

Practical - 6

Aim:- Standard probability distributions -

- Calculation of probability, mean and variance based on Normal distribution
- Calculation of probability based on Normal distribution

Probability Density Calculation

The following examples demonstrate how to calculate the value of the cumulative distribution function at (or the probability to the left of) a given number.

Normal (0, 1) Distribution:

- > x <- c(-2, -1, 0, 1, 2)
- > x
- > pnorm(x)

Binomial (n, p) Distribution:

- > x <- c(0, 1, 2, 5, 8, 10, 15, 20)
- > pbinom(x, size = 20, prob = .2)

Quantiles

The following examples show how to compute the quantiles of some common distributions for a given probability (or a number between 0 and 1).

Normal (0, 1) Distribution:

- > y <- c(.01, .05, .1, .2, .5, .8, .95, .99)


```
> qnorm(y, mean = 0, sd = 1)
```

Binomial (n, p) Distribution:

```
> y <- c(.01, .05, .1, .2, .5, .8, .95, .99)
```

```
> qbinom(y, size = 30, prob = .2)
```

Random Variable generation

The following examples illustrate how to generate random samples from some of the well-known probability distributions.

- Normal (μ, σ^2) Distribution:

The first sample is from $N(0, 1)$ distribution and the next one from $N(5, 1)$ distribution

```
> z <- rnorm(10)
```

```
> z
```

```
> w <- rnorm(1000, mean = 5, sd = 1)
```

```
> w
```

Binomial (n, p) Distribution:

```
> k <- rbinom(20, size = 5, prob = .2)
```

```
> k
```

Density Plots

- Plotting the probability density function (pdf) of a Normal distribution:

```
> x11()
```

```
> x <- seq(-4.5, 4.5, .1)
```


- > normdensity <- dnorm(x, mean = 0, sd = 1)
- > plot(x, normdensity, type = "l")
- Plotting the probability mass function (pmf) of a Binomial distribution:
 - > par(mfrow = c(2, 1))
 - > k <- c(1:30)
 - > plot(k, dbinom(k, size = 30, prob = .15), type = "h")
 - > plot(k, dbinom(k, size = 30, prob = .4), type = "h")
 - > par(mfrow = c(1, 1))
- Discrete Probabilities For a discrete random variable, you can use the probability mass to find $P(X = k)$
 - > dbinom(3, size = 10, prob = 0.5)

Q-Q plot

Use the function qqnorm for plotting sample quantiles against theoretical (population) quantiles of standard normal random variable.

- > stdnormsamp <- rnorm(100, mean = 0, sd = 1)
- > normsamp <- rnorm(100, mean = 5, sd = 1)
- > binomsamp <- rbinom(100, size = 20, prob = .25)
- > par(mfrow = c(2, 2))

- > qqnorm(stdnormsamp, main="Normal Q-Q plot: N (0,1) samples")
- > qqline(stdnormsamp, col = 2)
- > qqnorm(normsamp, main="Normal Q-Q plot: N (5,1) samples")
- > qqline(normsamp, col = 2)
- > qqnorm(binomsamp, col = 2, main = "Normal Q-Q plot: Bin (20, .25) samples")
- > qqline(binomsamp, col = 2)

Practical - 7

Aim :-

```
> Mydata = read.csv("C:\\Users\\User\\Desktop\\
InternetMobileTime.csv", header = TRUE)
> Mydata
> attach(Mydata)
> xbar = mean(x)
> s = sd(x)
> n = length(x)
> Mu = 144
> tstat = (xbar - Mu) / (s / (n^0.5))
> tstat
> Mu = 120
> tstat(xbar - Mu) / (s / (n^0.5))
> tstat
```

R functions: binom.test() and prop.test()

```
> binom.test(x, n, p = 0.5, alternative = "two.sided")
> prop.test(x, n, p = NULL, alternative = "two.sided",
> res <- prop.test(x = 95, n = 160, p = 0.5, correct = TRUE)
> res <- prop.test(x = 95, n = 160, p = 0.5, correct = FALSE)
> res
> prop.test(x = 95, n = 160, p = 0.5, correct = FALSE,
alternative = "less")
> prop.test(x = 95, n = 160, p = 0.5, correct = FALSE,
alternative = "greater")
```


Two Proportion Test

- > prop.test(x, n, p = NULL, alternative = "two.sided", correct = TRUE)
- > res <- prop.test(x = c(490, 400), n = c(500, 500))
- > res
- > prop.test(x = c(490, 400), n = c(500, 500), alternative = "less")
- > prop.test(x = c(490, 400), n = c(500, 500), alternative = "greater")

Practical - 8

Aim:- Small sample tests based on t and F -

- t-test for significance of single mean, population variance being unknown (single mean $H_0: \mu = \mu_0$)
- t-test for significance of the difference between two sample means (Independent samples)
- t-test for significance of the difference between two sample means (Related samples)
- F-Test to Compare Two Variances

```
> x = rnorm(10)
> y = rnorm(10)
> t.test(x, y)
> ttest = t.test(x, y)
> names(ttest)
> ttest$statistic
> ttest[['statistic']]
> range(ts)
> range(ts)
> pts = seq(-4.5, 4.5, length = 100)
> plot(pts, dt(pts, df = 18), col = 'red', type = 'l')
> lines(density(ts))
> t.test(x, y)
> t.test(x, y, var.equal = TRUE)
> tps = replicate(1000, t.test(rnorm(10), rnorm(10))
                                $p.value)
> plot(density(tps))
```


- Method1: `var.test(x, y, alternative = "two.sided")`
- Method2: `var.test(values ~ groups, data, alternative = "two.sided")`

Method: F-Test in R

```
> x <- c(18, 19, 22, 25, 21, 28, 41, 45, 51, 55)
> y <- c(14, 15, 15, 17, 18, 22, 25, 25, 27, 34)
> var.test(x, y)
```

Method2: F-Test in R

```
> data <- data.frame(values = c(18, 19, 22, 25, 21,
  28, 41, 45, 51, 55, 14, 15, 15, 17, 18, 22, 25, 25, 27, 34),
  group = rep(c('A', 'B'), each = 10))
> var.test(values ~ group, data = data)
```


Practical - 9

Aim:- Analysis of Variance -

- a. Perform One-way ANOVA
- b. Perform Two-way ANOVA

Our objective is to test the following assumption:

- H_0 : There is no difference in survival time average between group
- H_3 : The survival time average is different for at least one group.

You will proceed as follow:

- Step1: Check the format of the variable poison
- Step2: Print the summary statistic: count, mean and standard deviation
- Step3: Plot a box plot
- Step4: Compute the one-way ANOVA test
- Step5: Run a pairwise t-test

```
> library(dplyr)
> PATH <- "https://raw.githubusercontent.com/guru99-edu/R-Programming/master/poisons.csv"
> df <- read.csv(PATH) %>% select(-x) %>% mutate
> library(dplyr)(poison = factor(poison, order = TRUE))
> glimpse(df)
Rows: 48
Columns: 3
```



```
$ time < dbl > 0.31, 0.45, 0.46, 0.43, 0.36, 0.29, 0.40,
0.23, 0.22, 0.21, 0.18, 0.23, 0.82, 1.10, 0.88, 0.72, 0.92,
0.61, 0.49, 1.24, 0.30, 0.37, 0.38, 0.29, 0.43, 0.45,
0.63, 0.16, 0.4~
```

```
$ poison < ord > 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 1, 1, 1, 1,
2, 2, 2, 2, 3, 3, 3, 3, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 1, 1,
1, 1, 2, 2, 2, 2, 3, 3, 3, 3
```

```
$ treat < chr > "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A",
"A", "A", "B", "B", "B", "B", "B", "B", "B", "B", "B", "B", "B", "B",
"B", "C", "C", "C", "C", "C", "C", "C", "C", "C", "C", "C", "C", "~
```

- > levels(df\$poison)
- > df %>% group_by(poison) %>% summarise(count = n(), mean_time = mean(time, na.rm = TRUE), sd_time = sd(time, na.rm = TRUE))
- > library(ggplot2)
- > ggplot(df, aes(x = poison, y = time, fill = poison)) + geom_boxplot() + geom_jitter(shape = 15, color = "steelblue", position = position_jitter(0.2)) + theme_classic()
- > anova_one_way <- aov(time ~ position, data = df)
- > summary(anova_one_way)
- > TukeyHSD(anova_one_way)

Hypothesis in two-way ANOVA test:

- > anova_two_way <- aov(time ~ position + treat, data = df)
- > summary(anova_two_way)

Practical - 10

Aim:- Non Parametric Test

- Sign Test
- Wilcoxon Signed - Rank Test
- Mann - Whitney - Wilcoxon Test
- Kruskal - Wallis Test

Kruskal Wallis Test

- > my_data <- PlantGrowth
- > head(my_data)
- > kruskal.test(weight ~ group, data = my_data)

Wilcoxon Test

- > pairwise.wilcox.test(PlantGrowth\$weight, PlantGrowth\$group, p.adjust.method = "BH")

Mann Whitney Test

- > mtcars\$mpg
- > mtcars\$cyl
- > wilcox.test(mpg ~ cyl, data = mtcars)

Sign Test

- > binom.test(5, 18)