

Homework 3 R Exercise

We will use this exercise to graph uniform and normal density functions and highlight probabilities based on Exercises 5.3 and 5.4 and the standard normal density curve. Let's start out by graphing the density function of the uniform random variable $Y \sim Unif(10, 30)$. We first use a fine grid of 200 x values spaced from 10 to 30, even though we really need only two values because the uniform density is flat—it's good to have a generalizable approach. Then we evaluate the density and save it in **fx**. We plot the data so that the (x, y) values are connected by lines using the **type="l"** option, while expanding the axes so that the density stands out, and changing the y axis label to something more descriptive, $f(x)$. The **polygon** command fills in the density, and we add red highlights with the **lines** command.

```
x=seq(10,30,length=200)
fx=dunif(x,min=10,max=30)
plot(x,fx,type="l",xlim=c(5,35),ylim=c(0,0.06),ylab="f(x)")
#This polygon shades the entire density
polygon(c(10,x,30),c(0,fx,0),col="lightgray",border=NA)
#This outlines the density from 10 to 30
lines(x,fx,type="l",lwd=2,col="red")
#This outlines the density from 5 to 10
lines(c(5,10),c(0,0),type="l",lwd=2,col="red")
#This outlines the density from 30 to 35
lines(c(30,35),c(0,0),type="l",lwd=2,col="red")
```

Suppose you wanted to highlight the area in the graph corresponding to $P(20 < X < 25)$; adding another **polygon** statement will work. Note that you need to select the values of x and $f(x)$ where x is between 20 and 25. **R** uses a logical statement in brackets to select the right value; the syntax is unwieldy at first.

```
plot(x,fx,type="l",xlim=c(5,35),ylim=c(0,0.06),ylab="f(x)")
polygon(c(10,x,30),c(0,fx,0),col="lightgray",border=NA)
#This polygon shades the area between 20 and 25
polygon(c(20,x[x>20 & x<25],25),c(0,fx[x>20 & x<25],0),col="darkgray",border=NA)
lines(x,fx,type="l",lwd=2,col="red")
lines(c(5,10),c(0,0),type="l",lwd=2,col="red")
lines(c(30,35),c(0,0),type="l",lwd=2,col="red")
```

You can add labels for features you want to highlight by using the **text** command, e.g. $\mu - \sigma < X < \mu + \sigma$. The axis labels and tickmarks can get in the way, and may need to be removed depending on what you choose to plot. E.g., I chose not to include μ because it would coincide with both the x value 20 and the axis label x .

```
#Compute sigma and compute upper and lower limits
sigmax=sqrt((30-10)^2/12)
LL=20-sigmax
UL=20+sigmax
plot(x,fx,type="l",xlim=c(5,35),ylim=c(0,0.06),ylab="f(x)")
polygon(c(10,x,30),c(0,fx,0),col="lightgray",border=NA)
#Adjust the values to plot
polygon(c(LL,x[x>LL & x<UL],UL),c(0,fx[x>LL & x<UL],0),col="darkgray",border=NA)
lines(x,fx,type="l",lwd=2,col="red")
lines(c(5,10),c(0,0),type="l",lwd=2,col="red")
lines(c(30,35),c(0,0),type="l",lwd=2,col="red")
#The text command usually requires x, y and a simple text string in quotes. We need to use expression
text(14.23,-0.02,expression(mu-sigma),xpd=NA)
text(25.77,-0.02,expression(mu+sigma),xpd=NA)
#Some arrows help
```

```
arrows(14.23,-0.018,14.23,0,length=0.08,xpd=NA)
arrows(25.77,-0.018,25.77,0,length=0.08,xpd=NA)
```

To wrap up, let's try a graph of a normal probability, say $P(-1 < Z < 1.5)$. Change **x** to range from -3 to 3, and we evaluate the density using **dnorm** rather than **dunif**. The **plot** and **polygon** commands look similar, though a red border is added to the second polygon area. Printing the probability inside the shaded area is tricky, since we need to print a value that first needs to be calculated; the **R** functions **eval** and **expression** are needed (let's not dwell on that step).

```
x=seq(-3,3,length=200)
fx=dnorm(x,0,1)
plot(x,fx,type="l",xlim=c(-3,3),ylab="f(x)")
polygon(c(-3,x,3),c(0,fx,0),col="lightgray",border=NA)
polygon(c(-1,x[x>-1 & x<1.5],1.5),c(0,fx[x>-1 & x<1.5],0),col="darkgray",border="red")
lines(x,fx,type="l",lwd=2,col="red")
#It may be more convenient to compute the probability beforehand, record its value, then enter it as a
text(0,0.15,eval(expression(round(pnorm(1.5)-pnorm(-1),3))))
```