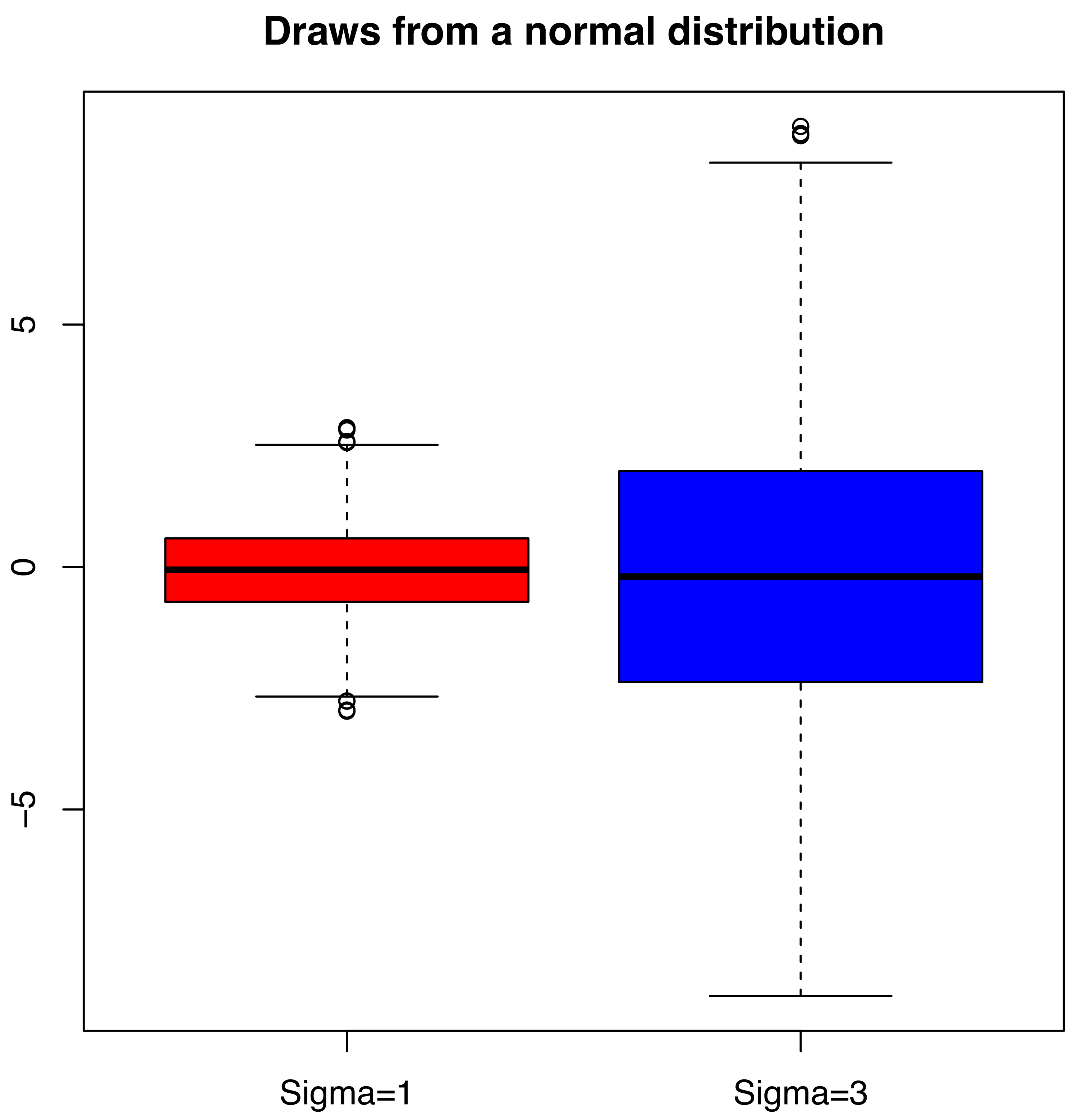
```
> #quantile take a vector of stuff, and returns the value q such that p%
of your distribution is less than q.
>
> #for example, find the 75th percentile of the standard normal
distribution:
> quantile(s1,0.75)
      75%
0.5899364
>
> #quantile with just a vector gives some interesting stuff:
> quantile(s1)
         0%         25%         50%         75%        100%
-2.97189479 -0.71435745 -0.05515638  0.58993639  2.87407876
```

Boxplot

```
> #boxplot:
> pdf(file="Normal_boxplot.pdf", width=6,height=6); #open the file
>
> par(mfrow=c(1,1), mar=c(4, 4, 3, 2)) #sets plotting area and margins
>
> boxplot(cbind(s1,s3),names=c("Sigma=1","Sigma=3"),main="Draws from a
normal distribution",col=c(2,4))
>
> dev.off()
quartz
     2
```

cbind sticks s1 and s3 side by side as two cols, each col is a box so if you use c() you just get one box

Boxplot



Histogram with both sets of data on same axes? Can we do it? YES WE CAN!

```
> #Let's make a histogram of these values, but putting both
on the same axes.
> #But, we need to have the same bin widths for both
datasets:
>
> bins<-seq(-10,10,by=1)
> hist(s1,breaks=bins)$breaks
 [1] -10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0   1   2
3   4   5   6   7   8   9  10
>
> hist(s3,breaks=bins)$breaks
 [1] -10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0   1   2
3   4   5   6   7   8   9  10
>
> #This looks good
>
> counts_s1<-hist(s1,breaks=bins)$counts
> counts_s3<-hist(s3,breaks=bins)$counts
```

bin range depends on your data of course

This is setting the bins

confirming we've set the bins for both and they're the same

now we're storing the counts based on our bins

Histogram with both sets of data on same axes? Can we do it? YES WE CAN!

```
> #now make the plot:
> pdf(file="normal_barplot.pdf", width=6,height=6); #open the
file
>
> par(mfrow=c(1,1), mar=c(4, 4, 3, 2)) #sets plotting area and
margins
>
```

rbind so each bin is counts from both sets

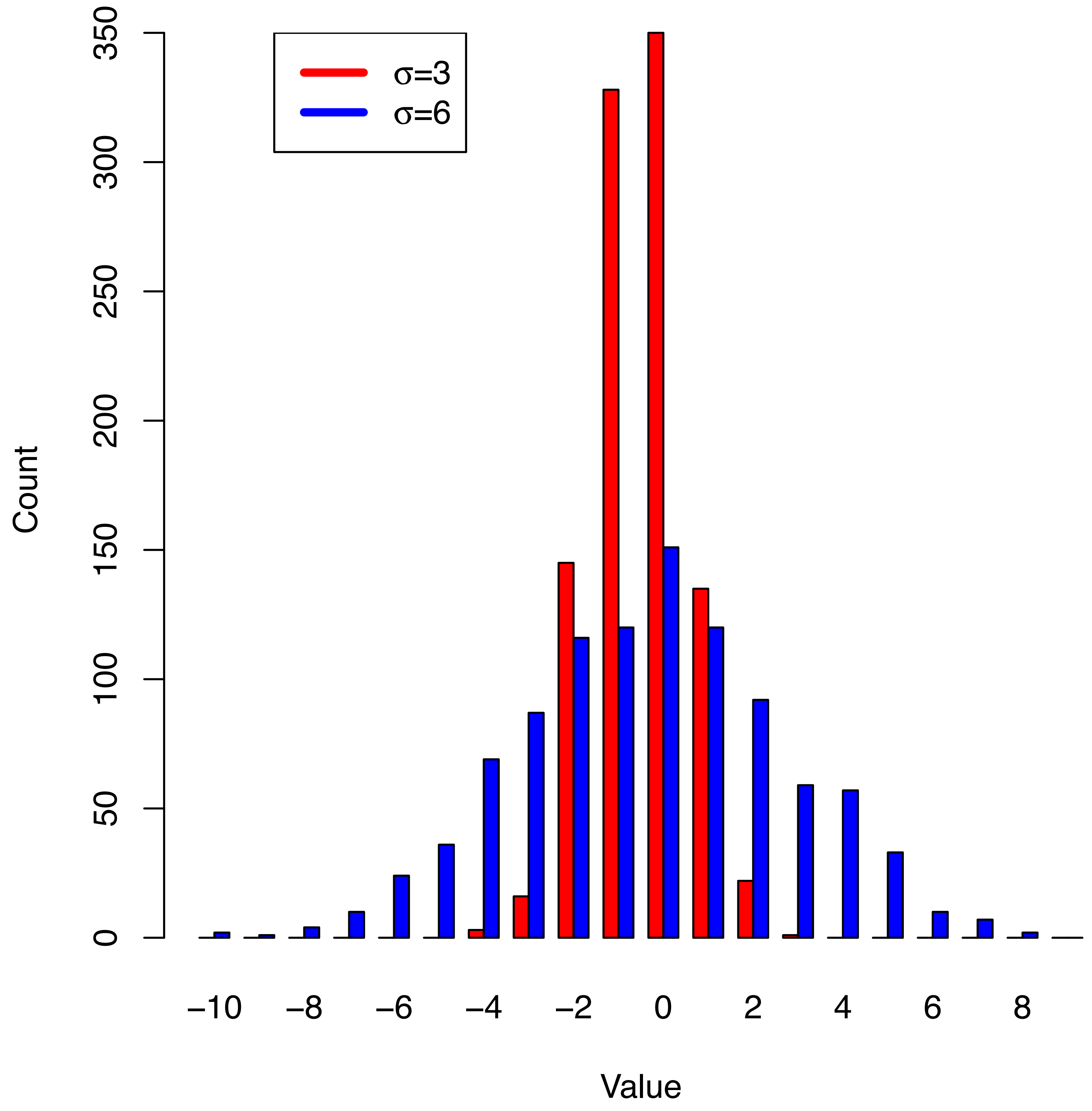
beside sets hist next to eachother

```
>
barplot(rbind(counts_s1,counts_s3),col=c(2,4),beside=T,names.arg=
seq(-10,9.5,by=1),xlab="Value",ylab="Count")
>
```

names will depend on your number of bins

```
>
legend(6,350,c(expression(paste(sigma,"=3")),expression(paste(sig
ma,"=6"))),col=c(2,4),lwd=4)
>
> dev.off()
pdf
  2
```

Histogram with both sets of data on same axes? Can we do it? YES WE CAN!



Finding extreme values

Say we want to find the % of values in a vector that are >X...

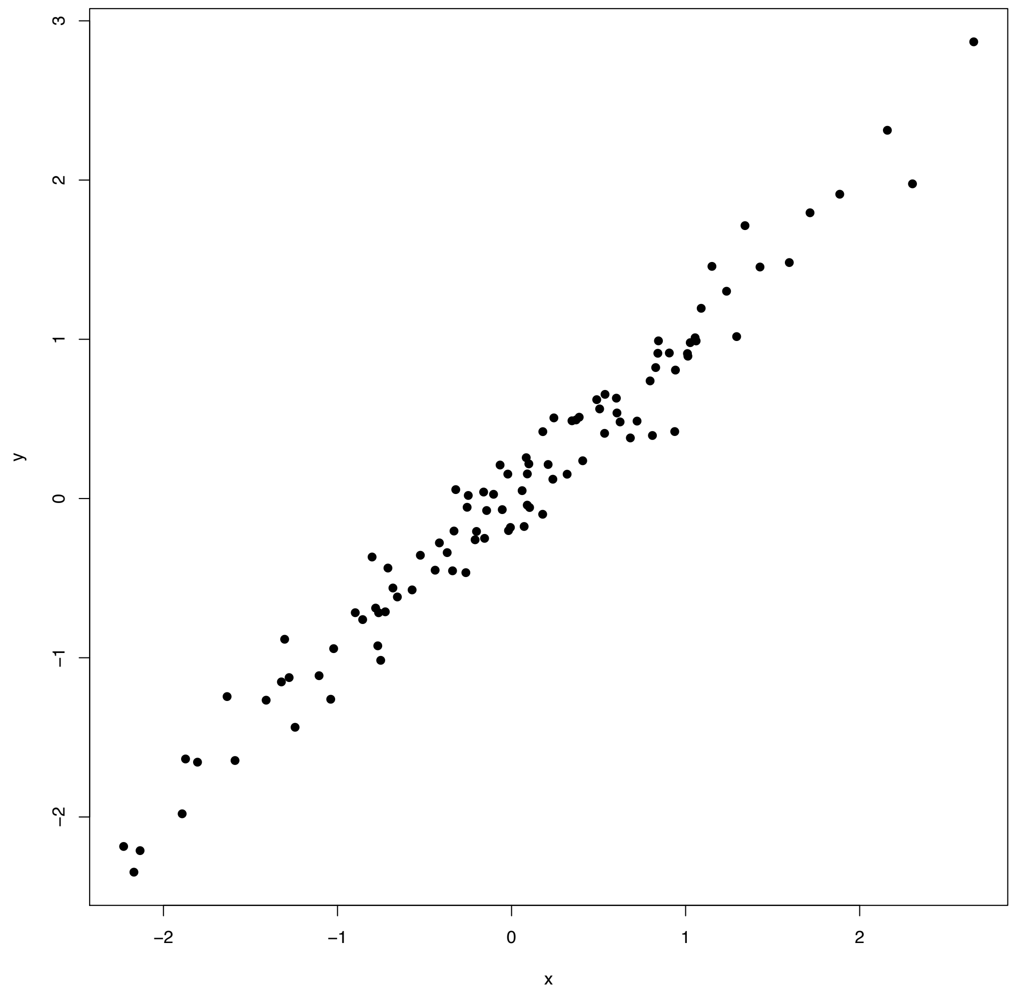
```
> > #We can find the % of values in s1 that are >3:
> mean(s1>3)
[1] 0.001
> #Only 1 of the 1000 values in s1 is >3
>
>
> mean(s3>3)
[1] 0.168
> #16.8% of values in s3 are >3
```

you can count them by doing sum(s1>3)

Scatterplot pitfalls

```
> #Simple scatterplot:
> pdf(file="/Users/kirk/Dropbox/Kirk_stuff/KEL_bootcamp/scatter_small.pdf",
width=10,height=10); #open the file
>
> par(mfrow=c(1,1), mar=c(4, 4, 3, 2)) #sets plotting area and margins
>
> x<-rnorm(100)
> y<-x+rnorm(100,sd=0.2)
>
> plot(x,y,pch=19)
>
> dev.off()
quartz
     2
```

The most annoying thing in R...



Huh?
What is plotted here?
My tired eyes can't read this....

One way to fix it

```
> #now, try again, make the labels bigger:
> #Simple scatterplot:
> pdf(file="/Users/kirk/Dropbox/Kirk_stuff/KEL_bootcamp/scatter_large.pdf",
width=10,height=10); #open the file
>
> par(mfrow=c(1,1), mar=c(5, 5, 3, 2)) #sets plotting area and margins
>
> x<-rnorm(100)
> y<-x+rnorm(100,sd=0.2)
>
> plot(x,y,pch=19,cex.lab=2,cex.axis=2)
>
> dev.off()
quartz
     2
```

.lab change size axis labels

.axis change size of numbers

One way to fix it