R-programming - 2

1. Probability and probability distributions
2. Statistical inference-t-test
3. Statistical inference- interval estimation
4. Correlation analysis using R
5. Regression analysis using R.

Introduction: In the previous session we have seen how we can enter the data to R-software and used R-programming to carry out simple statistical analysis. In this session we compute probabilities using R-Program, correlation and regression and some statistical inference.

Probability, Distributions and Simulation:

We look at some of the basic operations associated with probability distributions. There are a large number of probability distributions available, but we only look at a few. If you would like to know what distributions are available you can do a search using the command help.search("distribution").

To get a full list of the distributions available in R you can use the following command:

help(Distributions)

R allows for the calculation of probabilities (including cumulative), the evaluation of density/mass functions, and the generation of pseudo-random variables following a number of common distributions

Distribution	R name	Additional arguments	Argument defaults
beta	beta	shape1 (α), shape2 (β)	
binomial	binom	size (n), prob(p)	
Chi-square	chisq	df (degree of freedom)	
continuous uniform	unif	min(θ_1), max(θ_2)	min=0, max=1
exponential	exp	rate(=$\frac{1}{\beta}$)	rate=1

F distribution	f	df1,df2	
gamma	gamma	shape(α),scale(β)	scale=1
hypergeometric	hyper	m=r, n=N-r, k=n (sample size)	
normal	norm	mean (μ), sd(σ)	mean=0,sd=1
Poisson	pois	lambda(λ)	
t distribution	t	Df	
Weibull	weibull	Shape,scale	scale=1

Prefix each R name given above with'd' for the density or mass function, 'p' for the c.d.f, 'q' for the percentile function, and 'r' for the generation of pseudo random variables.

Example:

>x<-rnorm(100) # simulate 100 standard normal random variables put in x.

> w<-rexp(1000,rate=0.1) # simulate 1000 from exp(θ=10)

>dbinom(3,size=10,prob=0.25) #P(X=3) for X~b(n=10,p=0.25)

>pbinom(3,size=10,prob=0.25) #P(X≤3) in the above distribution.

>pnorm(12,mean=10,sd=2) #P(X≤12) for X~N(μ=10, σ=2)

>qnorm(0.75, mean=10,sd=2) # 3rd quartile of N(μ=10, σ=2)

>qchisq(0.10,df=8) #10th percentile of $\chi^2(8)$

>qt(0.95,df=20) #95th percentile of t(20)

STATISTICAL INFERENCE

- Independent sample t test

```
# The format in which data has to be entered
# x    y

# where x is  quantitatve
# where y is  quantitatve
#------------------------------------------------------------
#1. Read the data
 data<-read.table("E:/rdata.txt",header=TRUE);
 data;
```

```
 x<-data$x;
 y<-data$y;

#2. Test for normality of the data
 shapiro.test(x);
 shapiro.test(y); # If any one of the variable is significant
then proceed to step 7

#3. Test for randomness
 # install the package "randtests"
       library(randtests);
       runs.test(x);
       runs.test(y); # If any one of the variable is significant
then proceed to step 7

#4. Test for equality of variances
 var.test(x,y,ratio = 1,alternative="two.sided"); # If the test
is significant then proceed to step 6

#5. Independent sample t-test (Equal variance)

 #5.1 Two sided alternative
       t.test(x,y,alternative="two.sided",var.equal = TRUE);

 #5.2 Alternative mean of x greater than mean of y
       t.test(x,y,alternative="greater",var.equal = TRUE);

 #5.3 Alternative mean of x less than mean of y
       t.test(x,y,alternative="less",var.equal = TRUE);

#6. Independent sample t-test (Unequal variance)

 #6.1 Two sided alternative
       t.test(x,y,alternative="two.sided",var.equal = FALSE);
```

```
#6.2 Alternative mean of x greater than mean of y
     t.test(x,y,alternative="greater",var.equal = FALSE);

#5.3 Alternative mean of x less than mean of y
     t.test(x,y,alternative="less",var.equal = FALSE);

#7. Non-parametric test for independent sample t-test is Wilcoxon
test.

#7.1 Two sided alternative
     wilcox.test(x,y,alternative="two.sided");

#7.2 Alternative mean of x greater than mean of y
     wilcox.test(x,y,alternative="greater");

#7.3 Alternative mean of x less than mean of y
     wilcox.test(x,y,alternative="less");
```

INTERVAL ESTIMATION

Interval estimation of the population mean can be computed from functions of the following R packages

- stats - contains the `t.test`
- TeachingDemos - contains the `z.test`;
- BSDA - contains the zsum.test and tsum.test.

Example: The data is the score of 33 random students from college of science and mathematics 84, 93, 101, 86, 82, 86, 88, 94, 89, 94, 93, 83, 95, 86, 94, 87, 91, 96, 89, 79, 99, 98, 81, 80, 88, 100, 90, 100, 81, 98, 87, 95, and 94. The population of these scores are believe to be normally distributed with 6.8 standard deviation. Determine and interpret the 95% and 99% confidence interval of the population mean.

We observe that the sample size is greater than 30 hence we apply z-test

```
scores <- c(84, 93, 101, 86, 82, 86, 88, 94, 89, 94, 93, 83, 95,
86, 94, 87, 91, 96, 89, 79, 99, 98, 81, 80, 88, 100, 90, 100, 81,
98, 87, 95, 94)

library(BSDA)
z.test(scores, sigma.x = 6.8)
```

The output is as shown below:

```
data:  scores
z = 76.3126, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
88.01327 92.65340
sample estimates:
mean of x
 90.33333
#For 99% Confidence Interval
z.test(scores, sigma.x = 6.8, conf.level = 0.99)

One-sample z-Test
data:  scores
z = 76.3126, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
99 percent confidence interval:
87.28425 93.38241
sample estimates:
mean of x
90.33333
```

CORRELATION AND REGRESSION

Correlation is the study of the relationship between two or more variables

Then R-codes for the correlation is:

```
cor(variable1,variable2)
cor.test(variable1, variable2)
cor(variable1,variable2,method=”spearman”)
```

Simple Linear Regression Analysis

Objective: Describe the relationship between two variables, say X and Y as a straight line, that is Y is modeled as a linear function of X

X: explanatory variable

Y: response variable

Consider the following example: a random sample of service call records for a computer repair operation were examined and the length of each call (in minutes) and the number of components repaired or replaced were recorded. The data is as follows:

Sl.no	minutes	units
1	23	1
2	29	2
3	49	3
4	64	4
5	74	4
6	87	5
7	96	6
8	97	6
9	109	7
10	119	8
11	149	9
12	145	9
13	154	10
14	166	10

Let X be the independent variable i.e., units and Y be the dependent variable i.e., minutes

The expected model for the data is $Y = \beta_0 + \beta_1 X$

First save the data in the text format in "E" drive of your computer with the name as rawdata

The R –code is as follows

```
data<-read.table("E:/rawdata.txt",header=TRUE);
attach(data);
```

```
plot(units, minutes)
# fit the regression model using the function lm()
data1<-lm(minutes~units, data=data)
#use the function summary() to get some results
summary(data1)
```

The output is as follows:

```
Call:
lm(formula = min ~ uni, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-9.2318 -3.3415 -0.7143  4.7769  7.8033

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.162      3.355    1.24    0.239
uni           15.509      0.505   30.71 8.92e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.392 on 12 degrees of freedom
Multiple R-squared:  0.9874,    Adjusted R-squared:  0.9864
F-statistic: 943.2 on 1 and 12 DF,  p-value: 8.916e-13
```

Conclusion: In this session we have seen how we use R-Program for inferential statistics such as t-test, wilcoxon test, correlation and Regression.