Normal Distribution Sampling

Tarek El-Hajjaoui

2023-04-13

```
library(tidyr) #the pipe (%>%) tool is extremely useful
library(MASS) # used for mvrnorm
```

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 $P(|X_1 - X_2| > 2)$

- Let $X = X_1 - X_2$ then $P(X > 2) \sim N(0, 8)$
- Transform X -> Z_x then $P(Z_X > 2/\sqrt{8}) \sim N(0, 1)$
- Calculation of $1 - P(Z_X < 2/\sqrt{8})$ in R below:

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

```
## [1] 0.2397501
```

b) Calculate $P(|Y_1 - Y_2| > 2)$

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

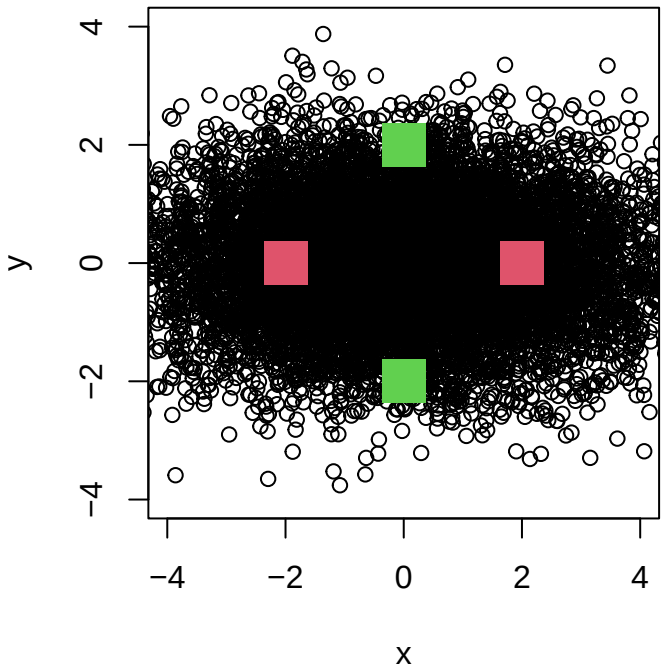
```
1 - pnorm(2/sqrt(2))
```

```
## [1] 0.0786496
```

Q1 continued

c) Estimate the two probabilities using simulations. The code in the previous page generates 1000 random samples. Change the sample size from n=1000 to n=10000 and then estimate the two probabilities. To do that, you need to examine all pairs of data points and then calculate the proportion of pairs satisfying a certain condition.

```
set.seed(20230404)
n <- 10000 # number of samples
# bivariate normal random sample parameters
bivariate_mu <- c(0,0)
cov_matrix <- matrix(c(4,0,0,1),2,2)
# bivariate normal sample
sample <- mvrnorm(n=n, mu=bivariate_mu, Sigma=cov_matrix)
# Extract the X values, X ~ N(0, 4)
x <- sample[, 1]
# Extract the Y values, X ~ N(0, 1)
y <- sample[, 2]
# plot
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
mean(x > 2)
```

```
## [1] 0.1589
```

```
# Calculate the proportion of pairs satisfying Y > 2
mean(y > 2)
```

```
## [1] 0.0237
```