ECON 21020: Econometrics
The University of Chicago, Spring 2022
Instructor: Thomas Wiemann
Problem Set # 1: Review of Probability Theory
Due: 11:59am on April 6, 2022

Problem 1 15 Points

Table 1 gives the joint probability mass function between employment status and college degree in the US working-age population.

Table 1: Joint Probability Mass Function

	Unemployed ($Y = 0$)	Employed ($Y = 1$)
Non-college grads ($X = 0$)	0.026	0.576
College grads ($X = 1$)	0.009	0.389

a)

Table 1 states $P(Y = 1, X = 1) = 0.389$. Give an economic interpretation of this statement.

b)

Compute the unconditional probability of being employed $P(Y = 1)$.

c)

Compute the unconditional probability of having a college degree $P(X = 1)$.

d)

Compute the unemployment share for college and non-college graduates – that is, calculate $P(Y = 0|X = 1)$ and $P(Y = 0|X = 0)$.

e)

Are employment status and college degree independent? Explain briefly.

Problem 2 10 Points

Let $X \sim \mathcal{N}(0, 1)$ and define $Y \equiv a + bX$ for $a, b \in \mathbb{R}$.

a)

Give an expression for $E[Y]$ involving only a, b, and $E[X]$.

b)

Give an expression for $Var(Y)$ involving only a, b and $Var(X)$.

Problem 3   10 Points

Take $n \in \mathbb{N}$ and let $\{X_i\}_{i=1}^n$ be a collection of independent random variables with $\operatorname{supp} X_i = \{0, 1\}$ and $p = P(X_i = 1)$, $\forall i = 1, \ldots, n$. Define $X_n = \sum_{i=1}^n X_i$.

a)

Show that

$$E[X_n] = np. \tag{1}$$

b)

Show that

$$Var(X_n) = np(1 - p). \tag{2}$$

Problem 4   15 Points

Let X be a discrete random variable with $\operatorname{supp} X = \mathcal{X}$. Define $Y \equiv g(X)$ and let $\operatorname{supp} Y = \mathcal{Y}$.

a)

By definition, we have $E[Y] = \sum_{y \in \mathcal{Y}} yP(Y = y)$. Define $g^{-1}(y) \equiv \{x \in \mathcal{X} \,|\, g(x) = y\}$, that is, the values of $x \in \mathcal{X}$ such that $g(x) = y$.[1] Show that

$$P(Y = y) = \sum_{x \in g^{-1}(y)} P(X = x). \tag{3}$$

[1] Formally, g^{-1} often referred to as the preimage (or inverse image) of g.

b)

Prove the law of the unconscious statistician for discrete random variables. That is, show that

$$E[g(X)] = \sum_{x \in \mathcal{X}} g(x) P(X = x). \tag{4}$$

(Hint: Note that $\cup_{y \in \mathcal{Y}} g^{-1}(y) = \mathcal{X}$. That is, the union of all preimages $g^{-1}(y)$ is equal to the support of X.)

Problem 5 10 Points

Let X be a random variable with $\operatorname{supp} X = \mathcal{X} \subset \mathbb{R}$ such that $E[X^2] < \infty$.

a)

Show that

$$\arg\min_{a \in \mathbb{R}} E\left[(X - a)^2\right] = E[X]. \tag{5}$$

That is, show that the mean of X is the best constant predictor of X under the L^2 loss function.

Problem 6 15 Points

Let X and Y be a random variables with $\operatorname{supp} X = \mathcal{X}$ and $\operatorname{supp} Y = \mathcal{Y}$ such that $E[X^2] < \infty$ and $E[Y^2] < \infty$. The propose of this exercise is to show that

$$E\left[X|Y\right] \in \arg\min_{g \in \mathbb{G}} E\left[(X - g(Y))^2\right], \tag{6}$$

where g is some function in $\mathbb{G}$, which denotes the set of functions from $\mathcal{Y}$ to $\mathcal{X}$ such that $E[g(Y)^2] < \infty$. Or – in other words – that the conditional expectation function $E\left[X|Y\right]$ is the best predictor of X given Y under the L^2 loss.

a)

Define $U \equiv X - E[X|Y]$. Show that

$$E[U|Y] = 0. \tag{7}$$

b)

Let h be some function of Y such that $E[|h(Y)|] < \infty$. Show that

$$E[Uh(Y)] = 0. \tag{8}$$

c)

Define $V \equiv X - g(Y)$ and $h(Y) \equiv g(Y) - E[X|Y]$. Note that we then have

$$V = (X - E[X|Y]) - h(Y) = U - h(Y). \tag{9}$$

Show that

$$E[V^2] \geq E[U^2]. \tag{10}$$

d)

Explain briefly how the result in part c) relates to the statement in Equation (6).

Problem 7   15 Points

Let X and Y be continuous random variables with $\operatorname{supp} X = \operatorname{supp} Y = \mathbb{R}$ with joint density $f_{y,x}(y, x), \forall (y, x) \in \mathbb{R}^2$. Assume existence of moments and joint support when necessary.

a)

Show that

$$X \perp\!\!\!\perp Y \Rightarrow E[Y|X] = E[Y]. \tag{11}$$

b)

Show that

$$E[Y|X] = E[Y] \Rightarrow corr(X, Y) = 0. \tag{12}$$

c)

Give an example of two random variables that are uncorrelated but not independent.

Problem 8 10 Points

This programming exercise must be completed in R using only the ggplot2 package (and no other packages). The below code snippet illustrates how you can install and load ggplot2.[2]

```
# Install dependencies (only once)
install.packages("ggplot2")

# Load dependencies (every time)
library(ggplot2)
```

If you upload your solutions to a GitHub repository and share the link in your homework solutions, you earn an extra credit of 5 percentage points on this problem set.

a)

Generate $n = 10000$ draws from a standard normal random variable $N(0, 1)$ and plot the simulated data in a histogram using ggplot2.

```
# Generate a vector of n draws from a standard normal rv
n <- 10000
mu <- 0
sigma <- 1
x <- rnorm(n, mu, sigma)

# Plot a histogram of the draws using ggplot2
# [INSERT YOUR CODE HERE]
```

b)

Generate $n = 10000$ draws from a uniform random variable $U(-1, 1)$ and plot the simulated data in a histogram using ggplot2.

```
# Generate a vector of n draws from a uniform(-1, 1) rv
n <- 10000
min_y <- -1
max_y <- 1
y <- runif(n, min_y, max_y)

# Plot a histogram of the draws using ggplot2
# [INSERT YOUR CODE HERE]
```

[2] Note that you only have to install a package once, but need to load it each time you open R.

Problem 9 15 Points (Extra credit)

This is an optional extra credit exercise.

This programming exercise must be completed in so-called base R. That is, don't load any dependencies.

a)

Let $X \sim \text{Bernoulli}(p)$ and $U \sim \mathcal{U}(0,1)$. Show that

$$P(\mathbb{1}\{U \leq p\} = 1) = p \tag{13}$$

and conclude that $\mathbb{1}\{U \leq p\}$ and X are identically distributed.

Solution:

$$\begin{aligned} P(\mathbb{1}\{U \leq p\} = 1) &= P(U \leq p) \\ &= \frac{p-0}{1-0} \\ &= p, \end{aligned}$$

where the second equation follows from the distribution function of uniform random variables. Hence we have $P(\mathbb{1}\{U \leq p\} = 1) = P(X = 1)$. As $\operatorname{supp} \mathbb{1}\{U \leq p\} = \operatorname{supp} X = \{0, 1\}$, it follows that U and X are identically distributed.

b)

R provides a random number generator for Bernoulli random variables through the `sample` command. For example, users can generate `n` draws from $X \sim$ Bernoulli(`p`) using the command `sample(c(0, 1), n, replace = TRUE, prob = c(1-p, p))`. The syntax is arguably not as neat as, for example, the generators for normal or uniform random variables you saw in Problem 8. The purpose of this exercise is letting you code up your own random number generator for Bernoulli random variables.

In particular, use the following code as a starting point to write a custom function `my_rbernoulli` that takes an integer `n` $\in \mathbb{N}$ and a scalar `p` $\in [0, 1]$ and returns `n` draws from $X \sim$ Bernoulli(`p`). Your solution *must* use the function `runif` and *cannot* use the command `sample`.

```
# Define a custom function that returns draws from a Bernoulli rv
```

```
my_rbernoulli <- function(n, p) {
    x <- # [INSERT YOUR CODE HERE]
    # Return draws
    return(x)
}#MY_RBERNOULLI

# Test the custom Bernoulli generator function
x <- my_rbernoulli(10000, 0.5)
length(x) == 10000 # should return TRUE
mean(x) # should a number near 0.5
```

c)

R also provides a random number generator for binomial random variables. For example, users can generate `n` draws from $X \sim \text{Binomial}(\texttt{p}, \texttt{m})$ using the command `rbinom(n, p, m)`. The purpose of this exercise is letting you code up your own random number generator for binomial random variables using your `my_rbernoulli` function defined in part b).

In particular, use the following code as a starting point to write a function that takes integers $\texttt{n}, \texttt{m} \in \mathbb{N}$ and a scalar $\texttt{p} \in [0, 1]$ and returns `n` draws from $X \sim \text{Binomial}(\texttt{p}, \texttt{m})$. Your solution *must* use the function `my_rbernoulli` and *cannot* use the command `rbinom`.

```
# Define a custom function that returns draws from a Binomial rv
my_rbinomial <- function(n, p, m) {
  # [INSERT YOUR CODE HERE]
}#MY_RBINOMIAL

# Test the custom Binomial generator function
x <- my_rbinomial(10000, 0.5, 10)
length(x) == 10000 # should return TRUE
mean(x) # should a number near 5
```