Normal Distribution Sampling

Tarek El-Hajjaoui

2023-04-13

```
library(tidyr) # pipe (%>%) tool
library(MASS) # murnorm()
library(resample) # colStdevs()
```

Q1

Suppose Suppose X_1, X_2, Y_1, Y_2 are mutually independent.

- X_1 and X_2 are iid from $N(\mu = 0, \sigma^2 = 2^2)$
- Y_1 and Y_2 are iid from $N(\mu = 0, \sigma^2 = 1^2)$

a) Calculate $P(|X_1 - X_2| > 4)$

- Let $X = |X_1 - X_2|$ then $P(|X| > 4) \sim N(0, 8)$
- Transform X -> Z_X then $P(|Z_X| > 4/\sqrt{8}) \sim N(0, 1)$
- Calculation in R below:

```
2 * (1 - pnorm(4/sqrt(8)))
```

```
## [1] 0.1572992
```

b) Calculate $P(|Y_1 - Y_2| > 4)$

- Let $Y = Y_1 - Y_2$ then $P(|Y| > 4) \sim N(0, 2)$
- Transform Y -> Z_y then $P(|Z_Y| > 4/\sqrt{2}) \sim N(0, 1)$
- Calculation of in R below:

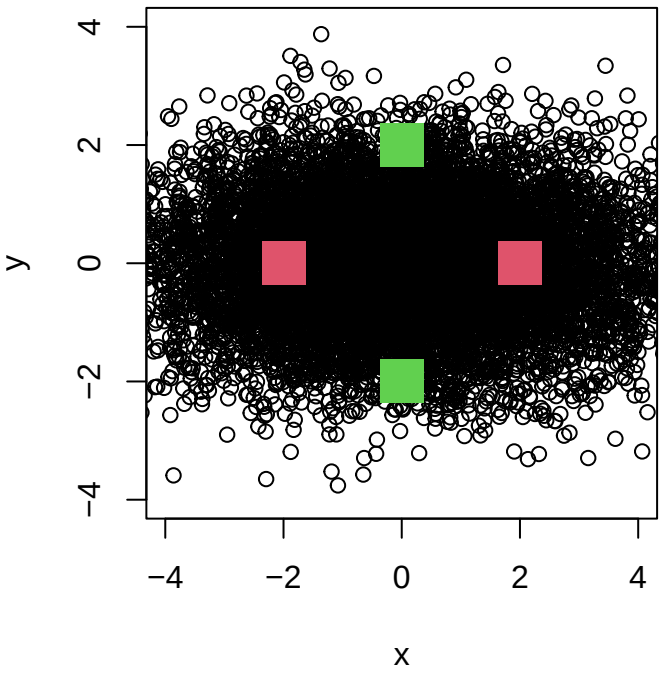
```
2 * (1 - pnorm(4/sqrt(2)))
```

```
## [1] 0.004677735
```

Q1 continued

c) Estimate the two probabilities using simulations

```
set.seed(20230404)
n <- 10000 # number of samples
bivariate_mu <- c(0,0) # bivariate normal mu
cov_matrix <- matrix(c(4,0,0,1),2,2) # bivariate var
# bivariate normal sample
sample <- mvrnorm(n=n, mu=bivariate_mu, Sigma=cov_matrix)
x <- sample[, 1] # Extract the X values, X ~ N(0, 4)
y <- sample[, 2] # Extract the Y values, X ~ N(0, 1)
par(pty="s") #to make sure the shape of figure is a square
sample %>%
  plot(xlab="x", ylab="y", xlim=c(-4,4), ylim=c(-4,4))
points(x=c(-2, 0, 0, 2), y=c(0, -2, 2, 0), pch=15, col=c(2,3,3,2),cex=3)
```



Calculate $P(|X| > 2)$ and calculate $P(|Y| > 2)$

```
# Calculate the proportion of pairs satisfying |X| > 2
X1 <- rnorm(n=n, mean = c(0,0), sd = 2)
X2 <- rnorm(n=n, mean = c(0,0), sd = 2)
p1 <- sum(abs(X1-X2)>4)/n
p1
```

```
## [1] 0.1584
```

```
# Calculate the proportion of pairs satisfying |Y| > 2
Y1 <- rnorm(n=n, mean = c(0,0), sd = 1)
Y2 <- rnorm(n=n, mean = c(0,0), sd = 1)
p2 <- sum(abs(Y1-Y2)>4)/n
p2
```

```
## [1] 0.0043
```

Q2

a) Find a matrix A such that AY gives the difference of mean vectors between iris setosa and iris versicolor

```
# Extracting the continuous feature columns from the dataset
Y=as.matrix(iris[, 1:4])
dim(Y)
```

```
## [1] 150   4
```

```
A=1/50*rep(c(1, -1, 0), each=50)
A
```

```
##   [1]  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02
##  [13]  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02
##  [25]  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02
##  [37]  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02
##  [49]  0.02  0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02
##  [61] -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02
##  [73] -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02
##  [85] -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02
##  [97] -0.02 -0.02 -0.02 -0.02  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
## [109]  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
## [121]  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
## [133]  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
## [145]  0.00  0.00  0.00  0.00  0.00  0.00
```

```
AY <- A%*%Y
AY
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## [1,]        -0.93       0.658       -2.798       -1.08
```

```
# Comparing AY with the colMean() method
colMeans(Y[1:50, ]) -colMeans(Y[51:100, ])
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##       -0.930        0.658       -2.798       -1.080
```

Q2 continued

b) Find a matrix B such that YB is column-standardized, i.e., the standard deviation of each column/feature is 1.

```
# Matrix B such that YB is column-standardized
B <- diag(1/colStdevs(Y))
dim(B)
```

```
## [1] 4 4
```

```
colVars(Y%*%B)
```

```
## [1] 1 1 1 1
```

c) Check the following

- Let $C = \mathbf{I}_{150} - \frac{1}{150}J$, where $J_{150\times 150}$ is an all-ones matrices . Use R to verify that CY centers each column/feature.

```
C=diag(1,150) - (1/150) * matrix(1, 150, 150)
dim(C)
```

```
## [1] 150 150
```

```
CY <- C %*% Y
dim(CY)
```

```
## [1] 150   4
```

```
# Compute column means of CY
col_means_CY <- colMeans(CY)
col_means_CY
```

```
##  Sepal.Length   Sepal.Width  Petal.Length   Petal.Width
##  5.799065e-16 -1.286471e-15 -7.956598e-17 -5.268933e-16
```

```
# Verify that column means of CY are 0
all.equal(col_means_CY, rep(0, 4), check.names = FALSE)
```

```
## [1] TRUE
```

Q2 continued

d) Let S be the sample covariance matrix. Use R to verify that each column of $CYS^{-1/2}$ has been centered and standardized (in fact, the columns have also been de-correlated).

```
Y <- as.matrix(iris[, 1:4])

# Compute the sample covariance matrix S
S <- cov(Y)

# Using eigen decomposition of the covariance matrix S
eig <- eigen(S)
D <- diag(1/sqrt(eig$values))
# Compute S^(-1/2)
S_inv_sqrt <- eig$vectors %*% D %*% t(eig$vectors)

#S_inv_sqrt <- calc_invsqrt_mat(S)
CYS_inv_sqrt <- C %*% Y %*% S_inv_sqrt

colMeans(CYS_inv_sqrt)
```

```
## [1]  2.691447e-15 -4.232170e-15 -9.582844e-16 -1.500466e-15
```

```
# Verify that column means of CY are 0
all.equal(colMeans(CYS_inv_sqrt), rep(0, 4), check.names = FALSE)
```

```
## [1] TRUE
```

```
apply(CYS_inv_sqrt, 2, sd)
```

```
## [1] 1 1 1 1
```

```
# Verify that column means of CY are 1
all.equal(apply(CYS_inv_sqrt, 2, sd), rep(1, 4), check.names = FALSE)
```

```
## [1] TRUE
```

Q3

Choose a picture you like and conduct approximations using singular value decomposition (SVD).

```
library(jpeg)
img=readJPEG("./lion.jpg", native = FALSE)
img=img[,,1]
dim(img)
```

```
## [1] 524 800
```

```
par(mfrow=c(2,2))

image(img, col=gray(c(0:10)/10))
obj.bm=svd(img)
n=10
img.approx=obj.bm$u[,1:n]%*% diag(obj.bm$d[1:n]) %*% t(obj.bm$v[,1:n])
image(img.approx, main="rank 10", col=gray(c(0:10)/10))
n=20
img.approx=obj.bm$u[,1:n]%*% diag(obj.bm$d[1:n]) %*% t(obj.bm$v[,1:n])
image(img.approx, main="rank 20", col=gray(c(0:10)/10))
n=30
img.approx=obj.bm$u[,1:n]%*% diag(obj.bm$d[1:n]) %*% t(obj.bm$v[,1:n])
image(img.approx, main="rank 30", col=gray(c(0:10)/10))
```