UNIT 10 INTRODUCTION TO R*

Structure

10.0 OBJECTIVES

After reading this unit, you will be able to:

- outline the features of the 'graphical user interface' of R with a brief note on installation of additional 'packages' in R;
- write the codes used in R for performing the basic mathematical operations and for generating 'matrices and data frame';
- state the code or command in R for 'importing a data file';
- describe the codes in R for obtaining the results for 'descriptive statistics' and for performing select 'tests of significance';
- illustrate the procedure for obtaining the results for 'one way ANOVA' in R;

* Dr. Sudip Mukherjee, Asst. Prof. of Economics, Dinabandhu Mahavidyalaya, Bongaon, West Bengal.

- explain the procedure for obtaining the results of 'regression' in R;
- present the details of testing for 'regression diagnostics' in R; and
- discuss the details for the testing of 'panel data regression' in R.

10.1 INTRODUCTION

R is an open source software. It is freely downloadable from the Comprehensive R Archive Network (CRAN) at http://cran.r-project.org. R Studio allows the user to run R in a user-friendly environment. R enables us to fulfil the following type of computations: (i) algebraic operations, (ii) statistical analysis and (iii) advanced visualisation of data using different graphical techniques. In this unit, we shall mainly deal with statistical analysis part of R. The unit mainly presents 'commands' or 'codes' for getting the results of 'statistical data analysis' in R. For illustrating the outputs presented by R, upon executing a command indicated, the unit uses the databases contained in R's library as well as other relevant data sets. The databases used relate to: (i) 'cars' data of R on 'distance and speed' for the results of simple regression model and test for multicollinearity, (ii) mreg data (dataset given in appendix) for results of multiple linear regression, normality test, heteroscedasticity and autocorrelation and (iii) EmplUK data of plm library (for results of panel regression).

10.2 GRAPHICAL USER INTERFACE OF R

Take a look at Fig. 10.1. The top left corner is the source window. It is used to edit the script and run it. The **R script** is where you keep a record of your work. We should write the commands on R script. On the top of the script there is a tab named Run. Run is the tab which is used to execute the command after writing in script. The window in the bottom left corner is '**console**'. The 'console' is where we see the 'output' presented by R. The 'console' can also be used to write the commands. The upper right window is the 'workplace window'. This shows all the active objects and stores all the variables used during the execution of commands. The '**history**' tab shows a list of commands used thus far. Next to it, way below, is the '**files**' tab. It shows all the 'files and folders' in the default workspace as if we are on a PC window. The '**plots**' tab shows the graphs. The above tabs, highlighted, are some of the important tabs useful in the 'user interface page' (start page) of R'.

In the R-studio (a term used to indicate the working space in R), 'packages' refer to collections of 'R functions, data and codes'. The directory is the place where packages are stored. It is called the 'library' in the 'R' environment. R comes with a standard set of packages i.e. by default R installs a set of packages (called R base) during its installation. More packages can be added-on for specific purposes.

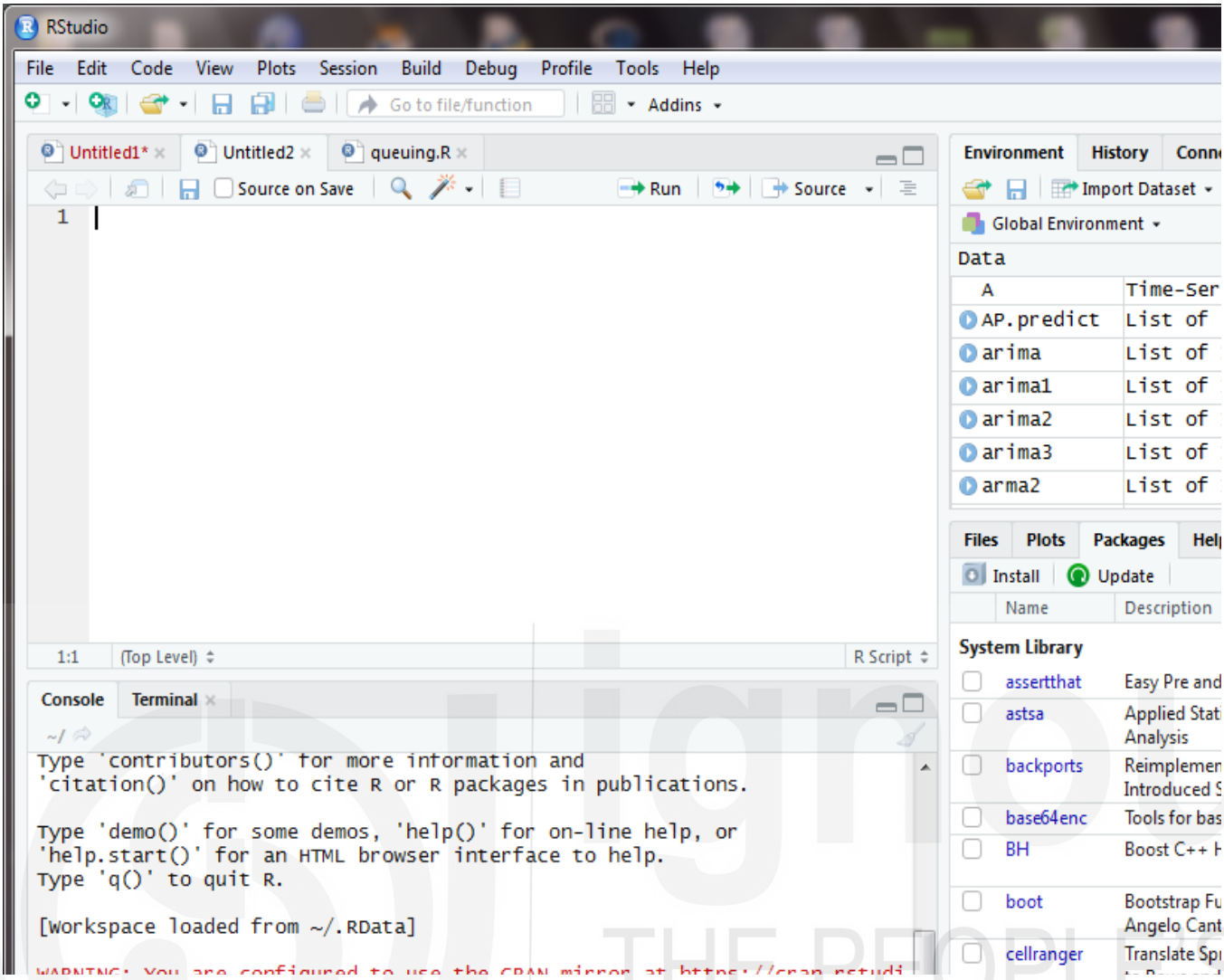

Fig. 10.1: Graphical User Interface of R Studio

When we start the R console, only the base packages are available by default. Other packages have to be explicitly 'loaded' for use by the R program. R can be extended up to 9,200+ packages available on CRAN (Comprehensive R Archive Network). Many useful 'R' functions come in packages. They are free 'libraries of codes' written by R's service community. To install an 'R' package, one needs to open an 'R' session and type in the command line:

install.packages ("<the package's name>") (10.1)

R will download the package from CRAN. Hence, one needs to be connected to the internet. Once the user has a package installed, he can use its contents in the current R session by typing/running the command:

library("<the package's name>") (10.2)

The above command (or code) will be executed by R to run the concerned package. For instance, we can install the MS EXCEL package in R by typing the command:

```
install.packages("readxl")
library("readxl")
```
(10.3)

Note that there are two lines in (10.3). Both the lines are needed to be typed in the 'R Script'. The first line 'installs' the excel package in the 'R's library'. The second line 'fetches' it from the library making it available for 'use' in the current session.

10.3 BASIC OPERATIONS AND RESULTS

R script is a plain text file in which we can write and save in R code. After opening the R studio, we first open the R script. For this, we first click on the 'file menu' in the top left corner of the opening page. We next click on 'New File' and then on 'R script'. A new R script will open. Therefore, the steps to open a new script are: File →New File→ R Script. We can use R studio as a calculator to obtain the result of any arithmetical operation. For instance, in R script we can write 4+7 and then click on run. In console window, R-studio returns the result as 11. Table 10.1 illustrates the arithmetic operator in R along with the command (or code) and result.

Table 10.1: Illustrative Codes and Result for Basic Operations in R

Description	Operator	Code	Result
Addition	+	**4+7**	**11**
Subtraction	-	**5-3**	**2**
Multiplication	*	**5*3**	**15**
Division	/	**15/3**	**5**
Exponent	^	**2^3**	**8**
Integer Division	%/%	**13%/%4**	**3**

We can create important objects in R like vector, matrix, data frame and list with simple commands. For instance, we use the c( ) function to write a vector. For example, we can create a vector named Income, say with three values 100, 120, 150, by writing in script and then clicking on run as follows: Income<-c(100,120,150). Thus, in general, for creating a vector R, the command is:

"name"<-c(x_1, x_2, …..x_n) (10.4)

Now, suppose we write similar three value entries for another vector, consumption as: consumption <–(70,90,110). Now, to subtract the 'consumption' vector from the 'income' vector to get the 'savings' vector, we use the command:

Savings<-Income-Consumption (10.5)

R will display the savings vector as: # [1] 30 30 40.

10.3.1 Matrix and Data Frame

A matrix is a rectangular array with *p rows and n columns.* An element in the *i-th row and j-th column is denoted by X[i, j], i* = 1,2,...,*n, j* = 1,2,...,*p.* In R, a 4 × 2 matrix 'X' can be created with the command*:*

>X<-matrix(nrow=4, ncol=2, data=c(1,2,3,4,5,6,7,8)) (10.6a)

The result appears in the console as follows:

```
X
[,1] [,2]
[1,] 1 5
[2,] 2 6
[3,] 3 7
[4,] 4 8
```

We can create the data frame using the following command:

data<- data.frame (Income, Consumption, Savings) (10.6_b)

data (10.6_c)

The output in console appears as:

Income	Consumption	Savings
1 100	70	30
2 120	90	30
3 150	110	40

10.3.2 Importing a Data File

The R command for importing data file needs it to be first read as:

read.<format>("file name") (10.7)

Then, to import a .csv file the command is:

read.csv(D:/mydata/test.csv, header=TRUE) (10.7.a)

Likewise, to import a .text file the command is

read.table (D:/mydata/test.txt, header=TRUE) (10.7.b)

The steps for importing an excel data file are:

Step 1: install.packages("readxl")

Step 2: library("readxl")

Step 3: File →Import→ From Excel

Step 4: click on excel tab, then click on 'browse' to select your data.

Note that giving blank space while giving command is allowed in R.

Check Your Progress 1 [answer within the space given in about 50-100 words]

1) Given, price<-(10,15,18) and quantity<-(40,35,30), compute 'expenditure'.

……

……

……

……

……

2) Create a 'data frame' using the vectors in 1 above.

……

……

……

……

……

3) State the steps to open a new script. What role a 'script' plays in R?

……

……

……

……

……

10.4 DESCRIPTIVE STATISTICS AND TESTS OF SIGNIFICANCE

We can calculate the measures of central tendency and dispersion in R using the commands given here. Let us create a vector X using the following code:

X<-c (23, 45, 29, 56, 78, 45). We can now calculate the mean of the vector X by writing the code 'mean(X)' in the script as follows:

<-mean(X) (10.8a)

After clicking the run tab the mean value appears in the console. Likewise, the codes for calculating the median and variance for the vector X are as follows:

<-median(X) (10.8_b)

<-var(X) (10.8_c)

After calculating the variance, we can calculate the standard deviation by giving the command:

B<-var(X) (10.8d)

sd<-sqrt(B) (10.8e)

The results for the sample data X on mean, median, variance and 'sd' that R produces on console along with sample data X are as below:

```
> X<-c (23, 45, 29, 56, 78, 45)
>mean(X)
[1] 46
>median(X)
[1] 45
>var(X)
[1] 388.8
> B<-var(X)
>sd<-sqrt(B)
>sd
[1] 19.71801
```

10.4.1 Independent t-Test

We use the t-test procedure to test the hypothesis that the means of two groups are not significantly different. In other words, it is used to test the equality of means among two groups. When the groups are same (e.g. before-after/pre-post) we use paired t-test. When the groups are different, we use independent t-test. Let us consider the average weight of half kg cake packets baked by two bakers X and Y. Let the weights measured for 8 independent packets be as follows:

X: 512 530 498 540 521 528 505 523

Y: 499 500 510 495 515 503 490 511

Our objective is to test that the average weight of packets is same for the two bakers. In this example, the two bakers are different and hence the test is independent t-test. So we use the command with a two-tailed alternative as follows:

X<- c(512,530,498,540,521,528,505,523)

Y<- c(499,500,510,495,515,503,490,511)

t.test (X, Y, alternative = 'two.sided',var.equal=TRUE) (10.9)

After executing the R code for t-test, the following test results are given by R (Table 10.2).

Table 10.2: Results of Independent t-test

Two Sample t-test
data: X and Y
t = 2.9058, df = 14, p-value = 0.01151
alternative hypothesis: true difference in means is ≠ 0
95 percent confidence interval:
4.386709 29.113291
sample estimates:
mean of X mean of Y
519.625 502.875

Since $p < 0.05$, we cannot say that the evidence supports the conclusion that the null is true. We therefore accept the alternative hypothesis and conclude that 'the difference in the two means is statistically significant for the difference'.

10.4.2 Paired t-Test

In a similar way, the R command for a 'paired t-test' [with the values of data on the 'test scores' of 10 students as entered] for the difference in scores for 'before and after' the consumption of a health booster beverage is as follows:

```
Y <– c(4,3,5,6,7,6,4,7,6,2)
X <– c(5,4,6,7,8,7,5,8,5,5)
t.test (xp, yp, paired=TRUE)                    (10.10)
```

The results of the paired t-test's generated by R is as in Table 10.3.

Table 10.3: Results of Paired t-test

data: X and Y
t = 3.3541, df = 9, p-value = 0.008468
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
0.325555 1.674445

The results of the paired t-test has the p-value (0.008) less than .05. Hence, we accept the alternative hypothesis. We conclude that there is a significant

difference in the points scored by the students between 'after and before'of drinking a cup of beverage.

10.4.3 One Way ANOVA

We can use the one-way ANOVA procedure to test the hypothesis that the means of two or more groups are not statistically different. To get the result in R on one-way ANOVA, we use the data on test scores in an examination. The test scores are taken as the 'dependent variable' and the type of examination administered (3 types) as the 'independent variable'. Our objective is to find out that the score obtained by the students do NOT vary due to the examination type administered. We key-in (i.e. enter) the test scores with the ANOVA command as follows:

```
type1.scores <- c(95,91,89,90,99,88,96,98,95)

type2.scores <-c(83,89,85,89,81,89,90,82,84)

type3.scores <- c(68,75,79,74,75,81,73,77,80)

Score <- c(type1.scores, type2.scores, type3.scores)

Type <- rep(c("type1", "type2", "type3"), Times=c( length(type1.scores),
length(type2.scores), length(type3.scores)))

data<- data.frame(Type, Score)

result<-aov(Score ~ Type, data = data)                    (10.11)
```

R produces the ANOVA output as in Table 10.4.

Table 10.4: Results of One Way ANOVA

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Type	2	352.7	176.33	13.01	0.000149 ***
Residuals	2	4 325.3	13.56		

Significance codes: '***' 0.001 '**' 0.01 '*' 0.05

The value of F statistic is 13.01 with the 'p' value 0.0001 which is less than .05. Hence, we reject the null hypothesis of 'no difference in test scores' and accept the alternative hypothesis. We conclude that the variable 'examination type' does make a significant difference to the scores of the students.

10.5 REGRESSION

In data analysis, we run regressions to examine the causality between the dependent and independent variables. After performing the regression analysis, we aim at concluding how much would be the response in terms of change in the value of dependent variable corresponding to a 1 unit change in

the value of the independent variable. In order to see how R can be used in regression, we first take up a two variable simple linear regression model. For data, we consider the dataset named 'cars' available in the R library. The two variables of this dataset are: speed and distance (dist). We consider 'dist' as the dependent variable and 'speed' as the independent variable. To find out the causal relationship between speed and dist, we consider the equation: dist = $\beta_0 + \beta_1$ speed + u. Before attempting the regression, we have to check for the linearity of the relationship between the dependent and the independent variable. For this, we use the 'scatter diagram'. The R command for the scatter diagram is:

scatter.smooth(x=cars$speed,y=cars$dist, main="Dist ~ Speed") (10.12)

Execution of the above command makes R generate the scatter diagram as in Fig. 10.2.

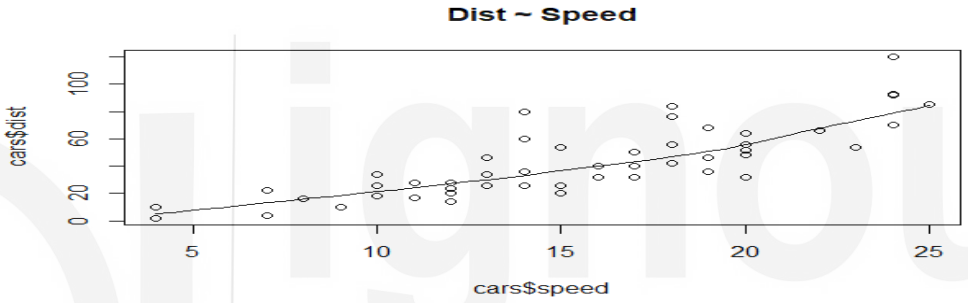

Fig. 10.2: Scatter Diagram Between Distance and Speed

It is clear from Fig. 10.2 that there is a linear relationship between distance and speed. We next calculate the correlation between the two variables using the following R command.

cor(cars$speed, cars$dist) (10.13)

The value of the correlation coefficient displayed by R is 0.80. Since the correlation coefficient of up to 0.80 is acceptable to proceed without treating for multicollinearity, we can proceed for doing the regression. In R, the command or code used for regression is 'lm()'. The following command is the R command to be used for regression.

linearMod<- lm(dist ~ speed, data=cars) (10.14_a)

summary(linearMod) (10.14_b)

anova(linearMod) (10.14_c)

The results of the above regression are presented in Table 10.5.The value of F-statistics is 89.57 with p-value zero. The model is therefore overall statistically significant. R^2=0.65 implies that 65% of the variation in distance is explained by the independent variable 'speed'. The coefficient of speed is 3.93 with p-value zero. Since the p value < 0.05, the variable 'speed' is a significant contributor to distance. Hence, if speed increases by one unit the

distance increases by 3.93 units. The regression model may therefore be written as:

'dist.' = –17.57 + 3.93 * speed.

Table 10.5: Result of Regression Analysis

```
Call:lm(formula = dist. ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
–29.069  –9.525  –2.272   9.215  43.201


Coefficients: Estimate    Std. Error  t value  Pr(>|t|)
(Intercept)  –17.5791      6.7584     –2.601   0.0123 *
Speed          3.9324      0.4155      9.464   1.49e-12 ***
---
Significance codes: '***' 0.001 '**' 0.01 '*' 0.05

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared: 0.6511, Adjusted R-squared: 0.6438
F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12
```

10.5.1 Multiple Linear Regression

To explain the multiple linear regression model, we consider the dataset named "mreg". We can import the data after saving in excel as indicated in sub-section 10.3.2. The Table A-1 (given in annexure) is the data on 'total fertility rate'(TFR) for the 20 states in India. We take TFR as the dependent variable and 'female literacy'(FLIT), location of residence (URBAN) and economic status (POV) as the three independent variables. To find out the causal relationship between the dependent and the independent variables, we consider the following equation:

$$TFR = \beta_0 + \beta_1 FLIT + \beta_2 URBAN + \beta_3 POV + u \qquad (10.15)$$

The following is the command in R to perform the multiple regression.

```
model<-lm(TFR~FLIT+URBAN+POV,data=mreg)        (10.16a)
summary (model)                                (10.16b)
```

The results of the multiple regression generated by R by running the above command is presented in Table 10.5.

Table 10.6: Results of Multiple Regression

Call:lm(formula = TFR ~ FLIT + URBAN + POV, data = mreg)

Residuals:

Min	1Q	Median	3Q	Max
–0.74293	–0.24584	0.02322	0.27228	0.71634

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.179605	0.604323	6.916	3.47e-06 ***
FLIT	–0.031897	0.008894	–3.586	0.00247 **
URBAN	–0.008654	0.010862	–0.797	0.43728
POV	0.013113	0.008279	1.584	0.13278

Significance codes: '***' 0.001 '**' 0.01 '*' 0.05

Residual standard error: 0.413 on 16 degrees of freedom

Multiple R-squared: 0.6505, Adjusted R-squared: 0.585

F-statistic: 9.927 on 3 and 16 DF, p-value: 0.0006166

The value of F-statistics is 9.93 with the p-value 0.0006 which is less than 0.05. Hence, the model is overall statistically significant. Further, R^2=0.65 implies that 65% of the variation in dependent variable is explained by the independent variables. The coefficient of FLIT is -0.032 with p-value 0.002 (which is < 0.05). Hence, FLIT is a significant variable. The negative sign to FLIT implies that TFR decreases by about 3 percent with a percentage increase in FLIT. The p-value associated with the coefficients of URBAN & POV are greater than 0.05 implying that these variables are not statistically significant in influencing the TFR.

10.5.2 Regression Diagnostics

We know that if we want the results from OLS method to hold with its 'best' properties, we should be sure that the dataset satisfies the assumptions of the 'classical linear regression models' (CLRM). Specifically, these assumptions relate to the following four:(i) there is no multicollinearity in our data, (ii) the variance of the residuals is constant (i.e. there is no heteroscedasticity), (iii) the values of the residuals are independent (i.e. there is no autocorrelation) and (iv) the values of the residuals are normally distributed. We shall now note the commands in R for testing or detecting for the presence or absence of the indicators behind these assumptions.

Test for Multicollinearity: . To detect this, we know that the 'variance inflation factor (VIF)' can be used. To compute the VIF, the following command is used in R.

```
library(car)                    (10.17a)
vif(model)                      (10.17b)
```

Execution of the above two commands, produces the output as in Table 10.6 by R.

Table 10.7 : VIF Values for Car Dataset

FLIT	URBAN	POV
1.309584	1.204745	1.202844

As a rule of thumb, a variable with VIF value greater than 2, needs further investigation. But in this example, the VIFs for the independent variables are all below 2. Thus, we can conclude that there is no serious multicollinearity in the data set.

Test for Heteroscedasticity: To check for the homogeneity of error variance the following is the R code.

```
library(lmtest)                 (10.18a)
bptest(model)                   (10.18b)
```

Execution of the above two commands in (10.18) produces the following output (Table 10.8) by R.

Table 10.8: Breusch-Pagan Test Result

Studentized Breusch-Pagan Test
data: model
BP = 1.3637, df = 3, p-value = 0.7141

The null hypothesis of the Breusch-Pagan (BP) test is 'error variance is constant' or 'there is no heteroscedasticity'. In this example, since the p-value is greater than 0.05, it suggests that there is no statistical significance to reject the null hypothesis. We accept the null and conclude that the error variance is constant.

Test for Autocorrelation: Another assumption for efficiency of the OLS estimator is that the values of the residuals are independent. To test for this, the following R command is used to check for the presence or otherwise of autocorrelation.

```
library(lmtest)                 (10.19a)
v<-dwtest(model)                (10.19b)
```

The result of DW test displayed by R is as in Table 10.9.

Table10.9: Result of Durbin-Watson Test

Durbin-Watson test
data: model
DW = 2.0656, p-value = 0.5605
alternative hypothesis: true autocorrelation is greater than 0

The value of the Durbin-Watson statistics is 2.066 with p-value greater than 0.05. Since the p-value is greater than 0.05, we have no basis to reject the null hypothesis which is 'there is NO autocorrelation'. The test result implies that there is no autocorrelation in the dataset.

Test for Normality: One of the assumptions of the OLS regression is error terms are normally distributed. To verify for this, we give the following command in R to obtain the quantile-quantile (Q-Q) plot.

plot(model) (10.20)

Fig.5.3 represents the normal Q-Q plot generated by R. When the dots are nearer to the diagonal line, then it can be concluded that the errors are distributed normally. In Q-Q plot, all the dots are close to the diagonal line. From the diagram we can conclude that the errors are normally distributed.

Check Your Progress 2 [answer within space given in about 50-100 words]

1) Write the command in R for obtaining the multiple linear regression for three independent variables as in (10.15). For this, consider that you are using the dataset stored in R library by the name 'mreg'. Also, state the expected relationship between the variables considered.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

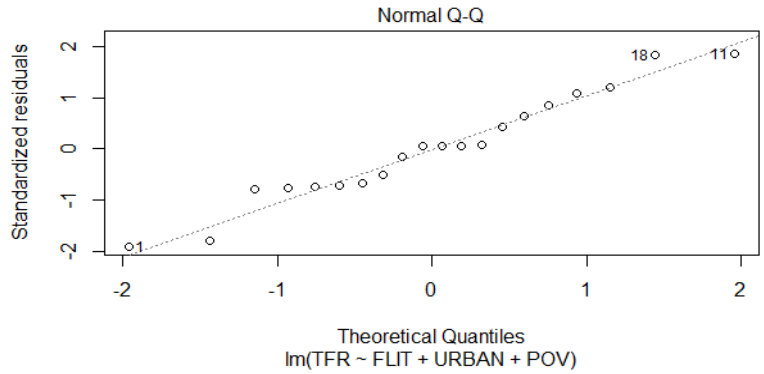

Fig. 5.3: Q-Q plot

2) In the results of the multiple linear regression obtained above, what is the significance status of FLIT? Why?

.....

.....

.....

.....

.....

10.6 PANEL DATA REGRESSION

In R, the function 'plm' is used for 'panel data regression'. We use panel data regression in data sets where both the cross section and time series dimensions are present. The elements of plm function are as follows:

Y = Dependent variable

X = independent variable

Data = data file name

Index = c("cross section id", "time series id")

Model = pooling – used for pooled regression

within- used for fixed effects model

random- used for random effects model

We write the plm function in the following form

plm(Y~X, data, index, model) (10.21)

10.6.1 Fixed Effects Model

For illustrating the results of 'panel regression', we use the 'Emp1UK' data of plm package in R. We can call for the data by using the command:

data("EmplUK") (10.22)

Suppose we are interested in finding the effect of wages on employment. Here 'employment' is the dependent variable and 'wage' is the independent variable. We 'call' the library by the command:

```
library(plm)                                                    (10.23)
```

For getting the results of panel regression for the case of the fixed effects model, the following command is to be given in R:

```
fe<-plm(emp~wage, data =EmplUK, index=c("firm","year"),
model="within")                                                 (10.24a)
summary(fe)                                                     (10.24b)
```

After clicking the run tab, the result displayed in the console by R is given in Table 10.10. The p-value of F-statistic is less than 0.05. This implies that the overall fit of the model is good. The p-value of the coefficient of wage is 0.0001 which is also less than 0.05. Hence, the variable 'wage' is a significant predictor of employment. This means, if wage increases by one unit, employment decreases by 0.13 units. The result suggests an inverse relationship between employment and wage.

Table 10.10: Results of Fixed Effects Model

```
One way (individual) effect Within Model

Call:

plm(formula = emp ~ wage, data = EmplUK, model = "within",

index = c("firm", "year"))

Unbalanced Panel: n = 140, T = 7-9, N = 1031

Residuals:

    Min.      1st Qu.       Median     3rd Qu.        Max.

-22.595206  -0.307023   0.027979   0.351118   27.454176

Coefficients:

          Estimate      Std. Error     t-value          Pr(>|t|)

wage-   0.136878      0.035533     -3.8521          0.0001255 ***

---

Significance codes: 0 '***' 0.001 '**' 0.01 '*'

Total Sum of Squares: 5030.6. Residual Sum of Squares: 4948.1

R-Squared: 0.0164.   Adj. R-Squared: -0.13832

F-statistic: 14.8391 on 1 and 890 DF. p-value: 0.00012548
```

10.6.2 Random Effects Model

Using the plm package from the R library, the codes for obtaining the random effects model results are as follows:

```
re<-plm(emp~wage, data =EmplUK, index=c("firm","year"),
model="random")                                          (10.25a)

summary(re)                                              (10.25b)
```

After clicking the run tab, the result displayed in the console is given below (Table 10.11).

Table 10.11: Results of Random Effects Model

```
One way (individual) effect Random Effect Model

 Call:

plm(formula = emp ~ wage, data = EmplUK, model = "random",

 index = c("firm", "year"))


Unbalanced Panel: n = 140, T = 7-9, N = 1031

Effects:

varstd.devshare

idiosyncratic      5.560        2.358          0.022

individual         248.638      15.768         0.978

 Min.      1st Qu.       Median     Mean      3rd Qu.       Max.

 0.9436    0.9436        0.9436     0.9450    0.9472        0.9502

Residuals:

      Min.      1st Qu.       Median      Mean      3rd Qu.      Max.

   -19.4797    -0.6047       -0.3073     -0.0075   0.1078       29.4516

Coefficients:

                Estimate      Std. Error     z-value          Pr(>|z|)

(Intercept)   11.557156     1.584263       7.2950           2.987e-13 ***

wage          -0.141571       0.035293       -4.0113          6.038e-05 ***

---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05
```

Total Sum of Squares: 5847.5
Residual Sum of Squares: 5750.7
R-Squared: 0.016588
Adj. R-Squared: 0.015633
Chi sq: 17.3224 on 1 DF, p-value: 3.1545e-05

The p-value of Chi square statistic is less than 0.05. This implies that the overall fit of the model is good. The p-value of the coefficient of 'wage' is 0.000 which is also less than 0.05. Hence, the variable 'wage' is a significant predictor of employment. This means, if wage increases by one unit, employment decreases by 0.14 units. The result suggests an inverse relationship between employment and wage.

10.6.3 Hausman Test

We can run Hausman test to decide between fixed and random effects. The null hypothesis of the Hausman test is 'the required model is of random effect'. Therefore, if the p-value of the Hausman test is greater the 0.05, then we reject the null and use the 'fixed effects' model. The code for Hausman test in R is:

"phtest()" (10.26)

By running (10.26) on our dataset, R produces the following output (Table 10.12).

Table 10.12: Results of Hausman Test

data: emp ~ wage
chisq = 1.2968, df = 1, p-value = 0.2548
alternative hypothesis: our model is inconsistent

The p-value of Hausman test is greater than 0.05. Hence, we reject the null hypothesis of 'random effect'. We conclude that we must go for the 'fixed effects' model.

10.7 LET US SUM UP

The unit has presented the R commands for performing t-test, one-way ANOVA, regression analysis, tests for normality, multicollinearity, auto correlation, heteroscedasticity and panel regressions. The codes or command for all these tests are indicated. We have used the datasets in the library of R for some of these tests. For a few others, we have entered the sample data

into the command. A summary of datasets used from the R-library, or entered sample data directly, by 'type of test' used for illustrating the various statistical tests in the unit is given in Table 10.13.

Note: The objectives indicated in Section 10.0 are generally stated. For questions in 'assignments' and 'term end exam' on these lines, you would be required to present the command in R for getting the results followed by 'presentation of the output' and comments by way of 'interpretation of the results'. For this, it is enough if major indicators (e.g. value of the test statistic, estimated co-efficient, value of p, etc.) are given. The values so given in the 'output' could be imaginary and not actual. Using them, in the interpretation of the results, you can show your analytical awareness on the expected 'direct or inverse' relationship between the variables, the justification to 'reject or not reject' the null hypothesis based on the p-value, etc. This part would therefore be not common and would carry the scope to vary between answers of different learners.

Table 10.13: Summary of Data Sets Used in the Unit

Sl. No.	Type of Test	Sample Data	Database from R Library	Sample Size
1	Independent t-Test	Yes	-	16
2	Paired t-Test	Yes	-	20
3	One Way ANOVA	Yes	-	27
4	Regression	No	'Cars'	50
5	Multiple Regression	No	mreg*	20
6	VIF Test	No	'mreg'	20
7	BP Test	No	'mreg'	20
8	DW Test	No	'mreg'	20
9	FE Model (PR)	No	'EmplUK'	1031/140 (N/n)
10	RE Model (PR)	No	'EmplUK'	1031/140 (N/n)
11	Hausman Test	No	'EmplUK'	1031/140 (N/n)

* Data set for 20 States is given in Annexure

10.8 KEY WORDS

library : This is the directory where the packages are stored. It is called as the "library" in the 'R' environment.

c() : This is the function to write a vector in R.

data.frame() : This is the function to write data frame.

t.test : This is the command for obtaining the results of t-test in R.

lm : This is the command for obtaining the results of regression in R.

plm : This is the command for obtaining the results of panel regression in R.

phtest() : This is the code for the application of Hausmann Test in R.

10.9 SUGGESTED BOOKS FOR FURTHER READING

1) Mailund, Thomas (2017). Beginning Data Science in R: Data Analysis, Visualization, and Modelling for the Data Scientist, Apress.

2) Heumann, Christian, Shalabh, Michael Schomaker (2016). Introduction to Statistics and Data Analysis with Exercises, Solutions and Applications in R, Springer.

3) Bhaumik, Sankar (2017). Principles of Econometrics, A Modern Approach Using Eviews, Oxford University Press.

10.10 ANSWERS/HINTS TO CHECK YOUR PROGRESS EXERCISES

Check Your Progress 1

1) Expenditure = price * quantity. Hence, Expenditure = <-(400,525,540)

2) <-data frame (price, quantity, expenditure)

3) File → New File → R Script. The 'R script' is where all records of our work is kept. The commands are written into the R script.

Check Your Progress 2

1) The commands in R for this multiple regression are as in (10.16). The dependent variable is the 'total fertility rate' TFR. It is assumed or hypothesized to have a linear relationship between three independent variables viz. female literary (FLIT), location of residence (URBAN) and economic status (POVERTY).

2) FLIT is a significant determinant of TFR because the p-value associated with the estimated coefficient of FLIT is < 0.05. In such situations, we cannot say there is evidence to say that the null is true (which in this case is: FLIT is NOT related to TFR). We therefore reject the null hypothesis

and accept the alternative hypothesis. This means FLIT is a significant determinant of TFR. This means, if FLIT increases, TFR decreases i.e. there is an inverse relationship between the two.

Annexure: TFR Dataset for States of India

(percent)

State	TFR	FLIT	URBAN	POV
Andhra Pradesh	1.8	50.4	27.3	15.8
Assam	2.4	54.6	12.9	19.7
Bihar	4	33.1	10.5	41.4
Chhattisgarh	2.6	51.9	20.1	40.9
Gujarat	2.4	57.8	37.4	16.8
Haryana	2.7	55.7	28.9	10
Himachal Pradesh	1.9	67.4	9.8	14
Jammu & Kashmir	2.4	43	24.8	5.4
Jharkhand	3.3	38.9	22.2	40.3
Karnataka	2.1	56.9	34	25
Kerala	1.9	87.7	26	15
Madhya Pradesh	3.1	50.3	26.5	38.3
Maharashtra	2.1	67	42.4	30.7
Orissa	2.4	50.5	15	46.4
Punjab	2	63.4	33.9	8.4
Rajasthan	3.2	43.9	23.4	22.1
Tamil Nadu	1.8	64.4	44	22.5
Uttar Pradesh	3.8	42.2	20.8	32.8
Uttarakhand	2.6	59.6	25.7	39.6
West Bengal	2.3	59.6	28	24.7

Source: Bhaumik(2017), Principles of Econometrics.