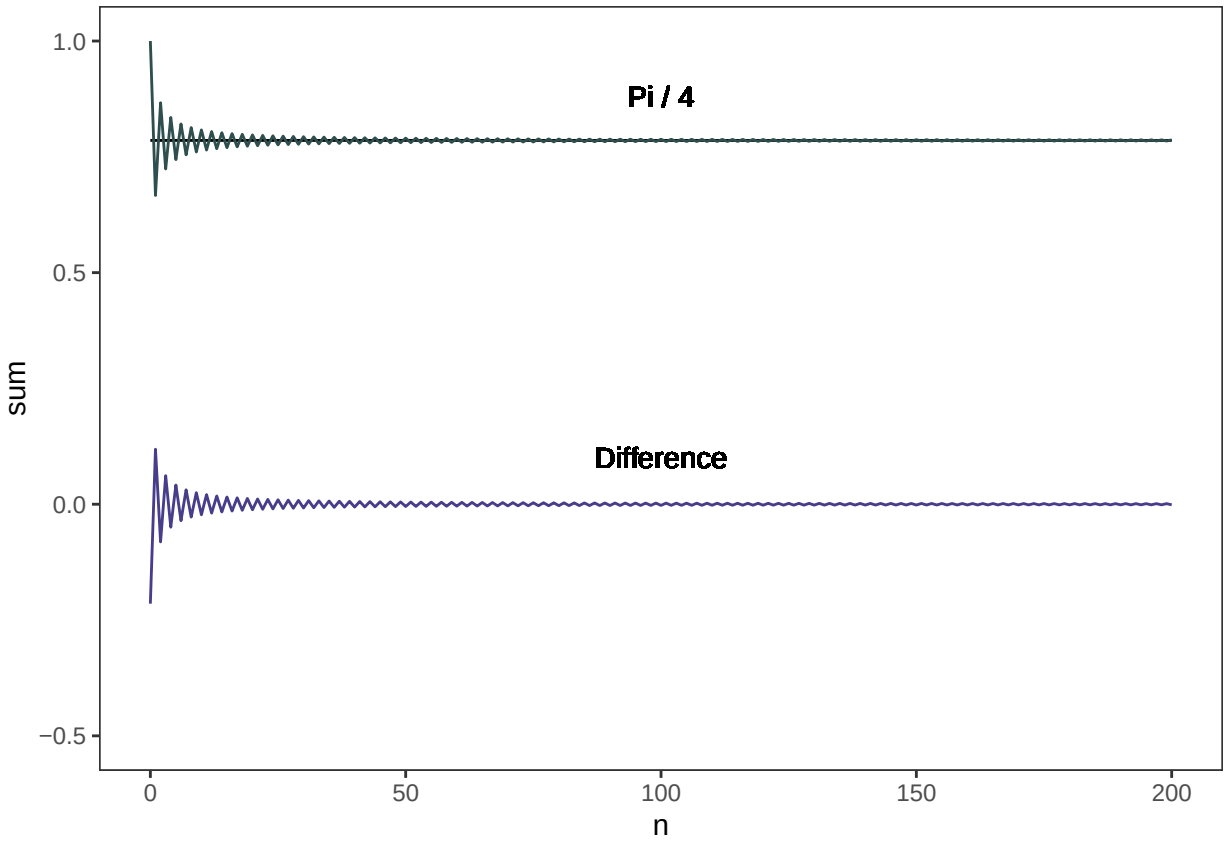
# Class Project #3

Jon Flees

12/11/2021

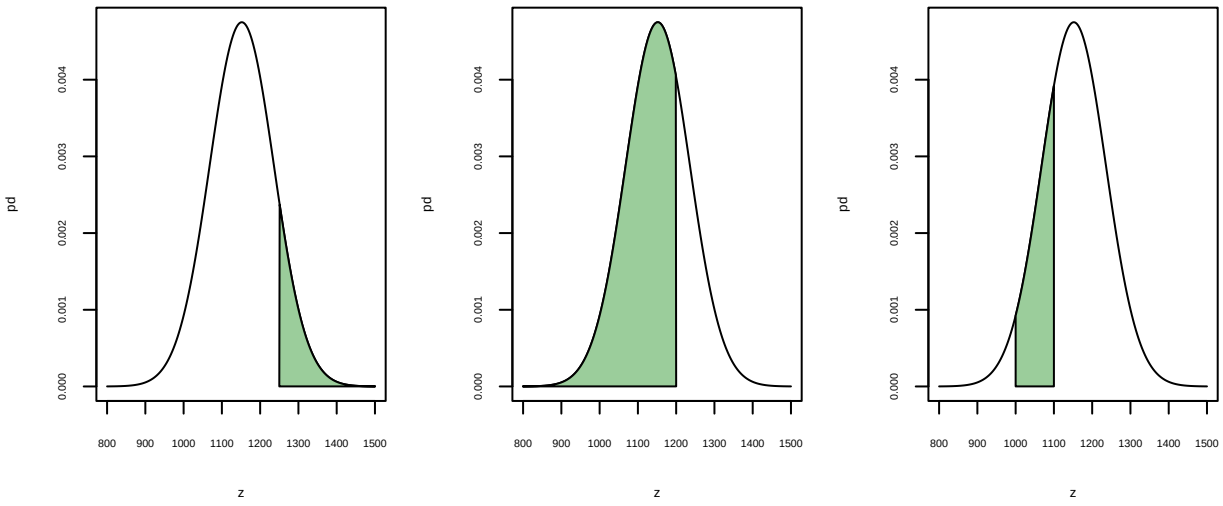
**Problem 1**

```
Leibniz <- function(n) {

  sum_vector <- vector()   # Initialize Sum Vector

  for (k in 0:n) {
    if (k == 0) {     # First Value in Sum Vector
      sum_vector <- c(sum_vector, (-1)^k / (2*k + 1) ) }
    else {            # Sums Leibniz with last Value in Sum Vector (k instead of k-1 bc
                      #                                              idx cannot be 0 in R)
      sum_vector <- c(sum_vector, ( (-1)^k / (2*k + 1) ) + sum_vector[k] ) }
  }
  return(sum_vector)
}

sum_vect <- Leibniz(200)
diff_vect <- (pi/4) - sum_vect
x_vals <- 0:200

df <- data.frame(x_vals, sum_vect, diff_vect)

ggplot(df, aes(x_vals)) +
  geom_line(aes(y=pi/4)) +
  geom_line(aes(y=sum_vect), colour="darkslategray") +    # Felt it needed some color
  geom_line(aes(y=diff_vect), colour="darkslateblue") +
  geom_text(label='Difference',x=100,y=0.1) +
  geom_text(label='Pi / 4',x=100,y=0.88) +
  ylab('sum') + xlab('n') + ylim(-0.5,1.0) +
  theme_bw() +                                  # Since ggplot was used in this problem,
  theme( panel.grid.major = element_blank(),    # Theme settings were used to turn off
         panel.grid.minor = element_blank())    # grid & change background to white
```




**Problem 2**

```
z <- seq(800,1500,1)
pd <- dnorm(z,1152,84)

# 1 x 3 sublot
# Layout function utilized to scale width & height to a more visually appealing ratio
#  than was available with par() alone
layout(matrix(c(1,2,3),ncol=3), widths=c(4,4,4), heights=c(5,5,5), TRUE)
par(mar=c(4,4,1,1))

# Over 1250
plot(z,pd,type='l',cex.axis=.5,cex.lab=.6)
polygon( c(z[z>1250],1250), c(pd[z>1250], pd[z==1500]), col='darkseagreen3')

# Under 1200
plot(z,pd,type='l',cex.axis=.5,cex.lab=.6)
polygon( c(z[z<1200],1200), c(pd[z<1200], pd[z==800]), col='darkseagreen3')

# Between 1000 & 1100
plot(z,pd,type='l',cex.axis=.5,cex.lab=.6)
polygon( c(1000, z[z>=1000 & z<=1100], 1100),
         c(pd[z==800], pd[z>=1000 & z<=1100], pd[z==800]), col='darkseagreen3')
```

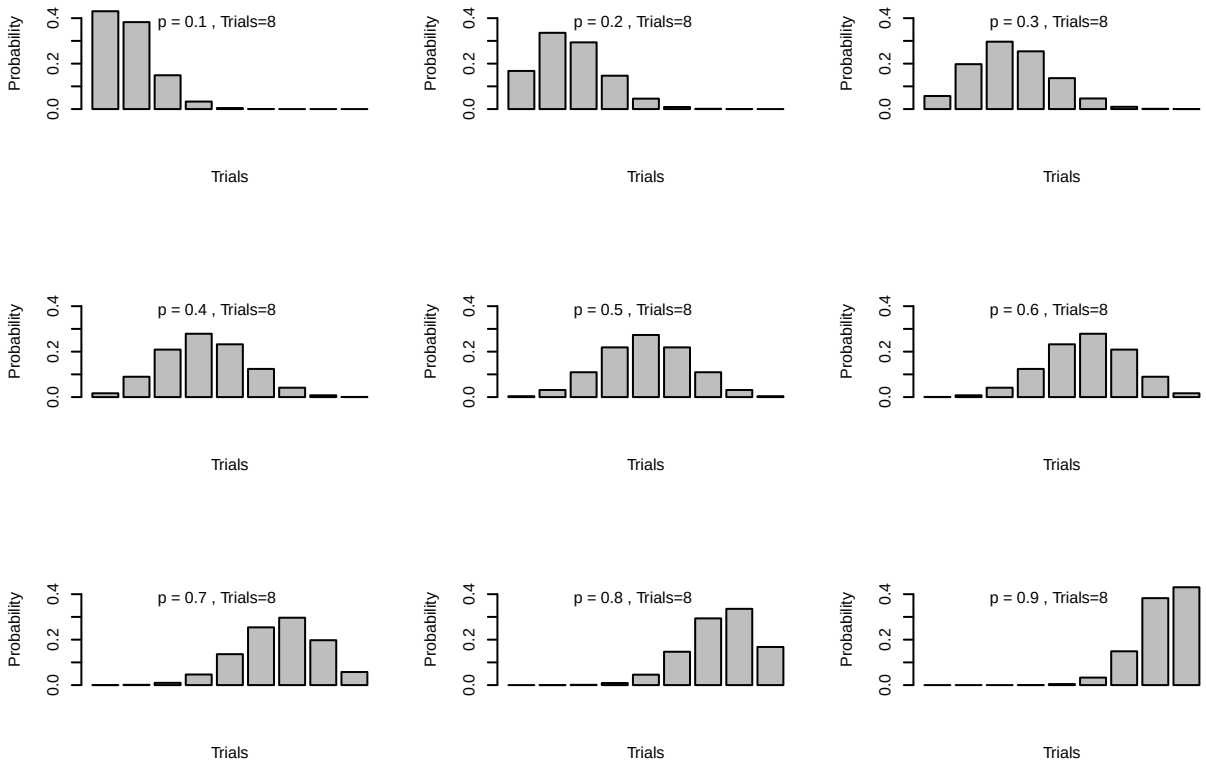



**Problem 3**

```
probSuccess <- function(p,n){
  pS <- vector()
  for (x in 0:n) {
    nCx <- factorial(n) / (factorial(x) * factorial(n-x))
    pS <- c(pS, nCx * (p^x) * (1-p)^(n-x) )
  }
  return(pS)
}

probSuccessTable <- function(p,n){
  pS <- vector()
  col_names <- vector()
  for (x in 0:n) {
    nCx <- factorial(n) / (factorial(x) * factorial(n-x))
    pS <- c(pS, nCx * (p^x) * (1-p)^(n-x) )
    col_names <- c(col_names, paste(x,"success"))
  }
  pS <- as.table(pS)
  pS[length(pS)+1] <- sum(pS)
  names(pS) <- c(col_names, "SUM of probs")
  return(pS)
}

probSuccessTable(.7,8)   # Replication of example table
```

```
##    0 success    1 success    2 success    3 success    4 success    5 success
##   0.00006561   0.00122472   0.01000188   0.04667544   0.13613670   0.25412184
##    6 success    7 success    8 success SUM of probs
##   0.29647548   0.19765032   0.05764801   1.00000000
```

```
# 3 x 3 subplot
par(mfrow=c(3,3), cex=0.5)
for (p in seq(0.1,0.9,0.1) ) {
  text <- paste("p =", p, ", Trials=8")
  barplot(probSuccess(p,8), width=2, xlab ='Trials', ylab='Probability',ylim=c(0,.49))
  text(x=10,y=.38,text)
}
```


p = 0.1 , Trials=8
p = 0.2 , Trials=8
p = 0.3 , Trials=8
p = 0.4 , Trials=8
p = 0.5 , Trials=8
p = 0.6 , Trials=8
p = 0.7 , Trials=8
p = 0.8 , Trials=8
p = 0.9 , Trials=8
Probability
Trials
0.0
0.2
0.4