Visualizing Data in R

Heide Jackson
heidej@umd.edu

University of Maryland Population Research Center

August 2019

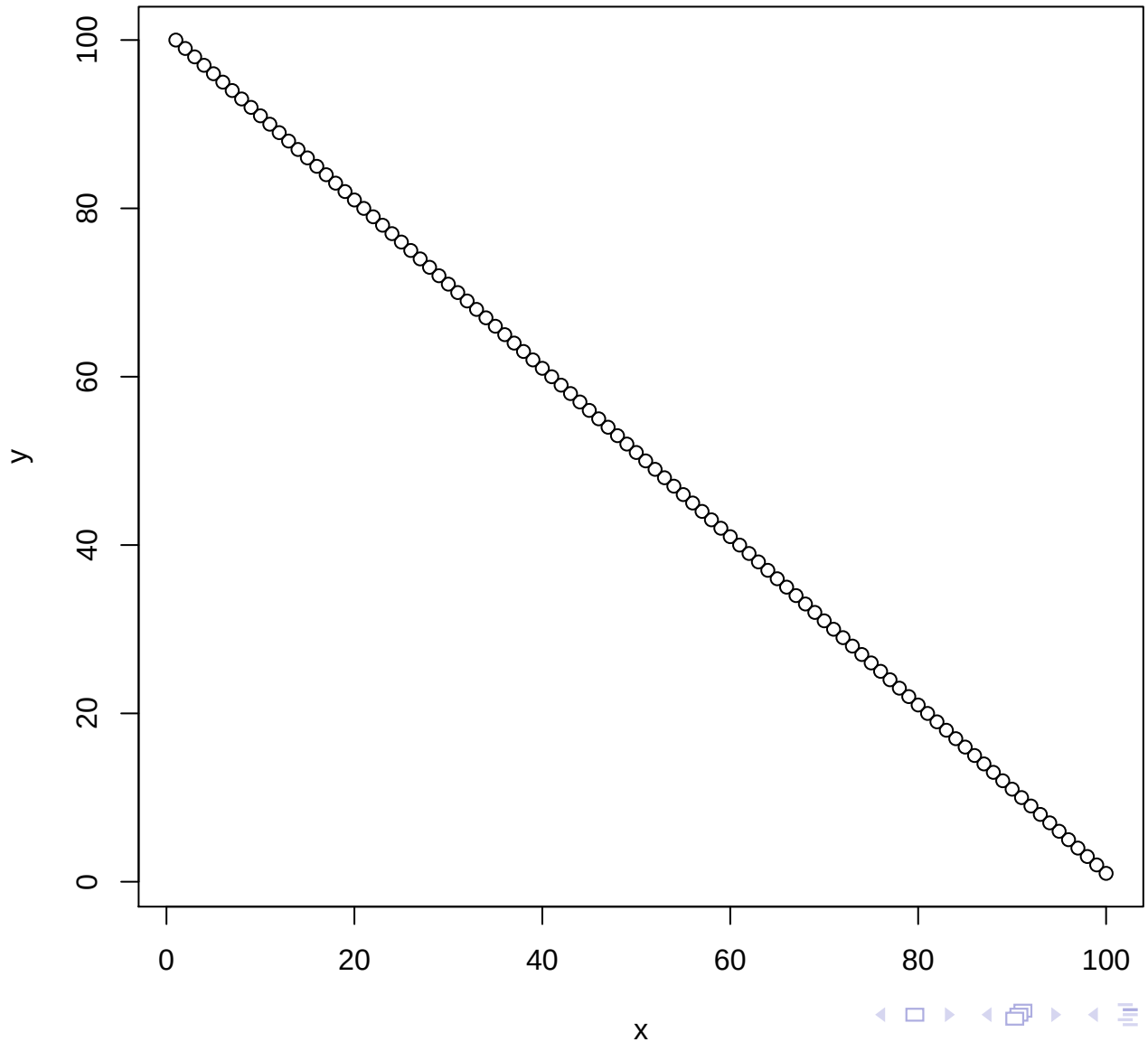
High Level Things to Know

- R makes it very easy to make very cool graphics.
- This functionality is available in base R and with the addition of other packages like ggplot2.
- This presentation will show simply what can be done in Base R without additional packages.
- Results from R models can often be easily plotted to visually interpret results and diagnose issues in model fit.

A Simple Graph

```
x<-1:100
y<-100:1
plot(x,y)
```

A simple graph

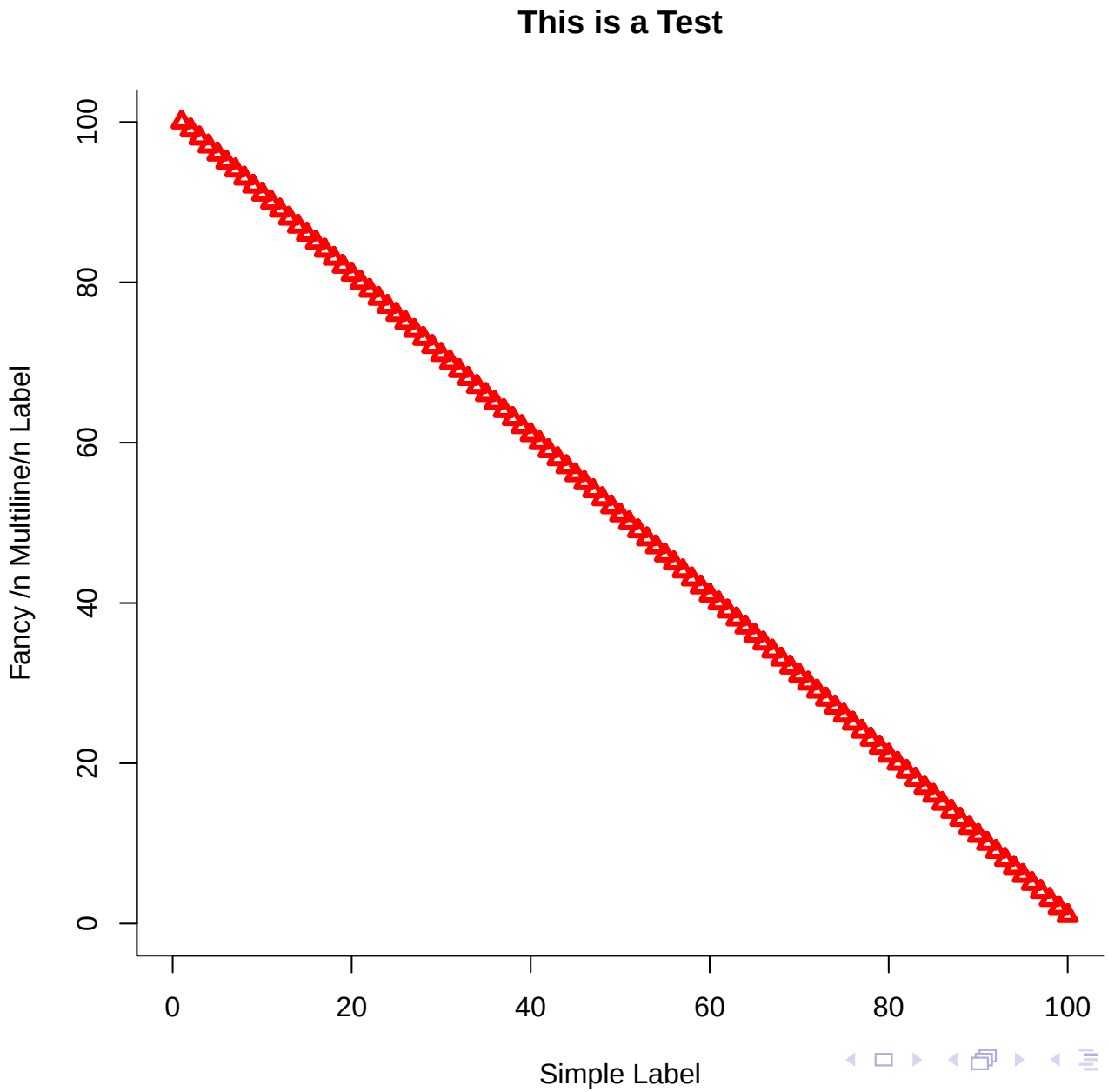


A simple graph

Many Options within Plot

```
x<-1:100
y<-100:1
plot(x,y,
ylab="Fancy /n Multiline/n Label", #labels Y axis
xlab="Simple Label", #labels X axis
main="This is a Test", #Graph title
xlim=c(0,100), #Range of X axis
ylim=c(0,100), #Range of Y axis
type="b", #Specify p for point, l for line, or b for both
lty=2, #Type of Line
pch=2, # Type of point, can also be a character in quotes
lwd=3, #specify line thickness
col="red", #Color of plotted objects
bty="l" #Type of box around plot
)
```

A Simple Graph with Options Added

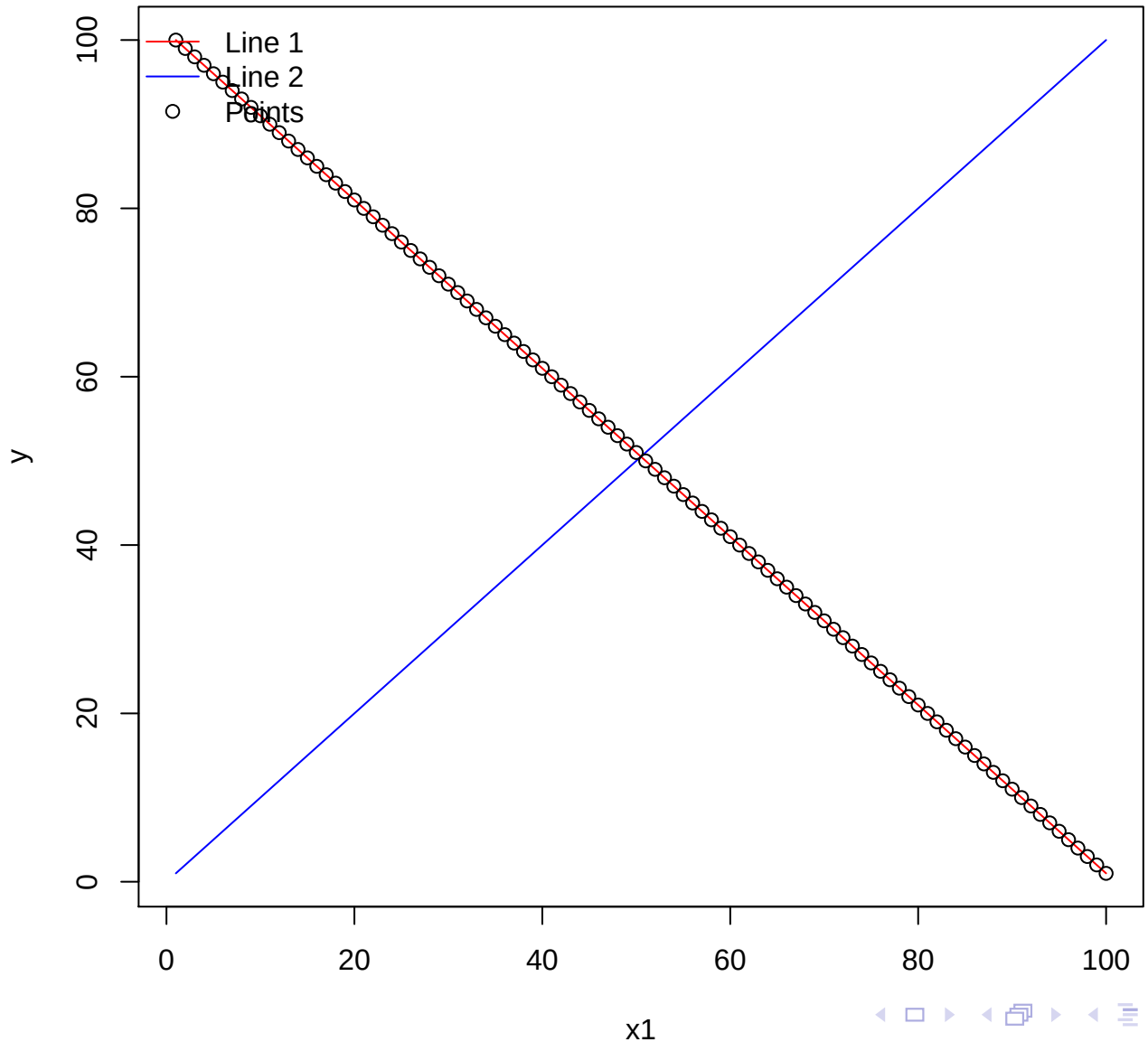


Adding to the Plot

- Plots can be added in multiple steps. Objects can be overlayed to an initial plot.

```
x1<-1:100
y<-100:1
x2<-100:1
plot(x1,y, col="red", type="l")
lines(x2, y, col="blue") #add a line that is blue
points(x1,x2) # add points with these coordinates
legend("topleft", c("Line 1", "Line 2", "Points"),
col=c("red", "blue", "black"), lty=c(1,1,NA),
pch=c(NA, NA, 1), bty="n")
```

Multi Object Plot



Adding to the Plot

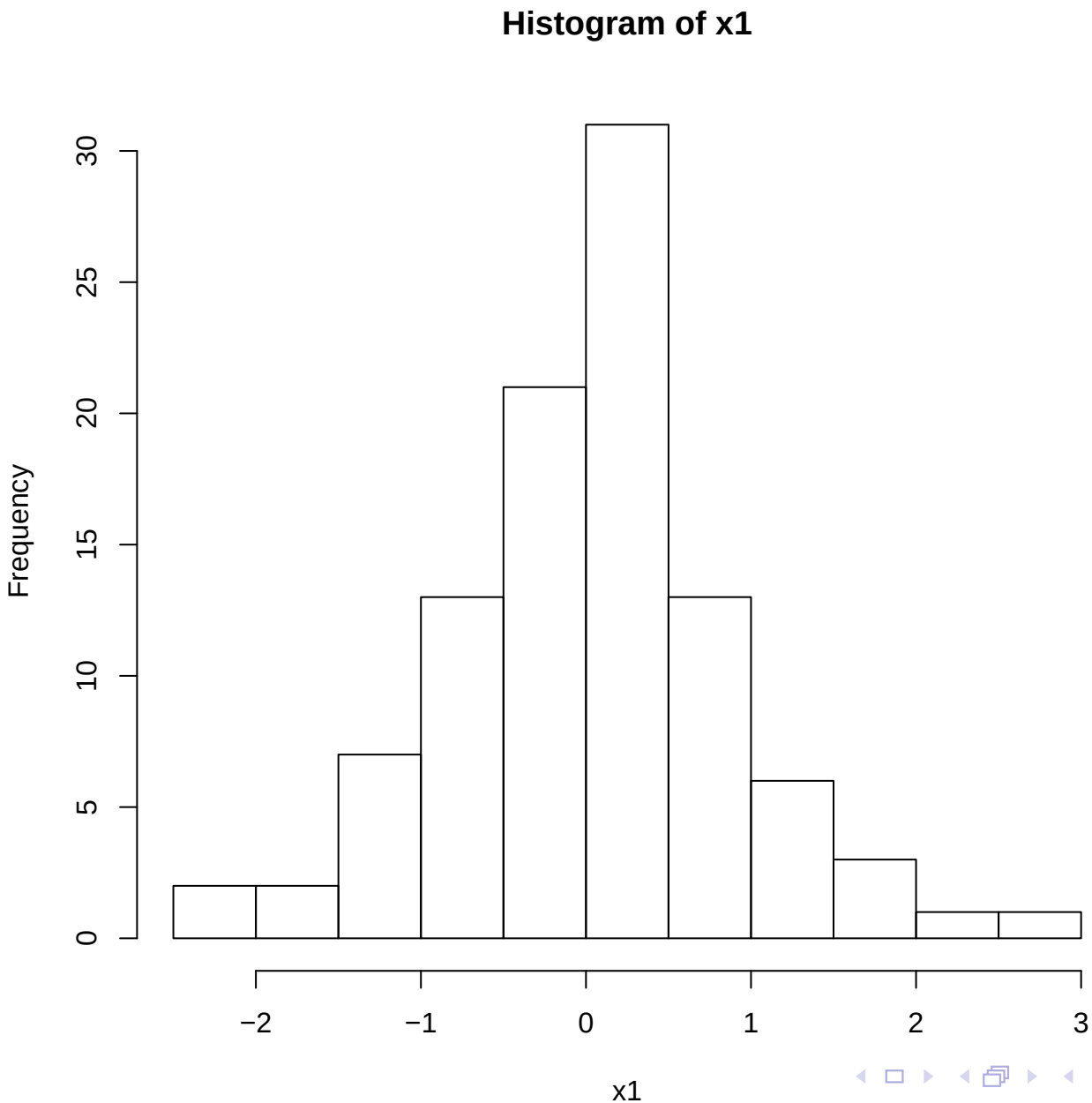
Unpacking the Legend

```
legend("topleft", #specify legend location
c("Line 1", "Line 2", "Points"), #Specify labels
col=c("red", "blue", "black"), #Specify object colors
lty=c(1,1,NA), #Specify line type for plotted objects
pch=c(NA, NA, 1), #Specify point types for legend objects
bty="n") # specify if there should be box around legend
```

Creating a Histogram

```
x1<-rnorm(100,0,1)
y<-100:1
hist(x1)
```

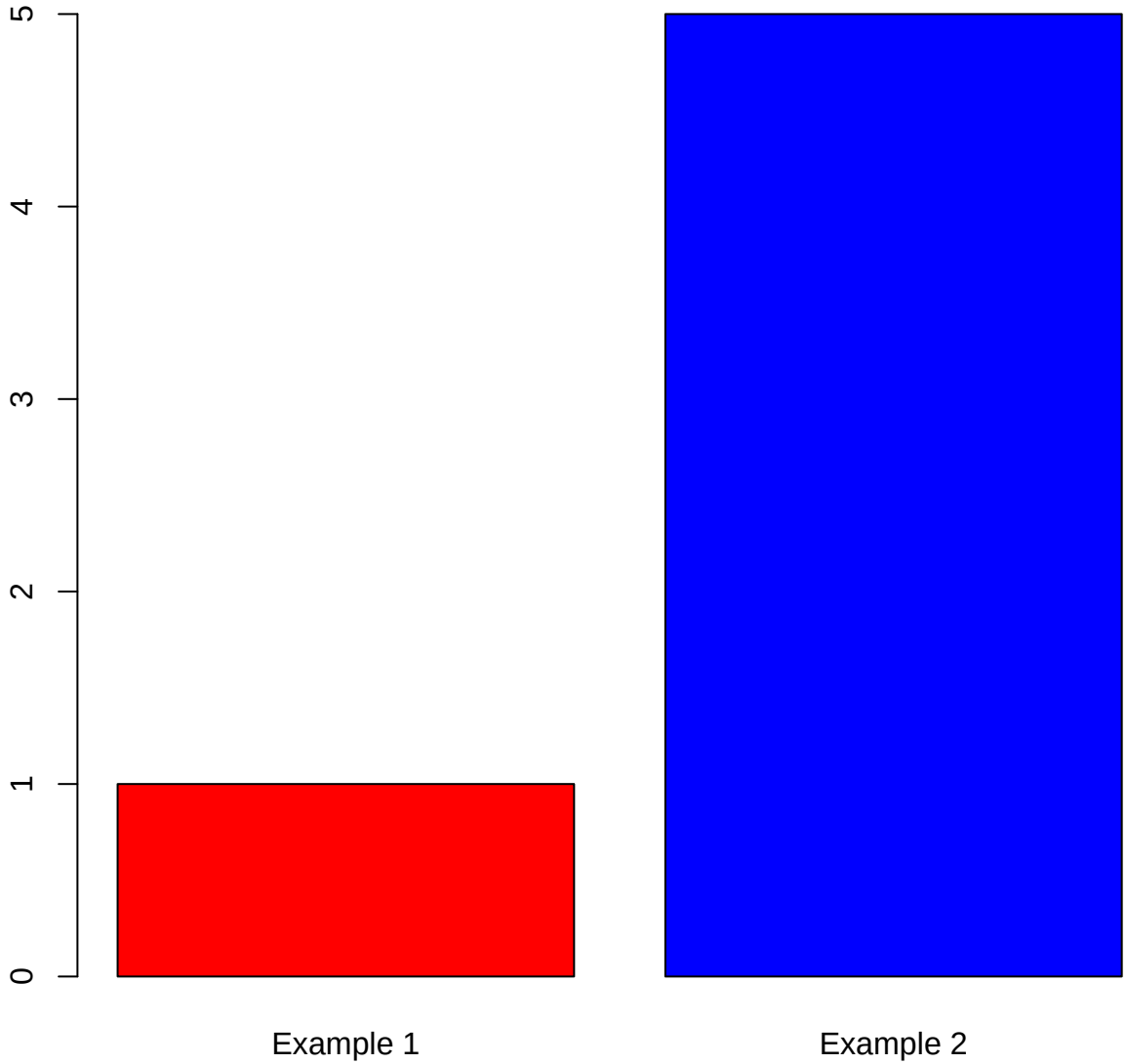
A Histogram



Creating a Bar Graph

```
name=c("Example 1", "Example 2")
x<- c(1,5)
barplot(x, names=name, col=c("red", "blue"))
```

A Bar Graph



Exporting the Data

- Standard text to export results placed ahead of and after created plot.
- R supports many formats but all have a similar syntax. Three I commonly use are pdf(), png(), and tiff()
- Here's a quick example:

```
pdf("test.pdf")
plot(x,y)
dev.off()
```

- Different formats allow for different specification of image size and in some cases resolution–this is particularly useful for posters and journal articles.