**Data Analytics (CMP330)**

# **Practical 4 – Normal distribution and probability density functions using R**

1. **Normal Probability Distribution**

R has four in built functions for normal distribution including dnorm, pnorm, qnorm, and rnorm function.

1. **dnorm()** gives height of the probability distribution at each point for a given mean and standard deviation.

Try the following code,

# Create a sequence of numbers between -10 and 10 incrementing by 0.1.

**x <- seq(-10, 10, by = .1)**

**x**

# Choose the mean as 2.5 and standard deviation as 0.5.

**y <- dnorm(x, mean = 2.5, sd = 0.5)**

# plot the figure

**plot(x,y, main= 'Probability Distribution')**

1. **pnorm()** gives the probability of a normally distributed random number to be less than the value of a given number. It is also called "Cumulative distribution Function".

Try the following code,

# Create a sequence of numbers between -10 and 10 incrementing by 0.1.

**x <- seq(-10, 10, by = .1)**

**x**

# Choose the mean as 2.5 and standard deviation as 0.5.

**y <- pnorm(x, mean = 2.5, sd = 0.5)**

# plot the figure

**plot(x,y, main= 'Cumulative Distribution')**

1. **qnorm()** takes the probability value and gives a number whose cumulative value matches the probability value. qnorm() is the inverse of the cumulative distribution function, e.g. if the probability is 0.05 it gives the 5th quantile.

Try the following code,

# Create a sequence of probability values incrementing by 0.01.

**x <- seq(0, 1, by = 0.01)**

**x**

# Choose the mean as 2.5 and standard deviation as 0.5.

**y <- qnorm(x, mean = 2.5, sd = 0.5)**

# plot the figure

**plot(x,y, main= 'qnorm')**

1. **rnorm()** is used to generate random numbers whose distribution is normal. It takes the sample size as input and generates that many random numbers. Try the following code snippet to draw a histogram to show the distribution of the generated numbers.

# Create three different vectors of random numbers from a normal distribution

**x10 <- rnorm(10, mean = 2.5, sd = 0.5)**

**x100 <- rnorm(100, mean = 2.5, sd = 0.5)**

**x1000 <-rnorm(1000, mean = 2.5, sd = 0.5)**

**x10**

# set par options

**par(mfrow=c(1,3))**

# Plot the histogram for these samples.

**hist(x10, main = 'Histogram of x10'**, **breaks = 5)**

**hist(x100, main = 'Histogram of x100', breaks = 20)**

**hist(x1000, main = 'Histogram of x1000', breaks = 100)**

**Exercise 1. Suppose the marks for a module are normally distributed with a mean of 63% and standard deviation of 9%.**

1. **Plot a normal distribution with these parameters and the corresponding cumulative distribution**
2. **What proportion of students would you be expected to have a mark of 50% or less?**
3. **What proportion of students would you expect to have a mark above 80%?**
4. **What is the 70th percentile for the distribution?**
5. **Determine the z-score of a student who got a mark of 70%.**
6. **Use the rnorm function to simulate the results of 100 students with this distribution. Plot the histogram (using the ‘freq = FALSE’ as one of the arguments). How does this compare the curve from part (a)?**
7. **From the simulation in part (f), determine the proportion of students who got 50% or less. How does this compare with your answer in part (b)?**
8. **Central limit theorem demonstration**

The Central limit theorem states that when a sufficiently large number of samples drawn from independent random variables, the arithmetic mean of their distributions will behave like a normal distribution, commonly known as a bell-shaped distribution.

Flipping a fair coin many times the probability of getting a given number of heads in a series of flips should follow a normal curve, with mean equal to half the total number of flips in each series. Here 1 represent heads and 0 tails.

# Draw 100 samples and take the arithmetic mean 10000 times

**k =10000**

**for (i in 1:k){**

**x[i] = mean(sample(0:1,100,replace = TRUE))}**

# Applying dev.off to reset par()

**dev.off()**

# plot the mean of sample

**hist(x, col ="light green", main="Sample size = 100",xlab ="flipping coin ")**

# plot the population mean

**abline(v = mean(x), col = "red")**

1. **Test for normality**

When dealing with a set of numbers where you are not certain if the distribution is actually normal, you can use the Shapiro-Wilk test in the lecture. Another test for normality is the Kolmogorov-Smirnov test. In the results a high p-value indicates normality. In contrast, a low p-value indicates the distribution is not normal.

Try the following code,

# generate a random sequence of numbers from a uniform distribution

**x = runif(1000)**

# plot the histogram of x

# Histogram of a random uniform distribution which shows what a non-normal distribution might look like

**hist(x, main='Histogram of x')**

# run Shapiro-Wilk normality test it is not a normal distribution, and we expect a low p-value

**shapiro.test(x)**

# generate a random sequence of numbers from a normal distribution

**y = rnorm(1000)**

# plot the histogram of x

# Histogram of a random uniform distribution which shows what a normal distribution might look like

**hist(y, main='Histogram of y')**

# run Shapiro-Wilk normality test it is a normal distribution, and we expect a high p-value

**shapiro.test(y)**

**Exercise 2. Using the Titanic dataset (see the file ‘train.csv’ that we used last week) you should investigate the distribution of ages (variable ‘Age’) to determine whether they are normally distributed. First, try this by omitting all the missing data, plot a histogram of the ages and apply the Shapiro-Wilk test. Then repeat the process by replacing missing ages with the mean age in the dataset.**

1. **Binomial Probability Distribution**

Binomial Probability Distribution is a probability distribution that summarizes the likelihood that a value will take one of two independent values under a given set of parameters or assumptions. Key features are as below:

* Fixed number of trials - n
* 2 possible outcomes for each trial
* Probability of success remains same for all trials
* Trials are independent of each other
* Interested in probability for x successes in n trials, given a success probability p for each trial.

R has four in built functions for Binomial Probability Distribution including dbinom, pbinom, qbinom, and rbinom function.

1. **dbinom()** gives the probability density distribution at each point.

Try the following code,

# Create a sample of 100 numbers which are incremented by 1.

**x <- seq(0,100,by = 1)**

# Create the binomial distribution.

**y <- dbinom(x,100,0.5)**

# Plot the graph for this sample.

**plot(x,y, main='Binomial Probability Distribution')**

1. **pbinom()** gives the cumulative probability of an event. It is a single value representing the probability.

Try the following code,

# Probability of getting 25 or less heads from 50 tosses of a coin.

**x <- pbinom(25,50,0.5)**

**x**

1. **qbinom()** takes the probability value and gives a number whose cumulative value matches the probability value, i.e. it is the inverse of the cumulative distribution function.

Try the following code,

# What is the number of heads such that the probability of getting no more

# than that number of heads is 0.05 when a coin is tossed 50 times?

# Or equivalently, what is the 5th quantile of the distribution?

**x <- qbinom(0.05,50,0.5)**

**x**

1. **rbinom()** generates required number of random values of given probability from a given sample

Try the following code,

# Find 10 random values from a sample of 100 with probability of 0.7.

**x <- rbinom(10,100,0.7)**

**x**

**Exercise 3. An online store has a conversion rate of 5% which means that 5% of visitors to the site make a purchase. Suppose there are 1000 visitors to the store on a given day. Modelling the number of sales using a binomial distribution, plot the distribution and estimate the probability that there will be more than 60 sales during the day.**

**Bonus exercise. A binomial distribution of *n* trials and probability of success *p* can be approximated by a normal distribution with a mean of *np* and standard deviation of . For the problem in exercise 3, plot this normal distribution on the same figure as the binomial distribution to compare them and use the normal distribution to estimate the probability that there will be more than 60 sales during the day. How close is this to the result you got previously?**

1. **Poisson Distribution**

Poisson distribution is a probability distribution that is used to show how many times an event is likely to occur over a specified period. Key features are as below:

* Outcomes are success or failure
* Average number of successes (mu) in the specific region (time, location) are known
* Outcomes are random. Occurrence of one outcome does not affect the outcome of the other.
* Outcomes are rare compared to possible outcomes

R has 3 built-in functions for Poisson distribution including dpois, ppois, rpois function.

1. **dpois** **(x, lambda)** gives the probability that there will be x successes per period for an event with an average number of lambda successes. P(X = x)

Try the following code,

# Data from the maternity ward in a certain hospital shows that there is a

# historical average of 4.5 babies born in this hospital every day. What is the

# probability that 6 babies will be born in this hospital tomorrow?

**x <- dpois(6, 4.5)**

**x**

1. **ppois (x, lambda, lower.tail = TRUE)** that there will be x or fewer successes per period for an event with an average number of lambda successes. P(X <= x)

Try the following code,

# What is the probability that 6 babies or less will be born in this hospital tomorrow?

**x <- ppois(6, 4.5)**

**x**

# What is the probability that more than 6 babies will be born in this hospital tomorrow?

**x <- ppois(6, 4.5, lower.tail = FALSE)**

**x**

1. **rpois(n, lambda)** generates required n random values that follow a Poisson distribution with an average number of lambda successes

Try the following code,

# Simulate births in this hospital for a year

**x <- rpois(365, 4.5)**

# plot histogram

**hist(x, main = 'Simulated births in a hospital with Pois (lambda = 4.5)')**

**Exercise 4. A company’s server experiences an average of two downtimes per month. Modelling the downtimes with a Poisson distribution, plot the distribution and determine a) the probability that the server will go down at most one time next month, and b) the probability that it will go down three times next month.**