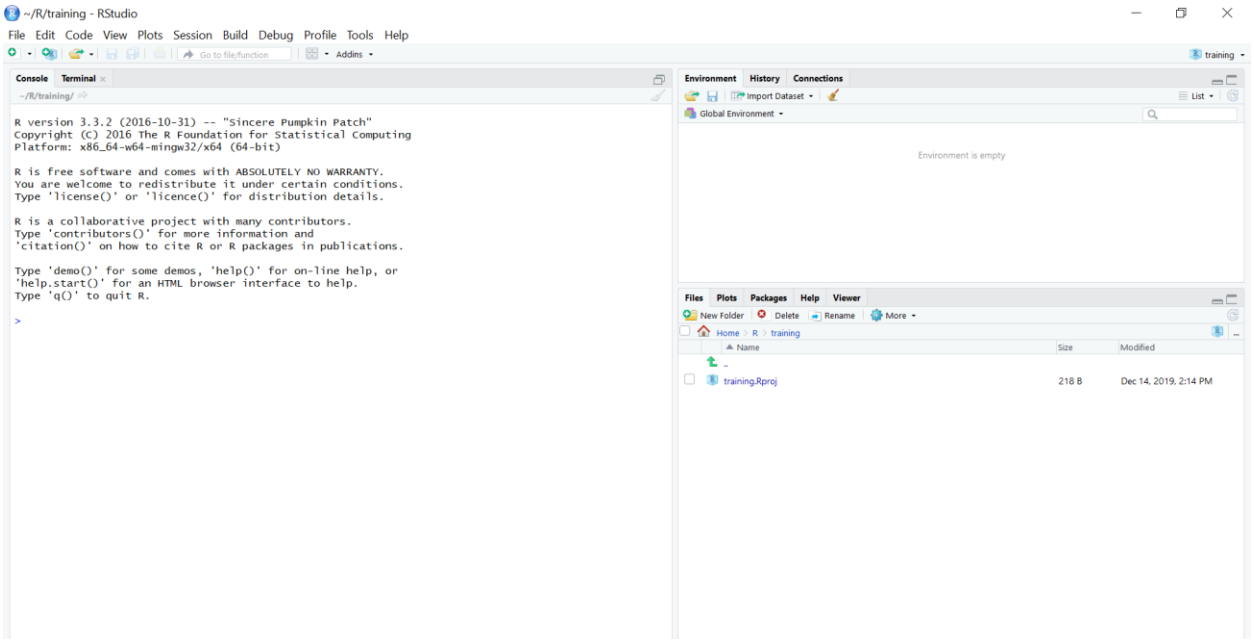
Mini-Tutorials for Creating Graphs in R

Normal Distributions with Shading

The examples below are intended to instruct you to create statistical graphs in R with minimal initial training in R. You should be able to follow the example codes to obtain graphs by modifying the included code. Some examples (and a key) will be included at the end of the document for practice. The screenshots I will show of the environment use R Studio, which is a free program you can find online. Other R environments will look different, but probably have similar functionality.

When you open up a new project environment in R Studio, it looks like this.



The command line environment is on the left. Images when we construct them will appear on the bottom right. As we add variables, they will appear in the list at the top right (name, dimensions and samples will display, which is useful for checking that you didn't skip entries when entering data by hand).

We are going to start by creating a bar graph (the same as a column graph in Excel). The data should be pre-summarized for this example. This method will work for pre-summarized histogram data, too.

Copy the commands shown into the command line.

Step 1. We don't need data for this graph. Just some information for the distribution we want to draw. First, we need to specify the properties of the normal distribution we want. We'll start

with the standard normal distribution where the mean is zero and the standard deviation is one. We also need to specify the upper and lower bounds to be shaded.

```
mean=0; sd=1
lb=-1.64; ub=0.5
```

Step 2. Then we need some points to plot for our graph for both x and y.

```
x <- seq(-4,4,length=100)*sd + mean
hx <- dnorm(x,mean,sd)
```

Step 3. Graph the results.

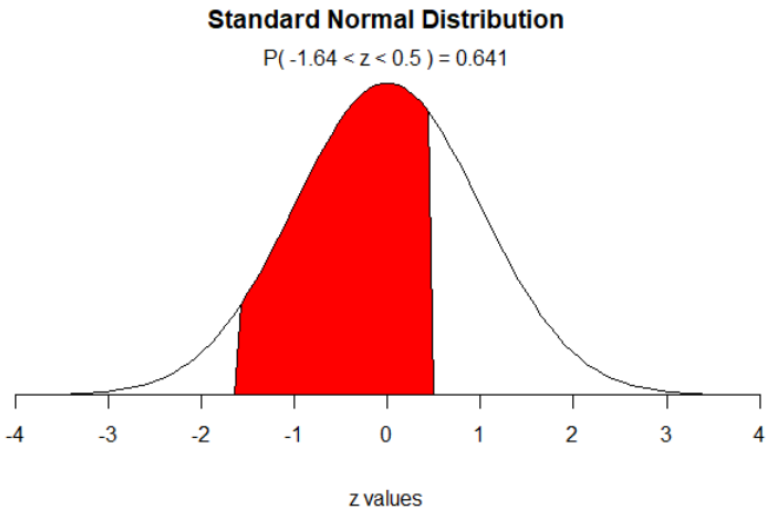
```
plot(x, hx, type="l", xlab="z values", ylab="", main="Standard Normal Distribution", axes
=FALSE)
```

Step 4. Then add the shading.

```
i <- x >= lb & x <= ub
lines(x, hx)
polygon(c(lb,x[i],ub), c(0,hx[i],0), col="red")
```

Step 5. And then, we get the probability to get the label for the header, and add the axis back in.

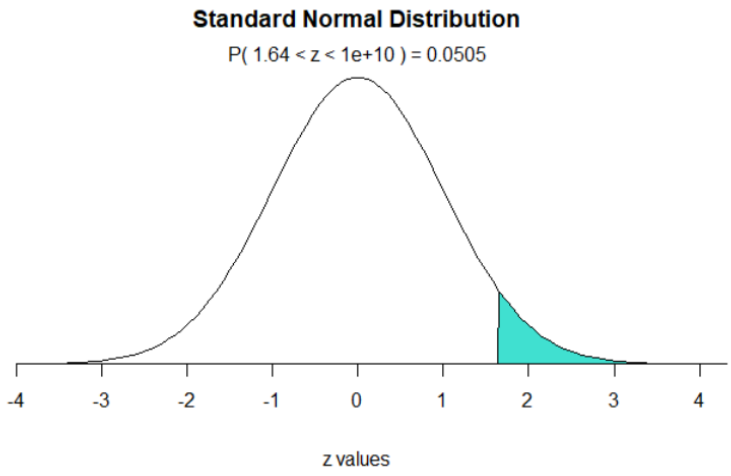
```
area <- pnorm(ub, mean, sd) - pnorm(lb, mean, sd)
result <- paste("P(",lb,"< z <",ub,") =", signif(area, digits=3))
mtext(result,3)
axis(1, at=seq(-4, 4, 1), pos=0)
```



Let's try a second one with an upper bound near infinity.

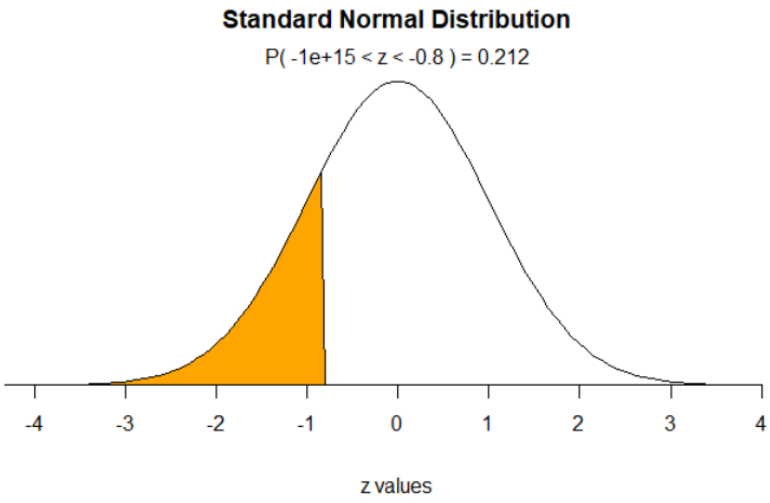
```
mean=0; sd=1
lb=1.64; ub=10^10
x <- seq(-4,4,length=100)*sd + mean
hx <- dnorm(x,mean,sd)
plot(x, hx, type="l", xlab="z values", ylab="", main="Standard Normal Distribution", axes
=FALSE)
i <- x >= lb & x <= ub
lines(x, hx)
polygon(c(lb,x[i],ub), c(0,hx[i],0), col="turquoise")
area <- pnorm(ub, mean, sd) - pnorm(lb, mean, sd)
result <- paste("P(",lb,"< z <",ub,") =", signif(area, digits=3))
mtext(result,3)
axis(1, at=seq(-4, 4, 1), pos=0)
```

I've highlighted the changes to get the graph below.



Practice.
For practice, make one now in the left tail, with $z < -0.8$.

Solutions.
Adjust the second line, and the color if you wish.



You can put multiple commands on one line separated by semi-colons so that you don't have to calculate the intermediate results one at a time.