Waseda University
School of Political Science and Economics

Homework 1

Daniel Fabio Groth

Econometrics, Fall 2024

Table of contents

Exercise 1

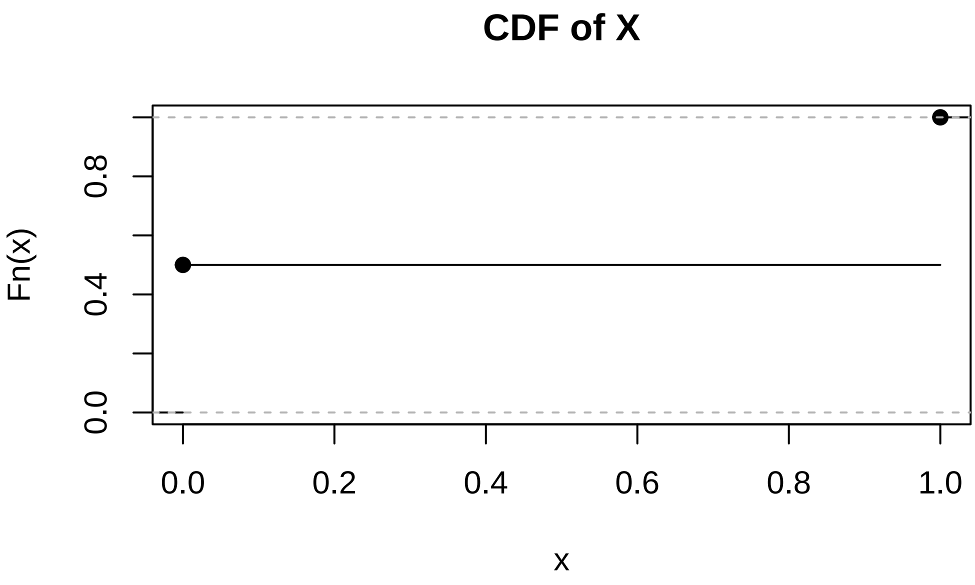
Let X denote the result of flipping a fair coin: X = 1 for head and X = 0 for tail.

1. Draw the CDF of X.

```
# Random variable X  between 0 and 1 where X = 1 is head and X = 0 is tail.
X <- c(1, 0)
X
```

```
[1] 1 0
```

```
# 1. CDF of X
F <- ecdf(X)
plot(F, xlim = c(0,1), main = "CDF of X")
```



2. Calculate E(X).

The expected value of X is given by:

$$E(X) = \sum_{i=1}^{n} x_i p_i$$

where x_i is the value of the random variable and p_i is the probability of the random variable taking the value x_i.

Calculating it will look like this:

$$E(X) = 1 \cdot 0.5 + 0 \cdot 0.5 = 0.5$$

```
# 2. E(X)
mean(X)
```

```
[1] 0.5
```

3. Calculate Var(X).

The variance of X is given by:

$$Var(X) = E(X^2) - E(X)^2$$

where $E(X^2)$ is the expected value of the random variable squared.

Calculating it will look like this:

$$Var(X) = E(X^2) - E(X)^2 = 1^2 \cdot 0.5 + 0^2 \cdot 0.5 - 0.5^2 = 0.25$$

```
# 3. Var(X)
# var in R is the sample variance
# so we need to calculate the population variance

var(X) * (1-1/length(X))
```

```
[1] 0.25
```

Exercise 7

Let X be a random variable whose PDF is given by:

$$f(x) = \begin{cases} \frac{1}{2} + \frac{1}{6}x & \text{for } x \in [-3, 0) \\ \frac{1}{2} - \frac{1}{2}x & \text{for } x \in [0, 1] \\ 0 & \text{otherwise} \end{cases}$$

1. Derive the CDF of X and draw a graph of it.

The CDF of X is given by:

$$F(x) = \int_{-\infty}^{x} f(t)dt$$

where $f(t)$ is the PDF of X.

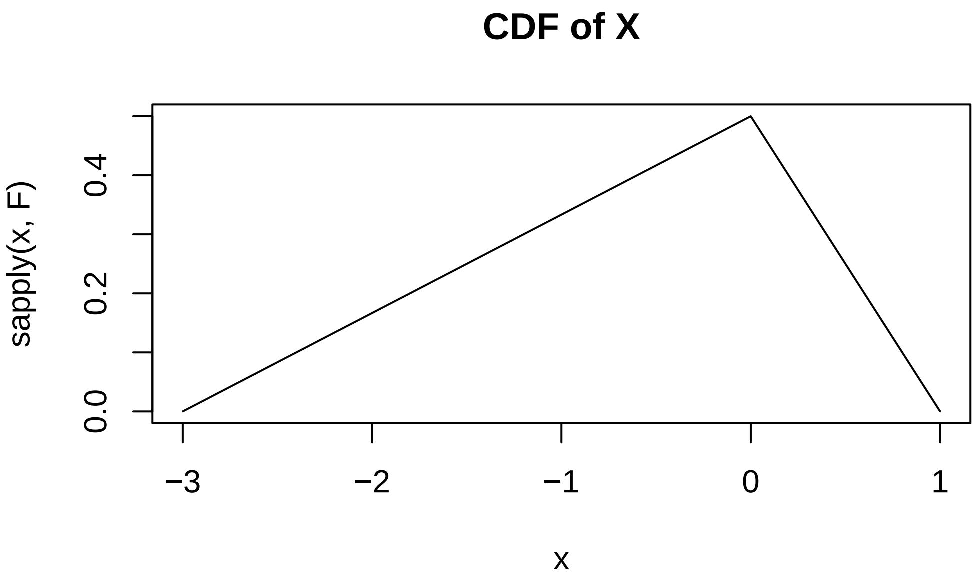
Calculating it will look like this:

$$F(x) = \int_{-3}^{0} \left(\frac{1}{2} + \frac{1}{6}t\right) dt + \int_{0}^{x} \left(\frac{1}{2} - \frac{1}{2}t\right) dt$$

```r
# 1. CDF of X
F <- function(x) {
  if (x >= -3 && x < 0) {
    return(1/2 + 1/6 * x)
  } else if (x >= 0 && x <= 1) {
    return(1/2 - 1/2 * x)
  } else {
    return(0)
  }
}

x <- seq(-3, 1, 0.01)

plot(x, sapply(x, F), type = "l", main = "CDF of X")
```



2. Calculate Pr(-1   X).

The probability that X is between -1 and 0 is given by:

$$Pr(-1 \leq X) = F(0) - F(-1)$$

Calculating it will look like this:

$$Pr(-1 \leq X) = F(0) - F(-1) = \left(\frac{1}{2} - \frac{1}{2} \cdot 0\right) - \left(\frac{1}{2} + \frac{1}{6} \cdot -1\right)$$

```
# 2. Pr(-1   X)
F(0) - F(-1)
```

```
[1] 0.1666667
```

$$Pr(-1 \leq X) = F(0) - F(-1) = \left(\frac{1}{2} - \frac{1}{2} \cdot 0\right) - \left(\frac{1}{2} + \frac{1}{6} \cdot -1\right) = \frac{1}{2} - \frac{1}{3} = 0.1666667$$

```
# or
(1/2 - 1/2 * 0) - (1/2 + 1/6 * -1)
```

```
[1] 0.1666667
```

3. Calculate E[X].

The expected value of X is given by:

$$E(X) = \int_{-\infty}^{\infty} x f(x) dx$$

where $f(x)$ is the probability density function of X.

Calculating it will look like this:

$$E(X) = \int_{-3}^{0} x \left(\frac{1}{2} + \frac{1}{6}x\right) dx + \int_{0}^{1} x \left(\frac{1}{2} - \frac{1}{2}x\right) dx = -\frac{3}{4} + \frac{1}{12} = -0.6666667$$

```r
# 3. E[X]

# Defining the PDF for each interval
f1 <- function(x) x * (1/2 + 1/6 * x)   # For interval [-3, 0)
f2 <- function(x) x * (1/2 - 1/2 * x)   # For interval [0, 1]

# Integrate over each interval
expectation1 <- integrate(f1, lower = -3, upper = 0)$value
expectation2 <- integrate(f2, lower = 0, upper = 1)$value

# Sum the expectations from both intervals
sum(expectation1, expectation2)
```

```
[1] -0.6666667
```

Exercise 18

Let X be a random variable with E[X] = 1 and V[X] = 5. Calculate the following.

1. $\mathrm{E}[(3+X)^2]$

The expected value of $(3+X)^2$ is given by:

$$E[(3+X)^2] = E[9+6X+X^2] = 9+6E[X]+E[X^2]$$

Calculating it will look like this:

$$E[(3+X)^2] = 9+6+1+5 = 21$$

```r
# 1. E[(3+X)^2]

9 + 6 + 1 + 5
```

```
[1] 21
```

2. $\mathrm{V}[5+5X]$

To find the variance of 5 + 5X we can use the property of variance that says:

$$V[a+bX] = b^2V[X]$$

$$V[5+5X] = V[5]+5^2V[X] = 0+25\cdot 5 = 125$$

Exercise 19

Let D be a binary random variable with Pr(D = 1) = 0.8 and Pr(D = -1) = 0.2, and X and Y be normally distributed as N(0,3) and N(0,5), respectively. Further, let Z = DX + (1-D)Y.

Assume X,Y and D are independent.

Calculate the following:

1. E[(XY)^2]

The expected value of $(XY)^2$ is given by:

$$E[(XY)^2] = E[X^2Y^2] = E[X^2]E[Y^2]$$

D does not affect the expected value of X and Y, so we can calculate the expected value of X and Y separately and then multiply them.

Due to the independence of X and Y, the expected value of their product is the product of their expected values.

$$E[X^2] = V[X] + E[X]^2 = 3 + 0 = 3$$

$$E[Y^2] = V[Y] + E[Y]^2 = 5 + 0 = 5$$

Calculating $E[(XY)^2]$ using independence will look like this:

$$E[(XY)^2] = 3 \cdot 5 = 15$$

Exercise 26

Let X_1 and X_2 be the results of two fair dices. We define $Y = X_1 + X_2$.

Compute the following:

1. E[Y]

The expected value of Y is given by:

$$E[Y] = E[X_1 + X_2] = E[X_1] + E[X_2]$$

Since the dice are fair, the expected value of each die is 3.5.

Calculating E[Y] will look like this:

$$E[Y] = 3.5 + 3.5 = 7$$

```
# 1. E[Y]
Dice_1 <- 1:6
Dice_2 <- 1:6

E_Y <-mean(Dice_1) + mean(Dice_2)
E_Y
```

```
[1] 7
```

2. E[Y|X_1 and X_2 are both even]

The expected value of Y given that X_1 and X_2 are both even restrics the possible values to:

$$X_1, X_2 \in \{2, 4, 6\}$$

Now we need to compute all the different outcomes under this condition:

```
# Define the possible values for dice 1 and 2
values <- c(2, 4, 6)

# Generate all possible combinations of dice 1 and 2
combinations <- expand.grid(Dice_1_even = values, Dice_2_even = values)

# Calculate Y = dice 1 + dice 2 for each combo
Y <- combinations$Dice_1_even + combinations$Dice_2_even

# Store results in a vector
```

```
Y_vector <- as.vector(Y)

# Display the outcomes of Y
Y_vector
```

```
[1]  4  6  8  6  8 10  8 10 12
```

The probability distribution for each pair is equally likely, giving 9 possible outcomes for Y since:

$$X_1, X_2 \in \{2, 4, 6\} \cdot \{2, 4, 6\}$$

with probability $\frac{1}{9}$ for each pair.

```
# Count the occurrences of each value of Y
Y_counts <- table(Y_vector)
# Extract the values and probabilities as numeric vector
Y_values <- as.numeric(names(Y_counts))
# Calculate the probabilities
Y_probs <- Y_counts / length(Y_vector)

# Calculate the expected value of Y by summing the products of the
#values and probabilities
expected_Y <- sum(Y_values * Y_probs)

expected_Y
```

```
[1] 8
```

```
# This is the easy way to calculate the expected value of Y given that X1 and X2
# are both even
Dice_1_even <- c(2,4,6)
Dice_2_even <- c(2,4,6)

E_Y_even <- mean(Dice_1_even) + mean(Dice_2_even)
E_Y_even
```

```
[1] 8
```

The expected value of Y given that X_1 and X_2 are both even is given by:

$$E[Y|X_1, X_2 \text{ are both even}] = 4 + 4 = 8$$

3. E[Y|$X_1 \neq X_2$]

The expected value of Y given that $X_1 \neq X_2$ restricts the possible values to:

$$X_1, X_2 \in \{1, 2, 3, 4, 5, 6\} \text{ and } X_1 \neq X_2$$

Now we need to compute all the different outcomes under this condition:

```
# Step 1: Define the values for each die
values <- 1:6

# Step 2: Generate all possible combinations of X1 and X2
combinations <- expand.grid(X1 = values, X2 = values)

# Step 3: Filter for combinations where X1 != X2
combinations_diff <- subset(combinations, X1 != X2)

# Step 4: Calculate Y = X1 + X2 for each combination
Y_vector_diff <- combinations_diff$X1 + combinations_diff$X2

# Step 5: Count occurrences of each unique Y value and calculate probabilities
Y_counts_diff <- table(Y_vector_diff)                    # Frequency of each Y
Y_values_diff <- as.numeric(names(Y_counts_diff))        # Unique Y values
Y_probs_diff <- Y_counts_diff / length(Y_vector_diff)   # Probability of each Y

# Step 6: Calculate the expected value E[Y | X1 != X2]
expected_Y_diff <- sum(Y_values_diff * Y_probs_diff)

# Display the expected value
expected_Y_diff
```

```
[1] 7
```

The expected value of Y given that $X_1 \neq X_2$ is given by:

$$E[Y|X_1 \neq X_2] = 7$$

```
# This is the easy way to calculate the expected value of Y given that
#X1 and X2 are different (code works but is technically not correct since I dont remove
# It works this way since we just remove the case where X1 = X2
#from the possible values of X1 and X2
# and the remaining values cover the same range as the original dice
Dice_1_diff <- 1:6
Dice_2_diff <- 1:6

E_Y_diff <- mean(Dice_1_diff) + mean(Dice_2_diff)
E_Y_diff
```

```
[1] 7
```

Usage of AI

Copilot

As I am writing this document in Rstudio, there is an integration of copilot which sometimes automatically suggests code snippets. Sometimes it works great and my latex math gets written perfectly, and other times it just gives me a bunch of random unrelevant latex math or code.

Here is a Link to copilot if you want to read about it, but I assume you already know of it.

ChatGPT

Here is the link to the conversation where I asked questions:

Homework 1 ChatGPT conversation