Workshop 1

Aprendizaje Estadístico 2

Solve the next exercises using computational tools like R or Python, R Mardown or Google Colab. Make a pdf report where you show the whole procedure. Take into account that you have to be clear because it is something to be assesed.

```
knitr::opts_chunk$set(echo = TRUE, message = F, warning = F)
```

```
library(tidyverse)
```

Experiment Setup

From exercise 1 to 5 consider the sample space of tossing a fair coin 3 times:

$$S = \{HHH, HHT, HTH, THH, HTT, THT, TTH, TTT\}$$

```
# Espacio muestral
S <- c("HHH","HHT","HTH","THH","HTT","THT","TTH","TTT")
```

with each outcome having probability 1/8.

1. Describe the next events:

A : "first toss is H."

```
A <- c("HHH","HHT","HTH","HTT")
A
```

```
[1] "HHH" "HHT" "HTH" "HTT"
```

B : "second toss is H."

```r
B <- c("HHH","HHT","THH","THT")
B
```

```
[1] "HHH" "HHT" "THH" "THT"
```

C : "all tosses are H."

```r
C <- c("HHH")
C
```

```
[1] "HHH"
```

D : "exactly one H."

```r
D <- c("HTT","THT","TTH")
D
```

```
[1] "HTT" "THT" "TTH"
```

2. Check if A and B are independent. Are they mutually exclusive?

```r
# Tamaños
n <- length(S)
PA <- length(A) / n
PB <- length(B) / n

# Intersección
interseccion <- intersect(A, B)
PAB <- length(interseccion) / n

# Independencia: P(A   B) == P(A) * P(B)
independientes <- (PAB == PA * PB)

# Mutuamente excluyentes: P(A   B) == 0
mutuamente_excluyentes <- (PAB == 0)

# Resultados
cat("P(A) =", PA, "\n")
```

```
P(A) = 0.5
```

```r
cat("P(B) =", PB, "\n")
```

```
P(B) = 0.5
```

```r
cat("P(A   B) =", PAB, "\n")
```

```
P(A   B) = 0.25
```

```r
cat("¿Independientes? ->", independientes, "\n")
```

```
¿Independientes? -> TRUE
```

```r
cat("¿Mutuamente excluyentes? ->", mutuamente_excluyentes, "\n")
```

```
¿Mutuamente excluyentes? -> FALSE
```

3. Check if C and D are independent. Are they mutually exclusive?

```r
# Tamaños
n <- length(S)
PC <- length(C) / n
PD <- length(D) / n

# Intersección
interseccion <- intersect(C, D)
PCD <- length(interseccion) / n

# Independencia: P(C   D) == P(C) * P(D)
independientes <- (PCD == PC * PD)

# Mutuamente excluyentes: P(C   D) == 0
mutuamente_excluyentes <- (PCD == 0)

# Resultados
cat("P(C) =", PC, "\n")
```

```
P(C) = 0.125
```

```r
cat("P(D) =", PD, "\n")
```

```
P(D) = 0.375
```

```r
cat("P(C   D) =", PCD, "\n")
```

```
P(C   D) = 0
```

```
cat("¿Independientes? ->", independientes, "\n")
```

```
¿Independientes? -> FALSE
```

```
cat("¿Mutuamente excluyentes? ->", mutuamente_excluyentes, "\n")
```

```
¿Mutuamente excluyentes? -> TRUE
```

4. Prove: If two events E, F are mutually exclusive and both have positive probability, then they cannot be independent.
5. Take event G = "first toss is head," G^C = "first toss is tail." Show G and G^C are mutually exclusive and compute whether they are independent.

```
G <- c("HHH","HHT","HTH","HTT")
G
```

```
[1] "HHH" "HHT" "HTH" "HTT"
```

```
Gc <- c("THH","THT","TTH","TTT")
Gc
```

```
[1] "THH" "THT" "TTH" "TTT"
```

```
# Probabilidades
P_G <- length(G) / length(S)
P_Gc <- length(Gc) / length(S)
P_intersection <- length(intersect(G, Gc)) / length(S)

cat("P(G) =", P_G, "\n")
```

```
P(G) = 0.5
```

```
cat("P(Gc) =", P_Gc, "\n")
```

```
P(Gc) = 0.5
```

```
cat("P(G   Gc) =", P_intersection, "\n")
```

```
P(G   Gc) = 0
```

```
# Verificar mutuamente excluyentes
if (P_intersection == 0) {
  cat("G y Gc son mutuamente excluyentes.\n")
} else {
  cat("G y Gc NO son mutuamente excluyentes.\n")
}
```

```
G y Gc son mutuamente excluyentes.
```

```r
# Verificar independencia
if (P_intersection == P_G * P_Gc) {
  cat("G y Gc son independientes.\n")
} else {
  cat("G y Gc NO son independientes.\n")
}
```

```
G y Gc NO son independientes.
```

6. Compute $P(Z \geq 1.96)$ for a standard normal $Z \sim \mathcal{N}(0, 1)$ and illustrate it by shading the right tail.

Using pnorm(), compute $P(Z \geq 1.96)$.

Make a plot of the standard normal density on $x \in [-4, 4]$ with the right tail ($x \geq 1.96$) shaded.

```r
# Probabilidad P(Z >= 1.96)
p <- 1 - pnorm(1.96)
p
```

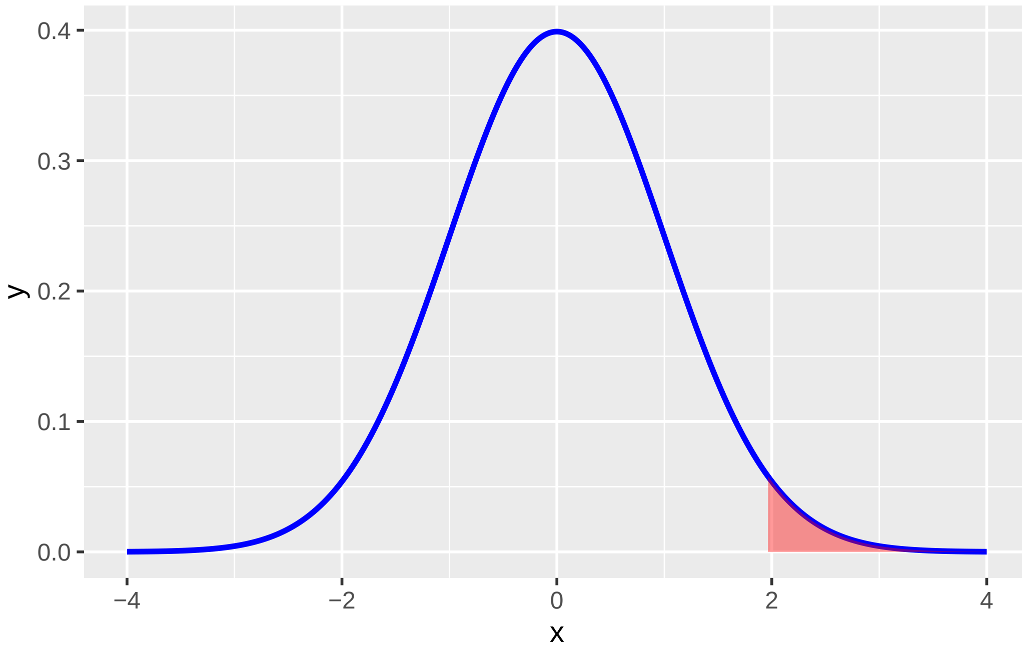
```
[1] 0.0249979
```

```r
# Datos para la curva normal estándar
x <- seq(-4, 4, length = 1000)
y <- dnorm(x)

df <- data.frame(x, y)

# Cola a sombrear (x >= 1.96)
df_tail <- subset(df, x >= 1.96)

ggplot(df, aes(x, y)) +
  geom_line(color = "blue", size = 1) +
  geom_area(data = df_tail, aes(x, y), fill = "red", alpha = 0.4)
```



7. For $Z \sim \mathcal{N}(0, 1)$ compute:

(a) $P(Z \leq -1.5)$

```
pnorm(-1.5)
```

```
[1] 0.0668072
```

(b) $P(Z \geq 2.1)$

```
1 - pnorm(2.1)
```

```
[1] 0.01786442
```

(c) $P(|Z| \geq 2)$

```
# P(|Z| >= 2) = P(Z <= -2) + P(Z >= 2)
pnorm(-2) + (1 - pnorm(2))
```

```
[1] 0.04550026
```

(d) The central 95% probability $P(-z \leq Z \leq z)$ and the value of z such that this equals 0.95.

```
qnorm(0.95)
```

```
[1] 1.644854
```

8. dd

   (a) Find the 90th percentile of $Z \sim \mathcal{N}(0, 1)$, i.e., compute $z_{0.90}$ such that $P(Z \leq z_{0.90}) = 0.90$.

   (b) Find z such that $P(|Z| \leq z) = 0.90$.

   (c) Suppose $X \sim \mathcal{N}(\mu = 100, \sigma = 15)$. Compute the score threshold c for the top 5%.

9. For $Z \sim \mathcal{N}(0, 1)$:

(a) Simulate $N = 100{,}000$ draws and estimate $P(Z \geq 1.96)$ empirically.

```r
Z <- rnorm(100000, mean = 0, sd = 1)
mean(Z >= 1.96)
```

```
[1] 0.02492
```

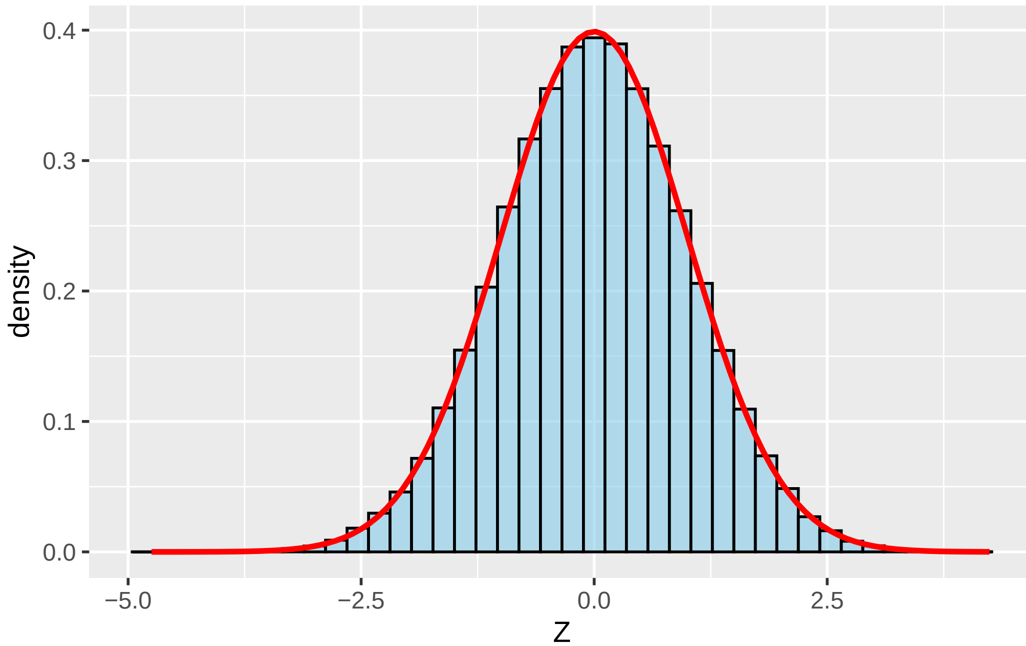
(b) Compare with the exact value.

```r
1 - pnorm(1.96, mean = 0, sd = 1)
```

```
[1] 0.0249979
```

(c) Plot a histogram with an overlaid theoretical density curve.

```r
ggplot(data.frame(Z), aes(x = Z)) +
  geom_histogram(aes(y = ..density..), bins = 40, fill = "skyblue", color = "black", alp
  stat_function(fun = dnorm, args = list(mean = 0, sd = 1), color = "red", size = 1)
```



10. Let $X \sim \mathcal{N}(\mu = 70,\ \sigma = 8)$. Compute:

(a) $P(X \le 60)$

(b) $P(65 \le X \le 85)$

(c) The 2.5th percentile of X

Do each both directly and by standardizing to $Z = \dfrac{X - \mu}{\sigma}$.

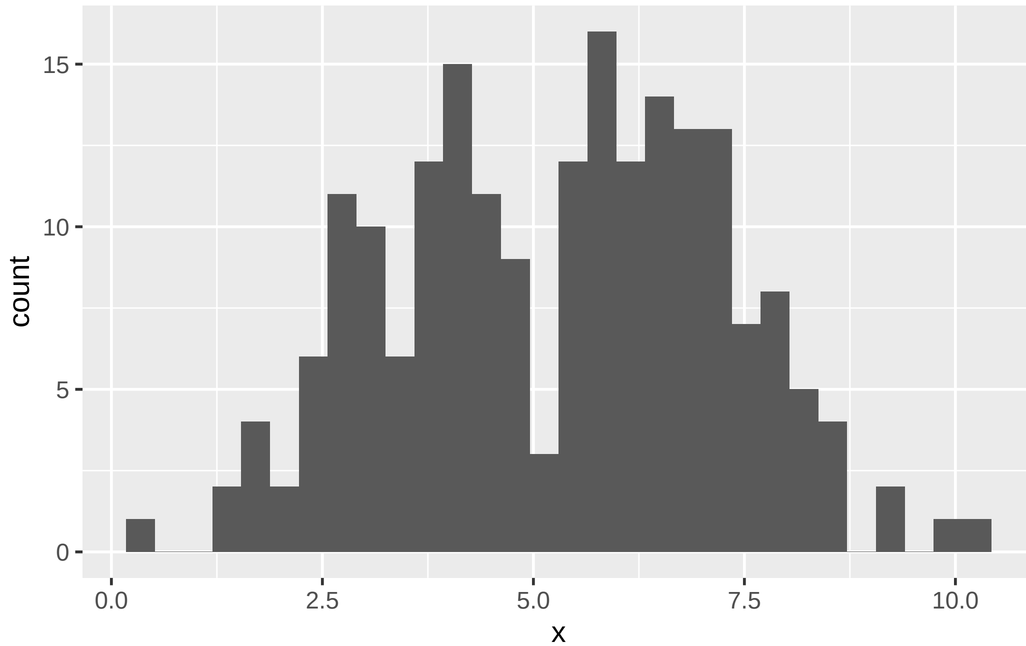
q <- 1-p

11. Generate $N = 200$ observations from $\mathcal{N}(\mu = 5, \sigma^2 = 2^2)$, then:

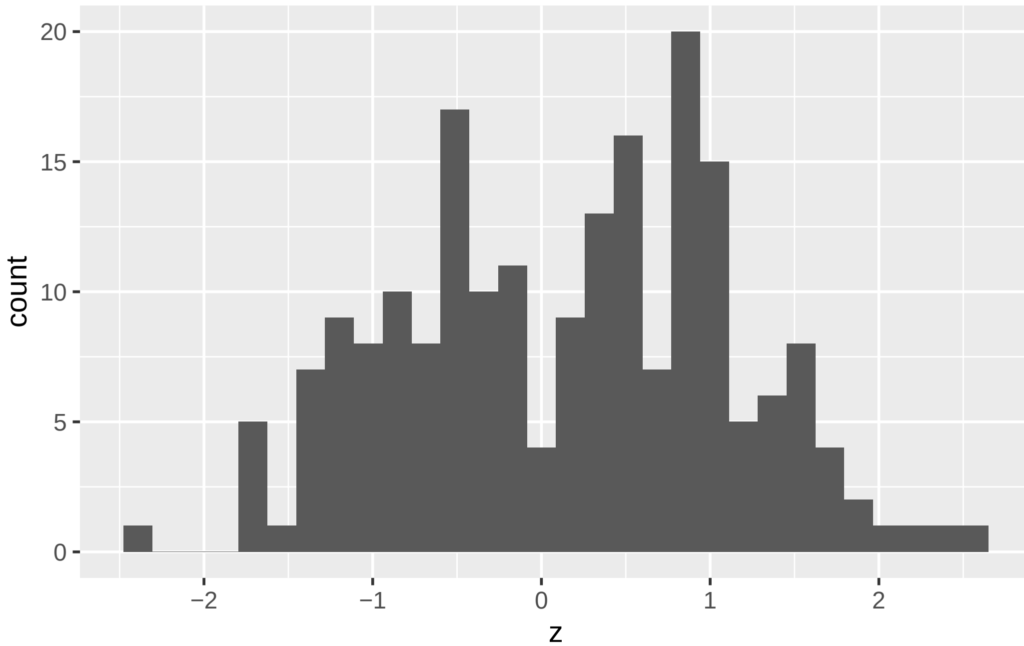
```
#directly
n <- 200
mu <- 5
sgm <- 2
x <- rnorm(n,mu,sgm)
df <- data.frame(x)
#  by standardizing
z <- (x - mu)/sgm
df_z <- data.frame(z)
```

(a) Plot histogram with density overlay.

```
#directly
ggplot(df , aes(x)) + geom_histogram()
```

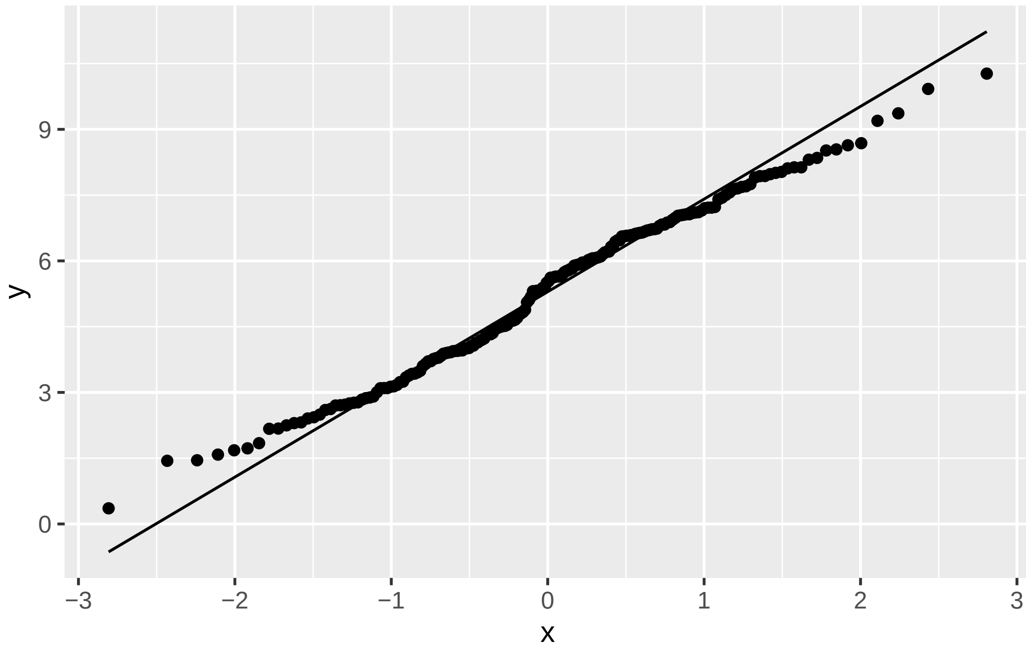



```
#  by standardizing
ggplot(df_z , aes(z)) + geom_histogram()
```

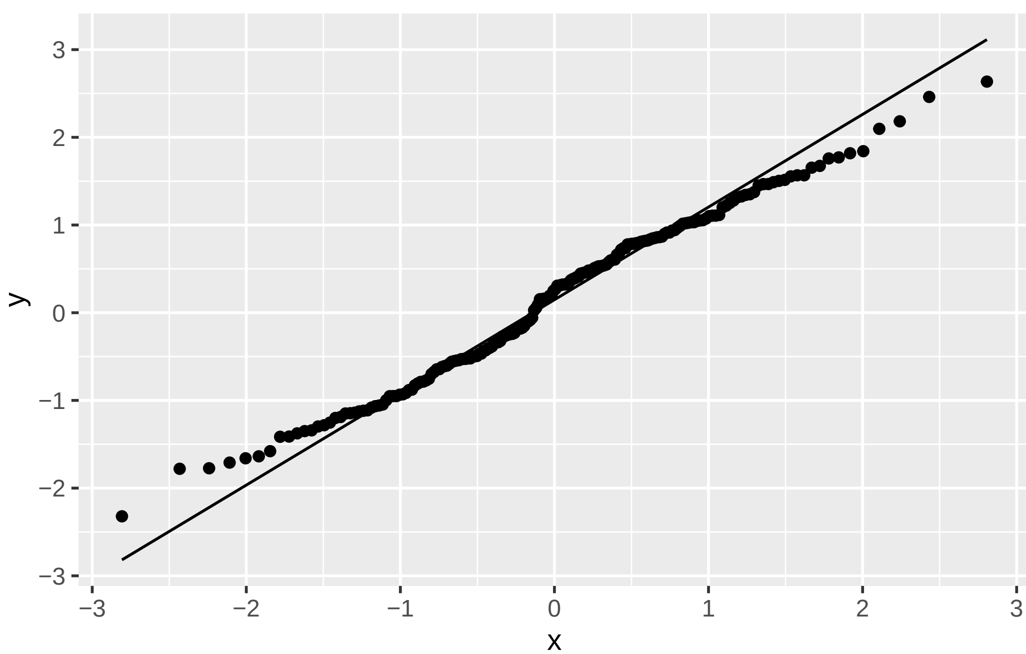


(b) Make a QQ-plot to assess normality.

```r
#directly
ggplot(df, aes(sample = x)) +
  stat_qq() +
  stat_qq_line()
```




```
#  by standardizing
ggplot(df_z, aes(sample = z)) +
  stat_qq() +
  stat_qq_line()
```



12. (a) For $Z \sim \mathcal{N}(0,1)$, compute z_α such that $P(|Z| \leq z_\alpha) = 0.99$.

```
p <- 0.99
qnorm((1 + p)/2)
```

```
[1] 2.575829
```

(b) For $X \sim \mathcal{N}(\mu = 120, \sigma = 10)$, find the symmetric 90% interval around the mean

(c) Verify by simulation that about 90% of draws fall in your interval from (b).