mid

Aaron

3/3/2018

Problem1-1

```
A <- matrix(1:16, nrow = 4)
apply(A, 1, mean)
```

```
## [1]  7  8  9 10
```

Problem1-2

```
A^2
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1   25   81  169
## [2,]    4   36  100  196
## [3,]    9   49  121  225
## [4,]   16   64  144  256
```

Problem1-3

```
A%*%A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   90  202  314  426
## [2,]  100  228  356  484
## [3,]  110  254  398  542
## [4,]  120  280  440  600
```

Problem1-4

```
eigen(A)$vectors%*%diag(eigen(A)$values)^2%*%ginv(eigen(A)$vectors)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   90  202  314  426
## [2,]  100  228  356  484
## [3,]  110  254  398  542
## [4,]  120  280  440  600
```

Problem1-5

```
A[3,]%*%A[,3]
```

```
##      [,1]
## [1,]  398
```

Problem1-6

```
c(1,1,1,1)%o%A[4,]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    4    8   12   16
## [2,]    4    8   12   16
## [3,]    4    8   12   16
## [4,]    4    8   12   16
```

Problem1-7

```
A[,c(1,4)]
```

```
##      [,1] [,2]
## [1,]    1   13
## [2,]    2   14
## [3,]    3   15
## [4,]    4   16
```

Problem1-8

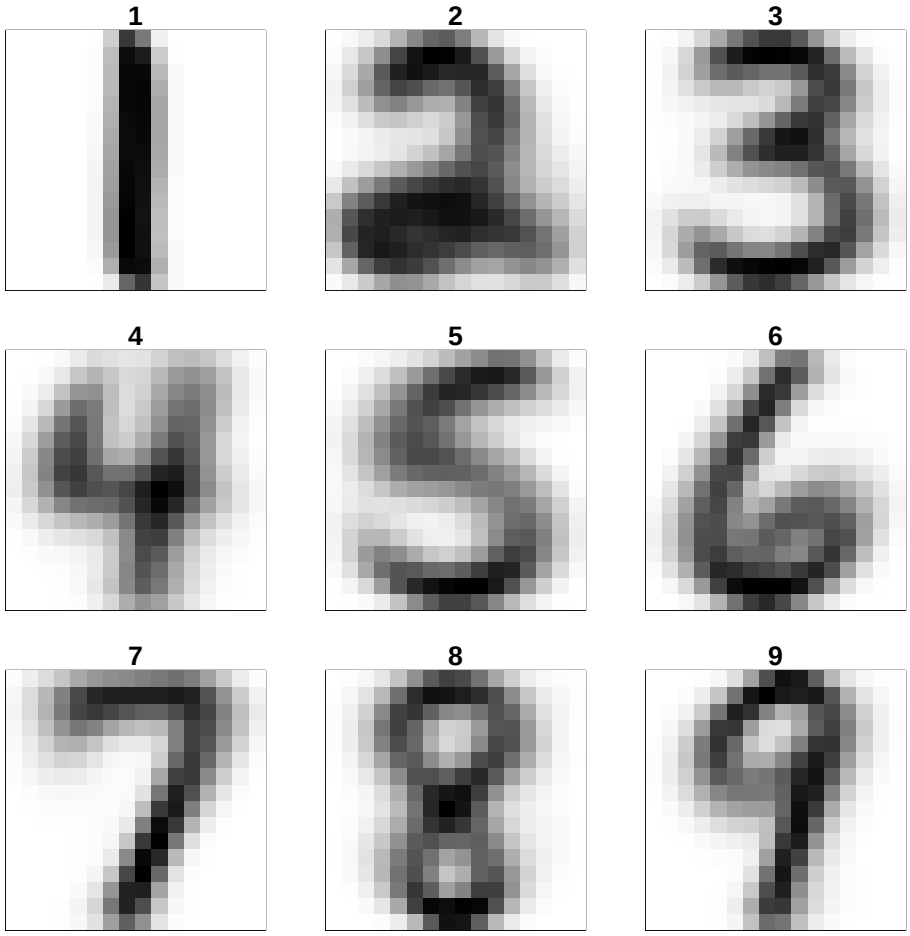
```
B <- t(sapply(1:(nrow(A)-1), function(x) {
  a <- c(rep(0,ncol(A)))
  a[x] = -1
  a[x+1] = 1
  return(a)
}))
B%*%A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    1    1    1    1
## [3,]    1    1    1    1
```

Problem2

```
zip <- readRDS('/Users/aaron/Desktop/R/midterm/zip.rds')
colors <- c('white','black')
cus_col <- colorRampPalette(colors=colors)

dig <- function(n){
  data <- array(filter(zip, V1 == n)[2:257])
  num <- array(colMeans(data), c(16,16))
  par(pty = "s", mar = c(1.2,0,1.2,0))
  image(num[,ncol(num):1], col = cus_col(256), main = n, tck = 0, xaxt = "n", yaxt = "n")
}

layout(matrix(seq(9), nrow=3, byrow=TRUE))
dig(1)
dig(2)
```

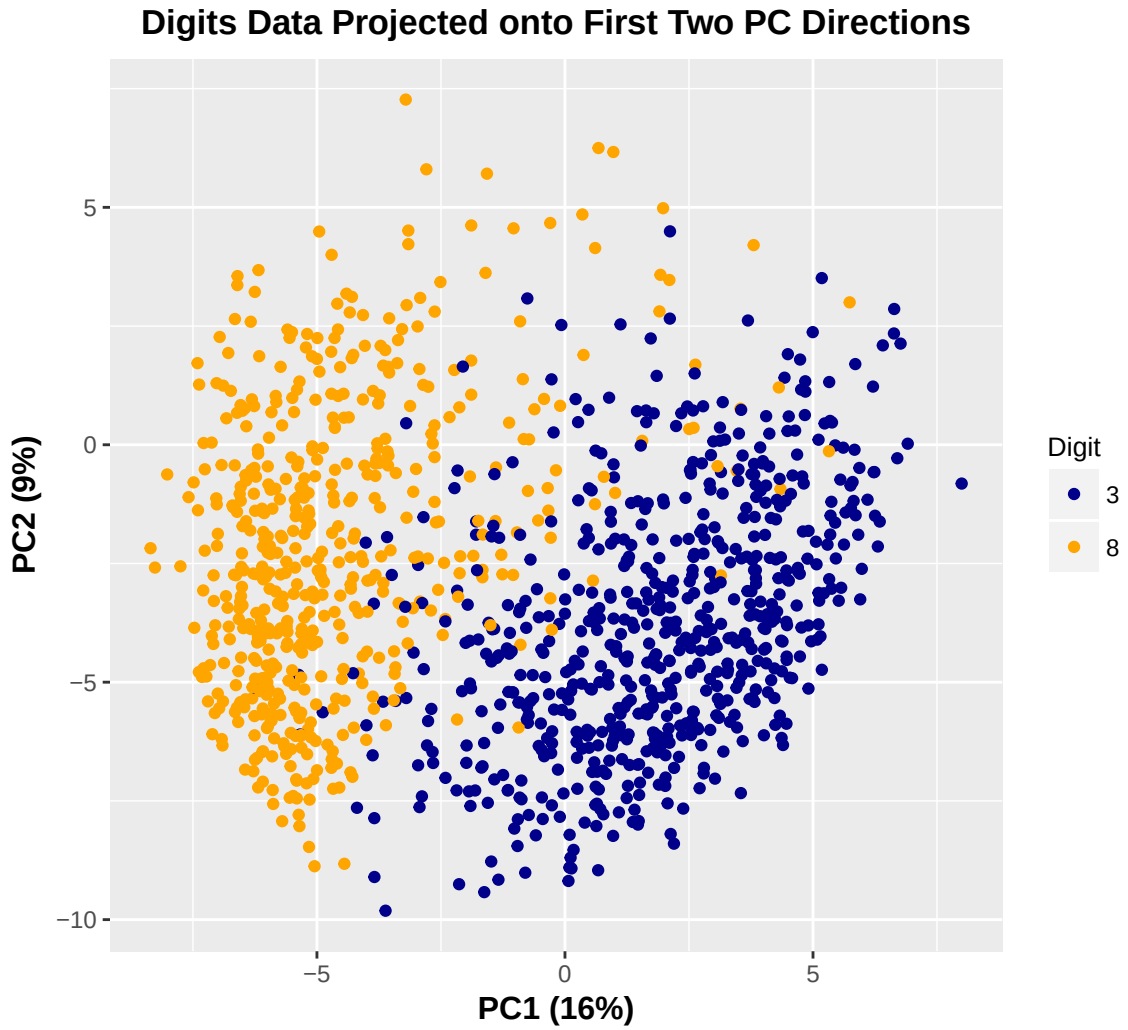
```
dig(3)
dig(4)
dig(5)
dig(6)
dig(7)
dig(8)
dig(9)
```



Problem3

```
zip <- readRDS('/Users/aaron/Desktop/R/midterm/zip.rds')
data <- data.matrix(subset(zip, V1 == 3 | V1 == 8)[2:257])
pca <- princomp(data)
prop.of.variance <- round(pca$sdev^2/sum(pca$sdev^2),2)
d <- data_frame(num = c(subset(zip, V1 == 3 | V1 == 8)[,1]),
                PC1 = c(data%*%pca$loadings[,1]),
                PC2 = c(data%*%pca$loadings[,2]))

ggplot(d)+
aes(x = PC1, y = PC2, colour = factor(num))+
geom_point()+
theme(
```

```
  plot.title = element_text(size = 13, hjust = 0.5, face = "bold"),
  axis.title = element_text(size = 12, face = "bold"),
  strip.text.x = element_text(size = 12, face = "bold"),
  legend.title = element_text(size = 10),
  legend.key.size = unit(0.7, "cm"),
  legend.text = element_text(size = 9),
  aspect.ratio = 1
)+
labs(
  x = paste("PC1 (",prop.of.variance[1]*100,"%)", sep = ""),
  y = paste("PC2 (",prop.of.variance[2]*100,"%)", sep = ""),
  shape = "Transmission"
)+
scale_colour_manual(
  values = c("darkblue", "orange"),
  name = "Digit",
  labels = c("3", "8")
)+
ggtitle("Digits Data Projected onto First Two PC Directions")
```



Problem4

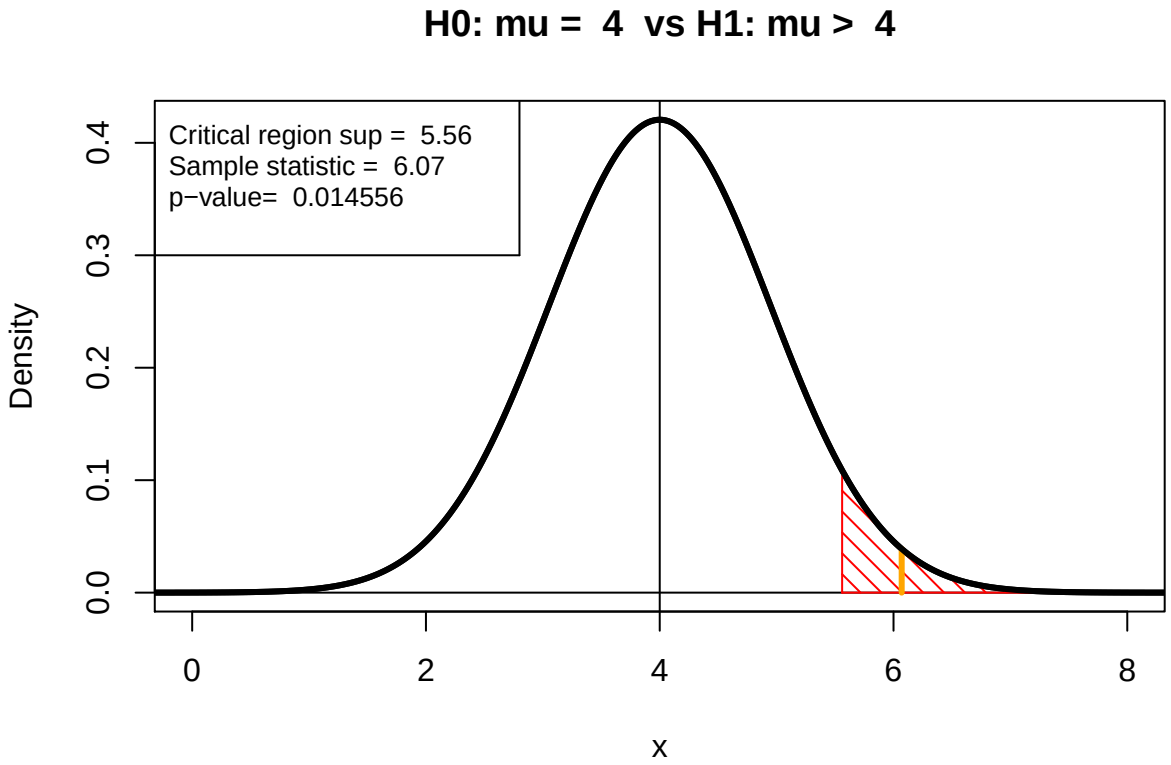
```
mu0 <- 4              ## Null hypothesis mean value
stdev <- 3            ## Known population standard deviation
signif.level <- 0.05  ## Test significance level
sample.mean <- 6.07   ## Mean of the random sample
n <- 10               ## Sample size
mu1 <- 6.2            ## Alternative hypotesis mean value to use for error type 2 and power


hyp.testing <- function(mu0, stdev, signif.level,
                        sample.mean, n,show_crit = T, show_pvalue = F ,
                        show_alt = F, mu1, show_beta = F,
                        show_power = F, two_sided = F) {

if(two_sided == F){
  cri_reg_sup <- qnorm(signif.level, mean = mu0, sd= sqrt(stdev^2/n), lower.tail = F)
  p_value <- pnorm(sample.mean, mean = mu0, sd= sqrt(stdev^2/n),lower.tail = F)
}
else{
  cri_reg_sup <- qnorm(signif.level/2, mean = mu0, sd = sqrt(stdev^2/n), lower.tail = F)
  cri_reg_inf <- mu0*2-cri_reg_sup
  p_value <- pnorm(sample.mean, mean = mu0, sd = sqrt(stdev^2/n), lower.tail = F)*2
}

up <- mu0+5*sqrt(stdev^2/n)
dn <- mu0-5*sqrt(stdev^2/n)
x <- seq(dn,up,length.out = 1000)
y <- dnorm(x, mean = mu0, sd= sqrt(stdev^2/n))
plot(x,y,type = "l", xlim = c(0,8), lwd=3, ylab="Density")
 if(two_sided == F){
   title(paste("H0: mu = ",mu0," vs H1: mu > ",mu0))
 }
 else{
   title(paste("H0: mu = ",mu0," vs H1: mu != ",mu0))
 }
abline(h=0)
abline(v=mu0)
  if(two_sided == F){
    text(-0.2,0.38,
    paste("Critical region sup = ",round(cri_reg_sup,2),"\nSample statistic = ",
          sample.mean,"\np-value= ", round(p_value,6)), cex=0.8,adj = 0)
  }
  else{
    text(-0.2,0.37,
    paste("Critical region sup = ",round(cri_reg_sup,3),"\nCritical region inf = ",
          round(cri_reg_inf,3),"\nSample statistic = ",sample.mean,"\np-value= ",
          round(p_value,6)),cex=0.8,adj = 0)
}
segments(-0.6,0.30,2.8,0.30)
segments(2.8,0.30,2.8,0.5)

#beta
```

```
if(show_beta == T){
x1 <- seq(1,11,length.out = 1000)
y1 <- dnorm(x1, mean = mu1, sd= sqrt(stdev^2/n))
lines(x1,y1,type = "l", xlim = c(0,8), lwd=1, ylab="Density")
xb <- seq(1, cri_reg_sup, length.out = 1000)
yb <- dnorm(xb, mean = mu1, sd= sqrt(stdev^2/n))
polygon(c(1,xb,cri_reg_sup), c(0,yb,0), col="blue")

  if(show_power == T){
  xb1 <- seq(cri_reg_sup, 11, length.out = 1000)
  yb1 <- dnorm(xb1, mean = mu1, sd= sqrt(stdev^2/n))
  polygon(c(cri_reg_sup,xb1,11), c(0,yb1,0), col="darkgreen")
  }
abline(v = mu1)
}


xc <- seq(cri_reg_sup,9,length.out = 1000)
yc <- dnorm(xc, mean = mu0, sd= sqrt(stdev^2/n))
polygon(c(cri_reg_sup,xc,9), c(0,yc,0),
        density = 13, angle = 135, col="red", lwd=1)
segments(sample.mean, 0, sample.mean,
         dnorm(sample.mean, mean = mu0, sd= sqrt(stdev^2/n)), col="orange", lwd=3)
if(two_sided == T){
  xct <- seq(-1,cri_reg_inf,length.out = 1000)
  yct <- dnorm(xct, mean = mu0, sd= sqrt(stdev^2/n))
  polygon(c(-1,xct,cri_reg_inf), c(0,yct,0),
          density = 13, angle = 135, col="red", lwd=1)
  segments(mu0*2-sample.mean, 0, mu0*2-sample.mean,
           dnorm(sample.mean, mean = mu0, sd= sqrt(stdev^2/n)), col="orange", lwd=3)
}

if(show_pvalue == T){
#p-value
xp <- seq(sample.mean, 9, length.out = 1000)
yp <- dnorm(xp, mean = mu0, sd= sqrt(stdev^2/n))
polygon(c(sample.mean,xp,9), c(0,yp,0),
        density = 13, angle = 45, col="orange", lwd=1)
if(two_sided == T){
  xpt <- seq(-1, mu0*2-sample.mean, length.out = 1000)
  ypt <- dnorm(xpt, mean = mu0, sd= sqrt(stdev^2/n))
  polygon(c(-1, xpt, mu0*2-sample.mean), c(0,ypt,0),
          density = 13, angle = 45, col="orange", lwd=1)
}
}
lines(x,y,type = "l", xlim = c(0,8), lwd=3, ylab="Density")



if(show_beta == T){
xt2 <- seq(0,8,length.out = 1000)
  if(two_sided == T){
    yt2 <- 1-pnorm(xt2, mean = cri_reg_inf, sd=sqrt(stdev^2/n))+
```

```
            pnorm(xt2, mean = cri_reg_sup, sd=sqrt(stdev^2/n))
  }
  else{yt2 <- pnorm(xt2, mean = cri_reg_sup, sd=sqrt(stdev^2/n))}
plot(xt2,1-yt2,type = "l", lwd=2, col="blue",
     xlab = "mu1",
     ylab = ifelse(show_power == F,"error type II", "error type II and power"))
  if(show_power == T){
  lines(xt2,yt2,type = "l", lwd=2, col="darkgreen")
  text(5.9,0.79,round(pnorm(mu1, mean = cri_reg_sup, sd=sqrt(stdev^2/n)),2),cex=1)
  }
abline(v=mu1)
text(5.9,0.23,1-round(pnorm(mu1, mean = cri_reg_sup, sd=sqrt(stdev^2/n)),2),cex=1)
}
}
```

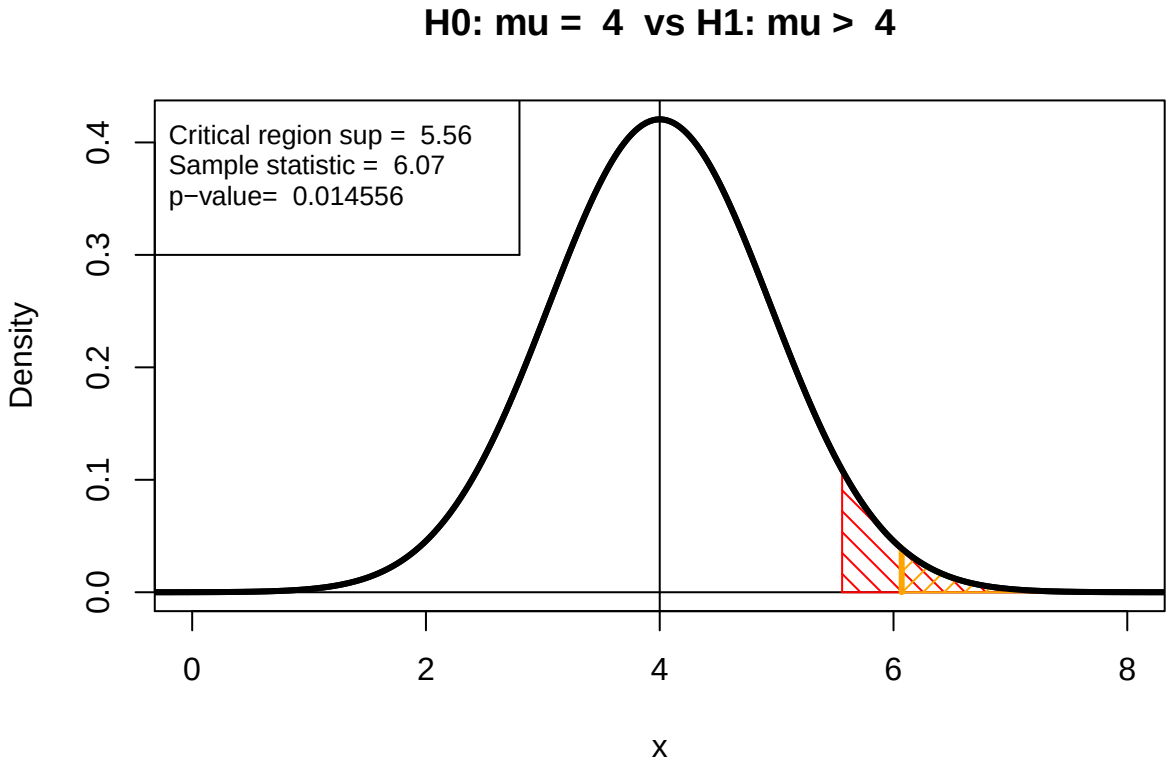
plot 1

```
hyp.testing(mu0 = 4, stdev = 3,signif.level=0.05,
            sample.mean = 6.07,n=10, mu1 = 6.2)
```



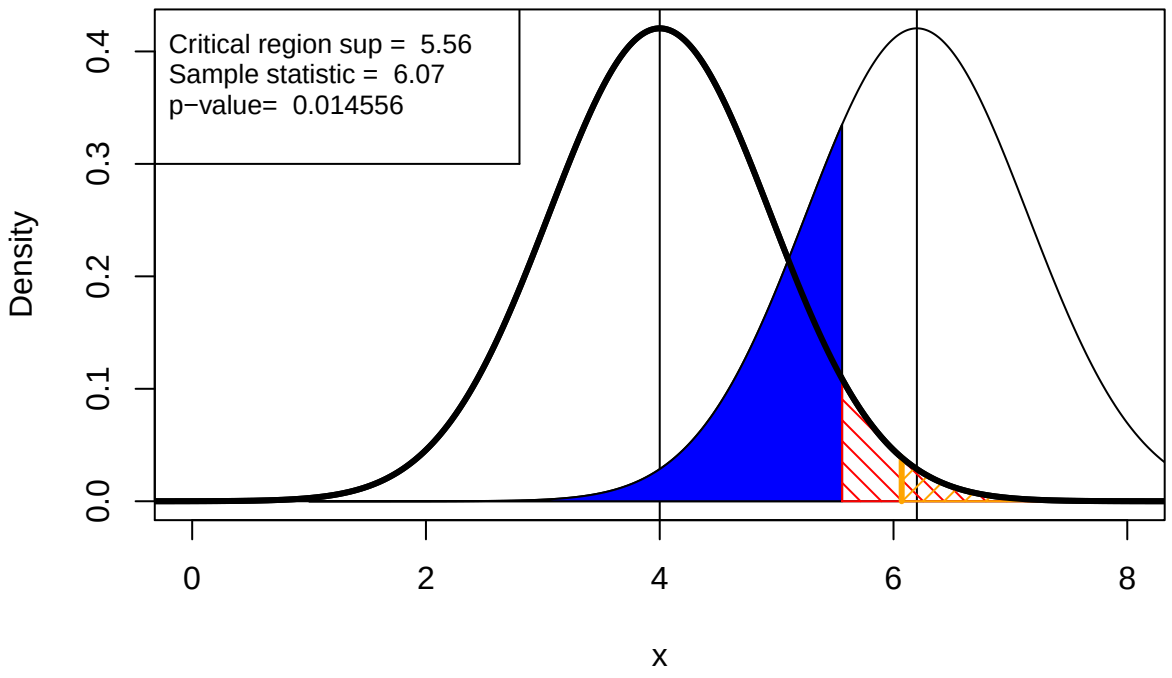
plot 2

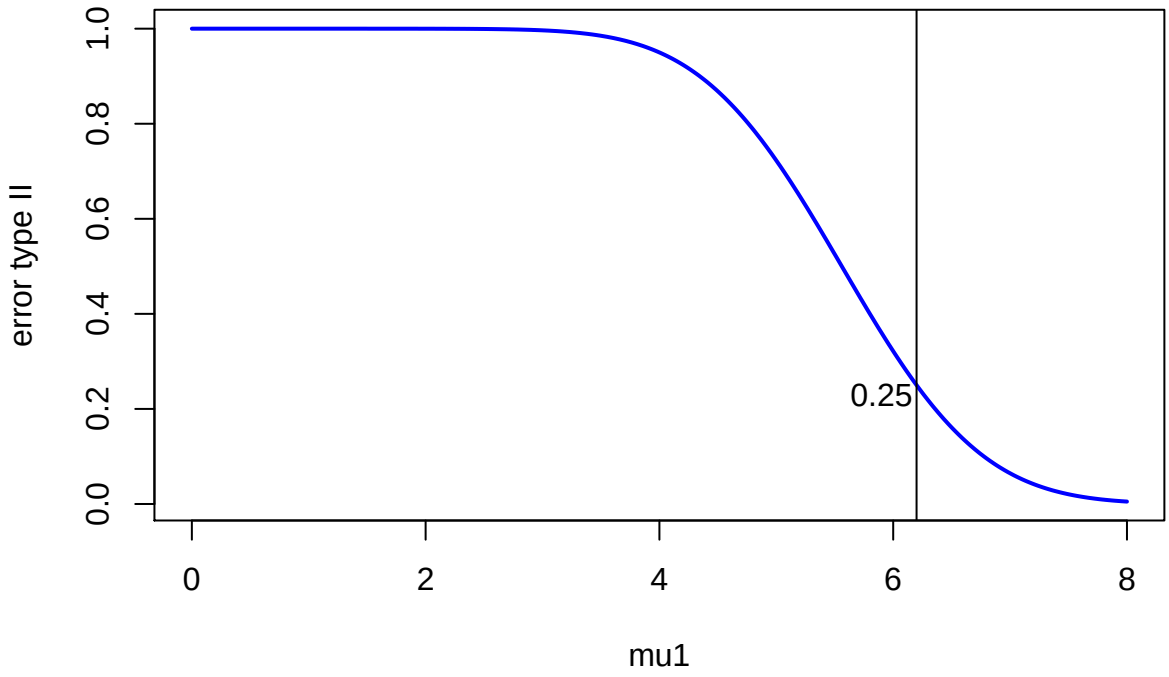
```
hyp.testing(mu0 = 4, stdev = 3,signif.level=0.05,
            sample.mean = 6.07,n=10, mu1 = 6.2, show_pvalue = T)
```



plot 3

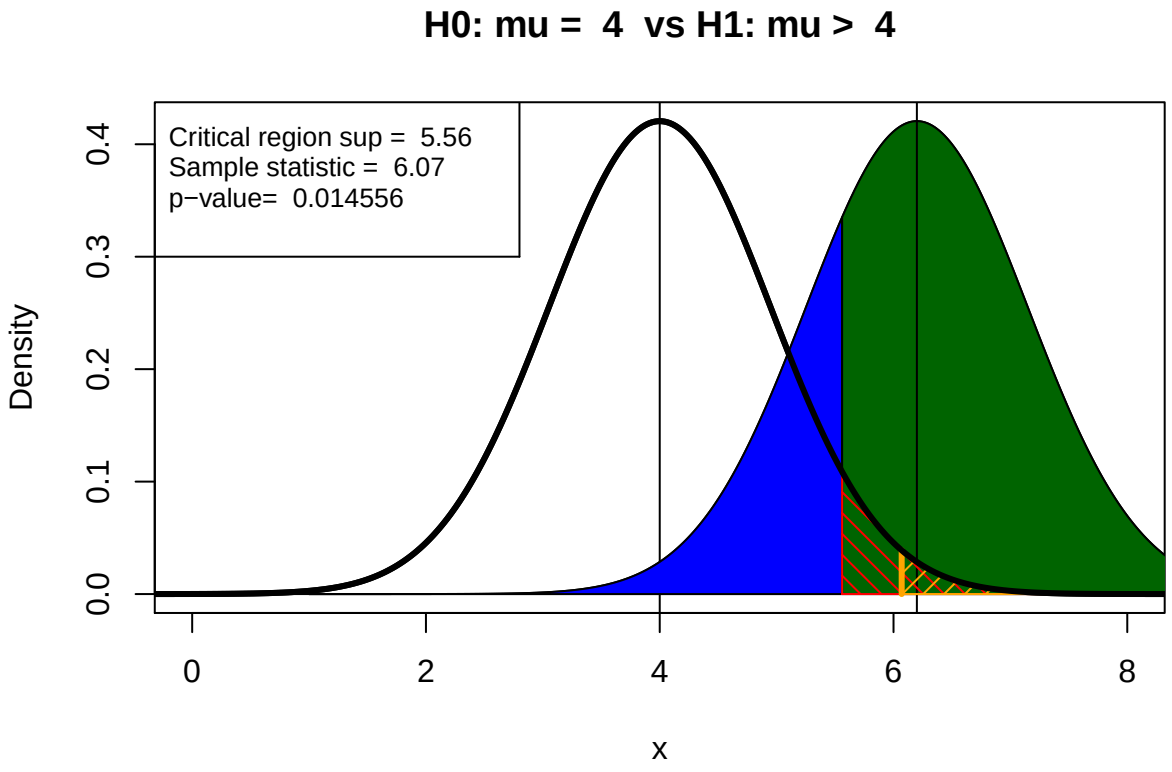
```
hyp.testing(mu0 = 4, stdev = 3,signif.level=0.05,
            sample.mean = 6.07,n=10, mu1 = 6.2, show_pvalue = T,
            show_beta = T)
```

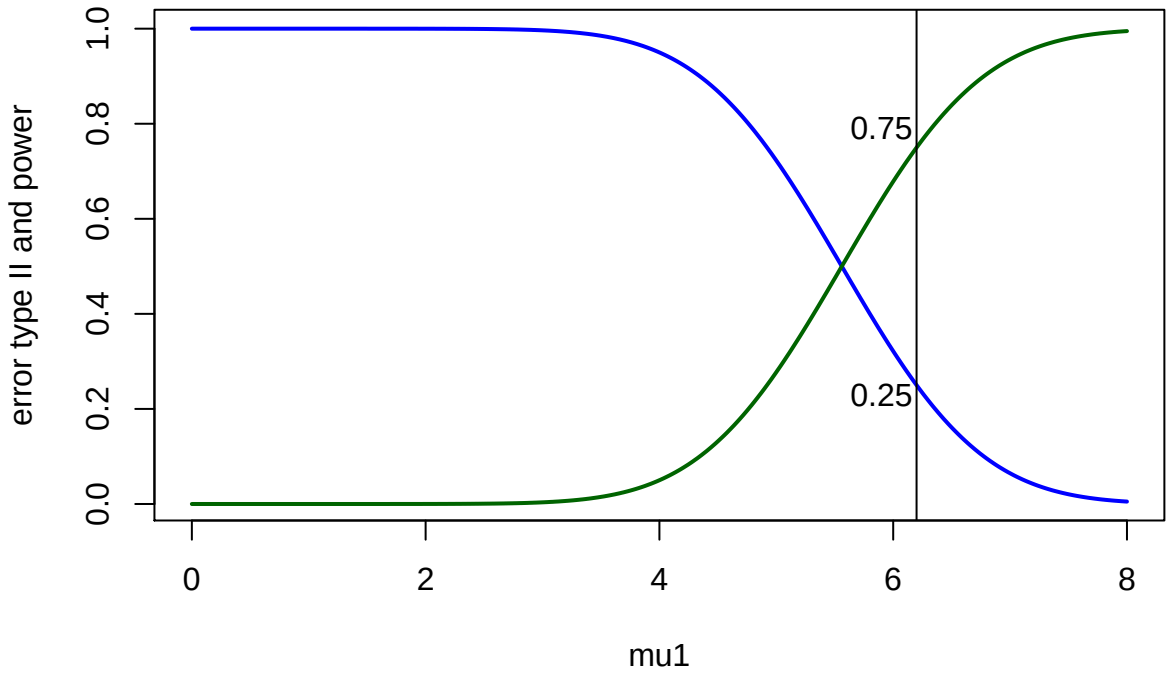




plot 4

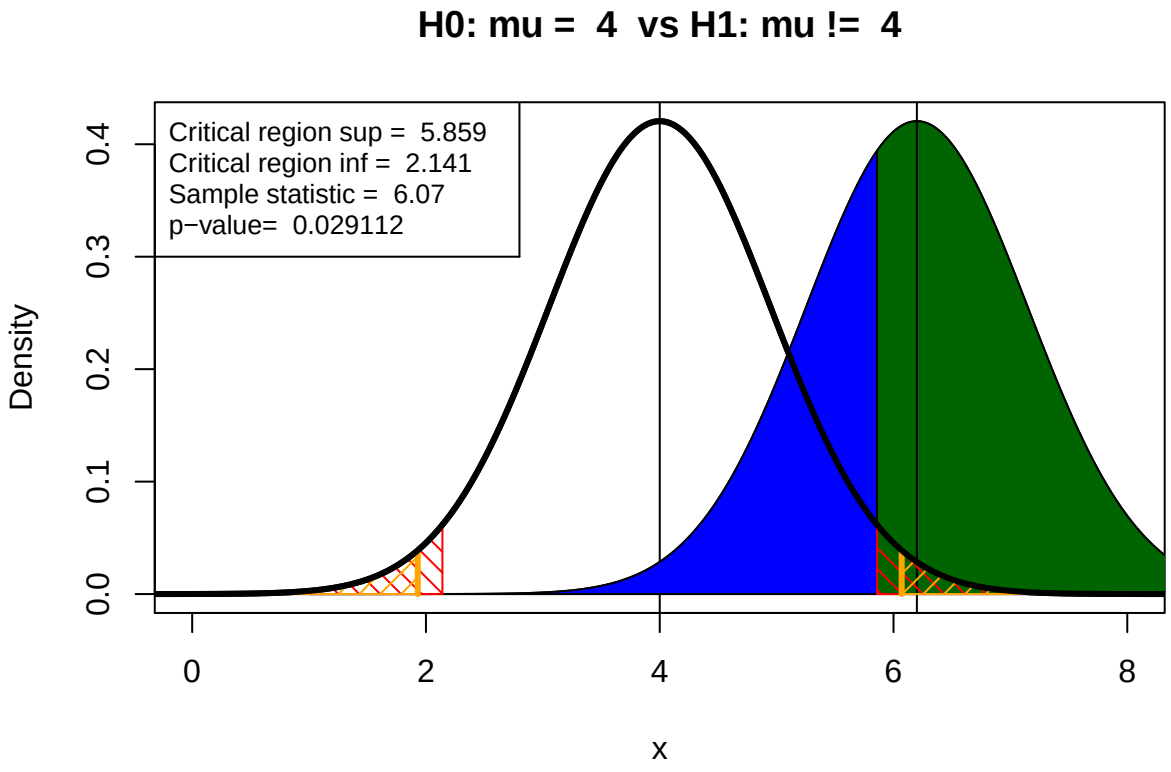
```
hyp.testing(mu0 = 4, stdev = 3,signif.level=0.05,
            sample.mean = 6.07,n=10, mu1 = 6.2, show_pvalue = T,
            show_beta = T, show_power = T)
```

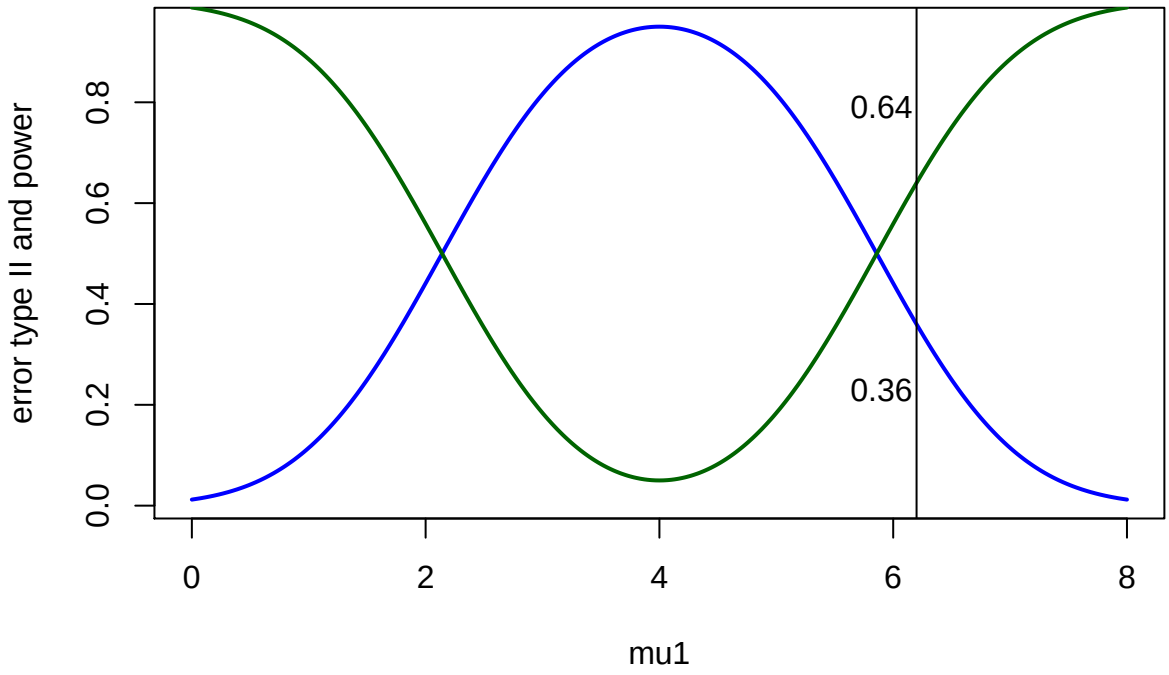




plot 5

```
hyp.testing(mu0 = 4, stdev = 3,signif.level=0.05,
            sample.mean = 6.07,n=10, mu1 = 6.2, show_pvalue = T,
            show_beta = T, show_power = T, two_sided = T)
```




error type II and power
0.0
0.2
0.4
0.6
0.8
0.64
0.36
0
2
4
6
8
mu1