

## 20210507

```
####2021/05/07####  
rm(list=ls())
```

這裡運用 sample 隨機產生 20 個樣本，並且定義為投手投球的球速，針對速度(speed)，探討平均速度大於 100，所以先定義出虛無假設(平均速度小於 100)與對立假設(平均速度超過 100)，先將此些條件新建出對應物件。

```
### 樣本設定 ###  
set.seed(12) #設計隨機編號  
ball_speed <- sample(50:150,20,replace = TRUE) #指派速度樣本  
ball_mean <- mean(ball_speed) #指派平均速度  
ball_n <- 20 #指派樣本數  
ball_sd <- sd(ball_speed) #指派速度標準差
```

運用 one.sample.z()函式，計算出其中 Z 值與 P-value 值，下面介紹此函數中個別參數意義。

```
###  
ball_z_test <- one.sample.z(ball_speed,      #放入預檢定資料集  
                             null.mu = 100,  #虛無假設  
                             xbar = ball_mean, #平均值  
                             sigma = ball_sd, #標準差  
                             alternative = "greater", #右尾  
                             conf = 0.95)    #信心水準  
  
ball_z_test  
  
##  
## One sample z-test  
##          z*      P-value  
## 1.785473 0.03709233
```

```
### 圖 ###  
sigma <- ball_sd #指派標準差  
mu0 <- 100 #虛無假設  
alpha <- ball_z_test$test$`P-value` #選取之前計算出來的p-value  
xLims <- c(0, 200) #定義X 座標軸  
crit <- qnorm(1-alpha, mu0, sigma)  
crit2 <- qnorm(0.95, mu0, sigma) #算出臨界值
```

```

#從crit 切出100 位置
right <- seq(crit, xLims[2], length.out=100)

#利用常態分配分別以剛剛切出100 位置分別計算出Y 軸位置
yH0r <- dnorm(right, mu0, sigma)

#利用算出來的平均值與標準差，畫出常態分配
curve(dnorm(x, mu0, sigma),
      xlim=xLims, lwd=2, col="red", xlab="x", ylab="density",
      main="Normal distribution", ylim=c(0, 0.018), xaxs="i")

#以臨界值向右塗上陰影
polygon(c(right, rev(right)),
        c(yH0r, numeric(length(right))), border=NA,
        col=rgb(1, 0.3, 0.3, 0.6))

#剛剛上色的部分為計算出來的p-value 值
text(crit+5, 0.0015, adj = 0, label="p-value")

#利用藍色線標記出計算出來的p-value 值對應的值
abline(v=crit, lty=1, lwd=3, col="blue")

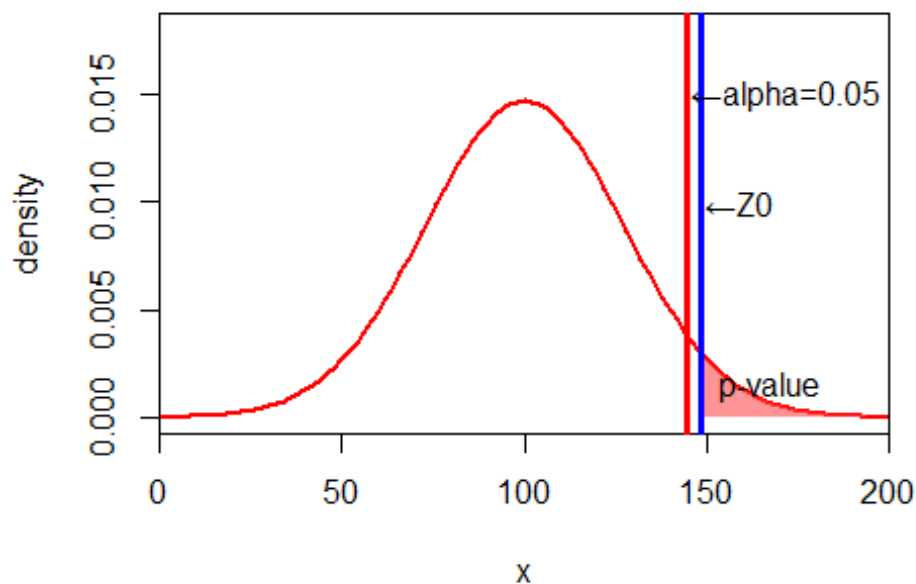
#標記此線段為計算出來的p-value 值對應的值
text(crit+1, 0.010, adj=0, label="←Z0")

#利用紅色線段標記95%信心對應X 軸的臨界值
abline(v=crit2, lty=1, lwd=3, col="red")

#標記此線段為95%信心對應X 軸的臨界值
text(crit2+1, 0.015, adj=0, label="←alpha=0.05")

```

## Normal distribution



### 檢定力 ###

```
mu1 <- 130 #假設樣本平均數130 為母體平均數
pow <- pnorm(crit, mu1, sigma) #檢定力(power)
beta <- 1-pow #反之用1 相減可算出所求 beta
```

#以臨界值到X 軸原點切出100 位置

```
left <- seq(xLims[1],
            crit, length.out=100)
```

#利用常態分配分別以剛剛切出100 位子分別計算出Y 軸位置

```
yH11 <- dnorm(left, mu1, sigma)
```

#利用算出來的平均值與標準差，畫出常態分配

```
curve(dnorm(x, mu0, sigma),
      xlim=xLims, lwd=2, col="red", xlab="x", ylab="density",
      main="Normal distribution", ylim=c(0, 0.018), xaxs="i")
```

#利用新的平均值與標準差，畫出常態分配

```
curve(dnorm(x, mu1, sigma),
      lwd=2, col="blue", add=TRUE)
```

#以臨界值向右塗上陰影

```
polygon(c(right, rev(right)),
        c(yH0r, numeric(length(right))), border=NA,
```

```

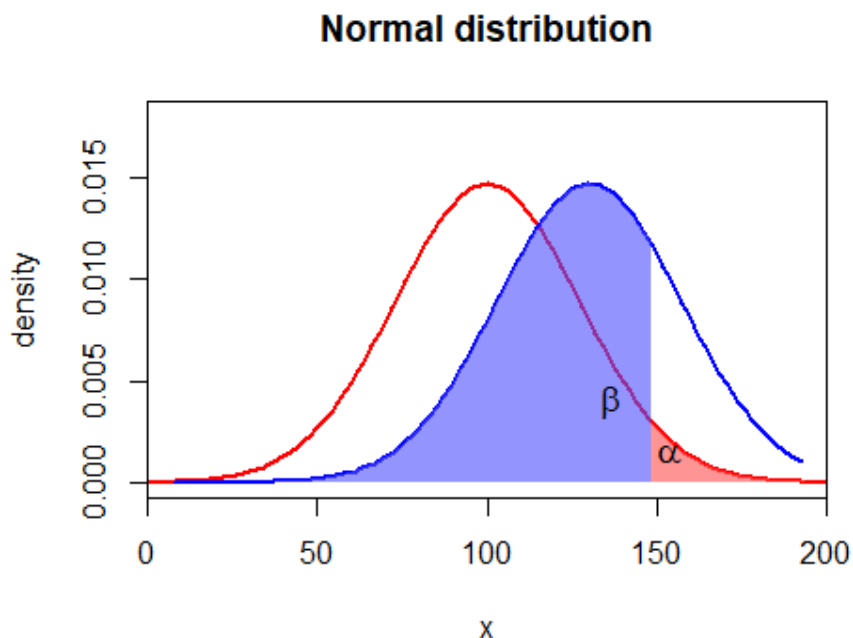
col=rgb(1, 0.3, 0.3, 0.6))

#以臨界值向左塗上陰影
polygon(c(left, rev(left)),
       c(yH11, numeric(length(left))), border=NA,
       col=rgb(0.3, 0.3, 1, 0.6))

#標記出  $\beta$ 
text(crit-12, 0.004,
     expression(beta), cex=1.2)

#標記出  $\alpha$ 
text(crit+5, 0.0015,
     expression(alpha), cex=1.2)

```



利用上面的是視覺化結果可以幫助理解在虛無假設下，拒絕域在什麼位置，並且對應的臨界值為多少，對於視覺化有興趣的同學可以試著操作左尾檢定與雙尾檢定如何運用視覺化來呈現。

注意：若是使用 mac 的同學，請去以下網址下載 XQuartz 套件：

<https://www.xquartz.org/>

再將電腦重新啟動，即可使用。