Contents

1 The R Programming Language

The R Programming Language is a statistical , data analysis , etc

R is a free software environment for statistical computing and graphics.

2 Writing R scripts

Editing your R script "R Editor".

- On the menu of the R console, click on file.
- Select open script or new script as appropriate.
- Navigate to your working directory and select your `.R` file
- A new dialogue box "`the R editor`" will open up.
- Input or select code you wish to compile.
- To compile this code, highlight it. Click the edit button on the menu.
- Select either "Run Line" or "Run Selection or All".
- Your code should now compile.
- To save your code, clink on "file" and then "`save as`".
- Save the file with the "`.R`" extension to your working directory.

3 Vector types

`R` operates on named data structures. The simplest such structure is the vector, which is a single entity consisting of an ordered collection of Numbers or characters.

- Numeric vectors
- Character vectors
- Logical vectors
- (also complex number vectors and colour vectors)

To create a vector, use the assignment operator and the concatenate function. For numeric vectors, the values are simply numbers.

```
># week8.r
>NumVec<-c(10.4,5.6,3.1,6.4)
```

Alternatively we can use the `assign()` command

For character vectors, the values are simply characters, specified with quotation marks.A logical vectors is a vector whose elements are TRUE, FALSE or NA

```
>CharVec<-c(‘‘blue", ‘‘green", ‘‘yellow")
>LogVec<-c(TRUE, FALSE)
```

4 Graphical data entry interface

`Data.entry()` is a useful command for inputting or editing data sets. Any changes are saved automatically (i.e. dont need to use the assignment operator). We can also used the `edit()` command, which calls the `R Editor`.

```
>data.entry(NumVec)
>NumVec <- edit(NumVec)
```

Another method of creating vectors is to use the following

```
numeric (length = n)
character (length = n)
logical (length = n)
```

These commands create empty vectors, of the appropriate kind, of length n. You can then use the graphical data entry interface to populate your data sets.

4.0.1 Accessing specified elements of a vector

The nth element of vector "Vec" can be accessed by specifying its index when calling "Vec".

```
>Vec[n]
```

A sequence of elements of vector "Vec" can be accessed by specifying its index when calling "Vec".

```
>Vec[l:u]
```

Omitting and deleting the nth element of vector "Vec"

```
>Vec[-n]
>Vec <- Vec[-n]
```

5 Reading data

5.1 inputting data

Concatenation

5.2 using help

?mean

5.3 Adding comments

5.4 Packages

The capabilities of R are extended through user-submitted packages, which allow specialized statistical techniques, graphical devices, as well as and import/export capabilities to many external data formats.

6 Managing Precision

- `floor()` -
- `ceiling()` -
- `round()` -
- `as.integer()` -

1cm

```
> pi
[1] 3.141593
> floor(pi)
[1] 3
> ceiling(pi)
[1] 4
> round(pi,3)
[1] 3.142
> as.integer(pi)
[1] 3
```

7 Basic Operations

7.1 Complex numbers

7.2 Trigonometric functions

8 Matrices

8.0.1 exponentials, powers and logarithms

1cm

```
>x^y
>exp(x)
>log(x)
>log(y)
#determining the square root of x
>sqrt(x)
```

8.1 vectors

1cm

```
R handles vector objects quite easily and intuitively.

> x<-c(1,3,2,10,5)    #create a vector x with 5 components
> x
[1]  1  3  2 10  5
> y<-1:5              #create a vector of consecutive integers
> y
[1] 1 2 3 4 5
```

```
> y+2                   #scalar addition
[1] 3 4 5 6 7
> 2*y                   #scalar multiplication
[1]  2  4  6  8 10
> y^2                   #raise each component to the second power
[1]  1  4  9 16 25
> 2^y                   #raise 2 to the first through fifth power
[1]  2  4  8 16 32
> y                     #y itself has not been unchanged
[1] 1 2 3 4 5
> y<-y*2
> y                     #it is now changed
[1]  2  4  6  8 10
```

8.1.1 Misc

`seq()` and `rep()` are useful commands for constructing vectors with a certain pattern.

8.2 Matrices

A matrix refers to a numeric array of rows and columns.

One of the easiest ways to create a matrix is to combine vectors of equal length using cbind(), meaning "column bind". Alternatively one can use rbind(), meaning "row bind".

8.2.1 Matrices Inversion

8.2.2 Matrices Multiplication

8.3 Data frame

A Data frame is

Descriptive Statistics

9 Basic Statistics

1cm

```
> X=c(1,4,5,7,8,9,5,8,9)
> mean(X);median(X)       #mean and median of vector
[1] 6.222222
[2] 7
> sd(X)                   #standard deviation of Vector
[1] 2.682246
> length(X)               #sample size of vector
[1] 9
> sum(X)
[1] 56
> X^2
[1]  1 16 25 49 64 81 25 64 81
> rev(X)
[1] 9 8 5 9 8 7 5 4 1
> sort(X)                 #items in ascending order
[1] 1 4 5 5 7 8 8 9 9
> X[1:5]
[1] 1 4 5 7 8
```

10 Summary Statistics

The R command `summary()` returns a summary statistics for a simple dataset. The R command `fivenum()` returns a summary statistics for a simple dataset, but without the mean. Also, the quartiles are computed a different way.

1cm

```
> summary(mtcars$mpg)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  10.40   15.43   19.20   20.09   22.80   33.90
>
> fivenum(mtcars$mpg)
[1] 10.40 15.35 19.20 22.80 33.90
```

11 Bivariate Data

1cm

```
> Y=mtcars$mpg
> X=mtcars$wt
>
> cor(X,Y)          #Correlation
[1] -0.8676594
>
> cov(X,Y)          #Covariance
[1] -5.116685
```

MTCARSboxplot.png

Figure 1: Boxplot

12 Histograms

Histograms can be created using the `hist()` command. To create a histogram of the car weights from the Cars93 data set 1cm

```
hist(mtcars$mpg, main="Histogram of MPG (Data: MTCARS) ")
```

`R` automatically chooses the number and width of the bars. We can change this by specifying the location of the break points. 1cm

```
hist(Cars93$Weight, breaks=c(1500, 2050, 2300, 2350, 2400,
2500, 3000, 3500, 3570, 4000, 4500), xlab="Weight",
main="Histogram of Weight")
```

13 Boxplot

Boxplots can be used to identify outliers.

By default, the `boxplot()` command sets the orientation as vertical. By adding the argument `horizontal=TRUE`, the orientation can be changed to horizontal. 1cm

```
boxplot(mtcars$mpg, horizontal=TRUE, xlab="Miles Per Gallon",
main="Boxplot of MPG")
```

Advanced R code

14 Data frame

A Data frame is

14.1 Merging Data frames

15 Functions

Syntax to define functions
1cm

```
myfct <- function(arg1, arg2, ...) { function_body }
```

The value returned by a function is the value of the function body, which is usually an unassigned final expression, e.g.: return()
Syntax to call functions 1cm

```
myfct(arg1=..., arg2=...)
```

16 Time and Date

It is useful . The length of time a program takes is interesting.
1cm

```
date() # returns the current system date and time
```

17 The Apply family

Sometimes want to apply a function to each element of a vector/data frame/list/array.
Four members: lapply, sapply, tapply, apply
lapply: takes any structure and gives a list of results (hence the 'l')
sapply: like lapply, but tries to simplify the result to a vector or matrix if possible (hence the 's')
apply: only used for arrays/matrices
tapply: allows you to create tables (hence the 't') of values from subgroups defined by one or more factors.

Data Visualization

18 Plots

This section is an introduction for producing simple graphs with the R Programming Language.

- Line Charts
- Bar Charts
- Histograms
- Pie Charts
- Dotcharts

18.0.1 Comparison of variances

Even though it is possible in R to perform the two-sample t test without the assumption that the variances are the same, you may still be interested in testing that assumption, and R provides the var.test function for that purpose, implementing an F test on the ratio of the group variances. It is called the same way as `t.test`:.

```
> var.test(expend~stature)
```

-
-

```
> code here
```

18.1 Charts

1cm

```
# Define 2 vectors cars <- c(1, 3, 6, 4, 9) trucks <- c(2, 5, 4,
5, 12)

# Calculate range from 0 to max value of cars and trucks g_range
<- range(0, cars, trucks)

# Graph autos using y axis that ranges from 0 to max # value in
cars or trucks vector.  Turn off axes and # annotations (axis
labels) so we can specify them ourself plot(cars, type="o",
col="blue", ylim=g_range,
   axes=FALSE, ann=FALSE)

# Make x axis using Mon-Fri labels axis(1, at=1:5,
lab=c("Mon","Tue","Wed","Thu","Fri"))
```

```
# Make y axis with horizontal labels that display ticks at # every
4 marks. 4*0:g_range[2] is equivalent to c(0,4,8,12). axis(2,
las=1, at=4*0:g_range[2])

# Create box around plot box()

# Graph trucks with red dashed line and square points
lines(trucks, type="o", pch=22, lty=2, col="red")

# Create a title with a red, bold/italic font title(main="Autos",
col.main="red", font.main=4)

# Label the x and y axes with dark green text title(xlab="Days",
col.lab=rgb(0,0.5,0)) title(ylab="Total", col.lab=rgb(0,0.5,0))

# Create a legend at (1, g_range[2]) that is slightly smaller #
(cex) and uses the same line colors and points used by # the
actual plots legend(1, g_range[2], c("cars","trucks"), cex=0.8,
   col=c("blue","red"), pch=21:22, lty=1:2);
```

18.2 Bar charts

1cm

```
# Define the cars vector with 7 values
cars <- c(1, 3, 6, 4, 9, 5, 7)
# Graph cars
barplot(cars)
```

18.3 Boxplots

18.4 Setting graphical parameters

18.5 Miscellaneous

The following code can be used to make variations of the plots.
1cm

```
# Make an empty chart
plot(1, 1, xlim=c(1,5.5), ylim=c(0,7), type="n", ann=FALSE)

# Plot digits 0-4 with increasing size and color
text(1:5, rep(6,5), labels=c(0:4), cex=1:5, col=1:5)

# Plot symbols 0-4 with increasing size and color
points(1:5, rep(5,5), cex=1:5, col=1:5, pch=0:4)
text((1:5)+0.4, rep(5,5), cex=0.6, (0:4))
```

```
# Plot symbols 5-9 with labels
points(1:5, rep(4,5), cex=2, pch=(5:9))
text((1:5)+0.4, rep(4,5), cex=0.6, (5:9))

# Plot symbols 10-14 with labels
points(1:5, rep(3,5), cex=2, pch=(10:14))
text((1:5)+0.4, rep(3,5), cex=0.6, (10:14))

# Plot symbols 15-19 with labels
points(1:5, rep(2,5), cex=2, pch=(15:19))
text((1:5)+0.4, rep(2,5), cex=0.6, (15:19))

# Plot symbols 20-25 with labels
points((1:6)*0.8+0.2, rep(1,6), cex=2, pch=(20:25))
text((1:6)*0.8+0.5, rep(1,6), cex=0.6, (20:25))
```

18.6 Lattice Graphs

18.7 setting up

Execute the following command: 1cm

```
library(lattice)
```

For information on lattice, type: 1cm

```
help(package = lattice)
```

The examples in this section are generally drawn from the R documentation and Murrell (2006).

Murrell gives three reasons for using Lattice Graphics:

They usually look better. They can be extended in powerful ways. The resulting output can be annotated, edited, and saved

18.8 3 Dimensional Graphs

How to do a 3-d graph

Statistical Analysis using R

19 Confidence Intervals

19.1 Confidence Intervals for Large Samples

19.2 Confidence Intervals for Small Samples

20 Linear Models

The Slope and Intercept 1cm

21 ANOVA

Normality Assumptions and Outliers

21.0.1 Grubbs Test for outliers

21.1 Anderson Darling Test

21.2 Normal Probability plots

21.2.1 Kolmogorov Smirnov Test

22 Matrices

22.1 Creating a matrix

Matrices can be created using the `matrix()` command. The arguments to be supplied are 1) vector of values to be entered 2) Dimensions of the matrix, specifying either the numbers of rows or columns.

Additionally you can specify if the values are to be allocated by row or column. By default they are allocated by column.

```
Vec1=c(1,4,5,6,4,5,5,7,9) # 9 elements
A=matrix(Vec,nrow=3) #3 by 3 matrix. Values assigned by column.
A
B= matrix(c(1,6,7,0.6,0.5,0.3,1,2,1),ncol=3,byrow =TRUE)
B           #3 by 3 matrix. Values assigned by row.
```

If you have assigned values by column, but require that they are assigned by row, you can use the transpose function

```
t().
t(A) # Transpose
A=t(A)
```

Another methods of creating a matrix is to "bind" a number of vectors together, either by row or by column. The commands are rbind() and cbind() respectively.

```
x1 =c(1,2) ; x2 = c(3,6)
rbind(x1,x2)
cbind(x1,x2)
```

Particular rows and columns can be accessed by specifying the row number or column number, leaving the other value blank.

```
A[1,]   # access first row of A
B[,2]   # access first column of B
```

Addition and subtractions For matrices, addition and subtraction works on an element- wise basis. The first elements of the respective matrices are added, and so on. A+B A-B

Matrix Multiplication To multiply matrices, we require a special operator for matrices; "If we just used the normal multiplication, we would get an element-wise multiplication. A * B Element-wise multiplication A We can compute crossproducts using the crossprod () command.

If only one matrix is used it crossprod(A,B) A'B crossprod(A) A'A Diagonals The diag() command is a very versatile function for using matrices. It can be used to create a diagonal matrix with elements of a vector in the principal diagonal. For an existing matrix, it can be used to return a vector containing the elements of the principal diagonal.

Most importantly, if k is a scalar, diag() will create a k x k identity matrix. Vec2=c(1,2,3) diag(Vec2) Constructs a diagonal matrix based on values of Vec2 diag(A) Returns diagonal elements of A as a vector diag(3) creates a 3 x 3 identity matrix diag(diag(A)) Diagonal matrix D of matrix A (Jacobi Method)

Determinants, Inverse Matrices and solving Linear systems To compute the determinant of a square matrix, we simply use the det() command det(A) det(B) To find the inverse of a square matrix, we use the solve() command, specifying only the matrix in question solve(A)

To solve a system of linear equations in the form Ax=b , where A is a square matrix, and b is a column vector of known values, we use the solve() command to determine the values of the unknown vector x. b=vec2 from before solve(A, b) Row and Column Statistics. Statistic on the rows and columns can easily be computed if required. rowMeans(A) Returns vector of row means. rowSums(A) Returns vector of row sums. colMeans(A) Returns vector of column means. colSums(A) Returns vector of coumn means. Eigenvalues and Eigenvectors The eigenvalues and eigenvectors can be computed using the eigen() function. A data object is created. This is a very important type of matrix analysis, and many will encounter it again in future modules. Y = eigen(A) names(Y) " $yvalaretheeigenvaluesofA"y$vec are the eigenvectors of A

? Part 2 Revision on Earlier Material " Accessing a column of a data frame " Accessing a row of a data frame

A particular row can be accessed by specifying the row index , while leaving the column index empty

Info [4,] Fourth row of "Info" is called

A sequence of rows can be accessed by specifying a sequence of rows as follows.

Info [10:15,] tenth row to fifteenth row of "Info" is called

22.2 Subsetting datasets by rows

Suppose we wish to divide a data frame into two different section. The simplest approach we can take is to create two new data sets, each assigned data from the relevant rows of the original data set.

Suppose our dataset "Info" has the dimensions of 200 rows and 4 columns. We wish to separate "Info" into two subsets , with the first and second 100 rows respectively. (We call these new subsets "Info.1" and "Info.2".)

```
Info.1 = Info[1:100,] #assigning "info" rows 1 to 100
Info.2 = Info[101:200,] #assigning "info" rows 101 to 200
```

More useful commands such as rbind() and cbind() can be used to manipulate vectors.

Part 2 Strategies for Data project

- Exploratory Data Analysis

  The first part of your report should contain some descriptive statistics and summary values. Also include some tests for normality.

- Regression You should have a data set with multiple columns, suitable for regression analysis. Familiarize yourself with the data, and decide which variable is the dependent variable.

  Also determine the independent variables that you will use as part of your analysis.

- Correlation Analysis Compute the Pearson correlation for the dependent variable with the respective independent variables. As part of your report, mention the confidence interval for the correlation estimate Choose the independent variables with the highest correlation as your candidate variables. For these independent variables, perform a series of simple linear regression procedures.

  ```
  lm(y~x1)
  lm(y~x2)
  ```

  Comment on the slope and intercept estimates and their respective p-values. Also comment on the coefficient of determination (multiple R squared). Remember to write the regression equations. Perform a series of multiple linear regressions, using pairs of candidate independent variables.

  ```
  lm(y~x1 +x2)
  lm(y~x2 +x3)
  ```

  Again, comment on the slope and intercept estimates, and their respective p-values. In this instance, compare each of the models using the coefficient of determinations. Which model explains the data best?

22.3 Analysis of residuals

Perform an analysis of regression residuals (you can pick the best regression model from last section). Are the residuals normally distributed? Histogram / Boxplot / QQ plot / Shapiro Wilk Test Also you can plot the residuals to check that there is constant variance.

```
y=rnorm(10)
x=rnorm(10)
fit1=lm(y~x)
res.fit1 = resid(fit1)
plot(res.fit1)
```

Probability Distributions

23 Generating a set of random numbers

1cm

```
rnorm(10)
```

24 The Poisson Distribution

25 The Binomial Distribution

26 Using probability distributions for simulations

27 Probability Distributions

27.1 Generate random numbers

Graphical methods

28 Scatterplots

29 Adding titles, lines, points to plots

```
library(MASS)
# Colour points and choose plotting symbols according to a levels of a factor
plot(Cars93$Weight, Cars93$EngineSize, col=as.numeric(Cars93$Type),
pch=as.numeric(Cars93$Type))

# Adds x and y axes labels and a title.
plot(Cars93$Weight, Cars93$EngineSize, ylab="Engine Size",
xlab="Weight", main="My plot")
# Add lines to the plot.
lines(x=c(min(Cars93$Weight), max(Cars93$Weight)), y=c(min(Cars93$EngineSize),
max(Cars93$EngineSize)), lwd=4, lty=3, col="green")
abline(h=3, lty=2)
abline(v=1999, lty=4)
# Add points to the plot.
```

Programming

30 Writing Functions

A simple function can be constructed as follows:

```
function_name <- function(arg1, arg2, ...){
commands
output
}
```

You decide on the name of the function. The function command shows R that you are writing a function. Inside the parenthesis you outline the input objects required and decide what to call them. The commands occur inside the .

The name of whatever output you want goes at the end of the function. Comments lines (usually a description of what the function does is placed at the beginning) are denoted by "#".

```
sf1 <- function(x){
x^2
}
```

This function is called sf1. It has one argument, called x. Whatever value is inputted for x will be squared and the result outputted to the screen. This function must be loaded into `R` and can then be called. We can call the function using:

```
sf1(x = 3)
#sf1(3)
[1] 9
To store the result into a variable x.sq
x.sq <- sf1(x = 3)
x.sq <- sf1(3)
> x.sq
[1] 9
```

Example

```
sf2 <- function(a1, a2, a3){
x <- sqrt(a1^2 + a2^2 + a3^2)
return(x)
}
```

This function is called sf2 with 3 arguments. The values inputted for a1, a2, a3 will be squared, summed and the square root of the sum calculated and stored in x. (There will be no output to the screen as in the last example.) The return command specifies what the function returns, here the value of x. We will not be able to view the result of the function unless we store it.

```
sf2(a1=2, a2=3, a3=4)
sf2(2, 3, 4) # Can't see result.
res <- sf2(a1=2, a2=3, a3=4)
res <- sf2(2, 3, 4) # Need to use this.
res
[1] 5.385165
```

We can also give some/all arguments default values.

```
mypower <- function(x, pow=2){
x^pow
}
```

If a value for the argument pow is not specified in the function call, a value of 2 is used.

```
mypower(4)
[1] 16
```

If a value for "pow" is specified, that value is used.

```
mypower(4, 3)
[1] 64
mypower(pow=5, x=2)
[1] 32
```

30.0.1 Two Sample t test

The two-sample t test is used to test the hypothesis that two samples may be assumed to come from distributions with the same mean.

The theory for the two-sample t test is not very different in principle from that of the one-sample test. Data are now from two groups, $x_{11}, ..., x_{1n1}$ and $x_{21}, ..., x_{2n2}$, which we assume are sampled from the normal distributions $N(_1, \sigma_2^1)$ and $N(_2, \sigma_2^2)$, and it is desired to test the null hypothesis $\mu_1 = \mu_2$. You then calculate

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S.E.(\bar{X}_1 - \bar{X}_2)}$$

30.0.2 slidename

–

–

```
> code here
```

30.0.3 slidename

–

–

```
> code here
```

30.0.4 slide234

The TS are ¡equation here¿ The p-values for both of these tests are 0 and so there is enough evidence to reject H_0 and conclude that both 0 and 1 are not 0, i.e. there is a significant linear relationship between x and y. Also given are the R^2 and R^2 adjusted values. Here $R^2 = SSR/SST = 0.8813$ and so 88.13% of the variation in y is being explained by x. The final line gives the result of using the ANOVA table to assess the model t.

30.0.5 slide235

In SLR, the ANOVA table tests ¡EQN¿The TS is the F value and the critical value and p-values are found in the F tables with (p - 1) and (n - p) degrees of freedom.

This output gives the p-value = 0, therefore there is enough evidence to reject H0 and conclude that there is a signicant linear relationship between y and x. The full ANOVA table can be accessed using :

¡TABLE HERE¿

30.0.6 slide236

Once the model has been tted, must then check the residuals. The residuals should be independent and normally distributed with mean of 0 and constant variance. A Q-Q plot checks the assumption of normality (can also use a histogram as in MINITAB) while a, plot of the residuals versus fitted values gives an indication as to whether the assumption of constant variance holds.

¡HISTOGRAM¿

30.0.7 slidename

```
> xbar <- 83
> sigma <- 12
> n <- 5
> sem <- sigma/sqrt(n)
> sem
[1] 5.366563
> xbar + sem * qnorm(0.025)
[1] 72.48173
> xbar + sem * qnorm(0.975)
[1] 93.51827
```

30.0.8 Testing the slope (II)

You can compute a t test for that hypothesis simply by dividing the estimate by its standard error

$$t = \frac{\hat{\beta}}{S.E.(\hat{\beta})} \tag{1}$$

which follows a t distribution on n - 2 degrees of freedom if the true β is zero.

- The standard χ^2 test in chisq.test works with data in matrix form, like fisher.test does.
- For a 2 by 2 table, the test is exactly equivalent to prop.test.

```
> chisq.test(lewitt.machin)
```

30.0.9 Chi-squared Test

A chi^2 test is carried out on tabular data containing counts, e.g. the number of animals that died, the number of days of rain, the number of stocks that grew in value, etc.

Usually have two qualitative variables, each with a number of levels, and want to determine if there is a relationship between the two variables, e.g. hair colour and eye colour, social status and crime rates, house price and house size, gender and left/right handedness.

The data are presented in a contingency table: right-handed left-handed TOTAL

	right-handed	left-handed	TOTAL
Male	43	9	52
Female	44	4	48
TOTAL	87	13	100

The hypothesis to be tested is $H0$:There is no relationship between gender and left/right-handedness $H1$:There is a relationship between gender and left/right-handedness The values that we collect from our sample are called the observed (O) frequencies (counts). Now need to calculate the expected (E) frequencies, i.e. the values we would expect to see in the table, if H0 was true.

30.0.10 Two Sample Tests

All of the previous hypothesis tests and confidence intervals can be extended to the two-sample case.

The same assumptions apply, i.e. data are normally distributed in each population and we may want to test if the mean in one population is the same as the mean in the other population, etc.

Normality can be checked using histograms, boxplots and Q-Q plots as before. The Anderson-Darling test can be used on each group of data also.

30.0.11 Implementation

This can be carried out in R by hand:

```
>obs.vals <- matrix(c(43,9,44,4), nrow=2, byrow=T)
>row.tots <- apply(obs.vals, 1, sum)
>col.tots <- apply(obs.vals, 2, sum)
>exp.vals <- row.tots%o%col.tots/sum(obs.vals)
>TS <- sum((obs.vals-exp.vals)^2/exp.vals)
>TS
>[1] 1.777415
```

R Graphics

31 E

nhancing your scatter plots

31.1 Adding lines

Previously we have used scatter plots to plot bivariate data. They were constructed using the plot() command. Recall that we can use the arguments `xlim` and `ylim` to control the vertical and horizontal range of the plots, by specifying a two element vector (min and max) for each.

Using the `abline()` command, we can add lines to our scatter plots. We specify the argument according to the type of line required. A demonstration of three types of line is provided below. Additionally we change the colour of the added lines, by specifying a colour in the `col` argument. We can also change the line type to one of four possible types, using the `lty` argument.

The line types are follows

– `lty =1` Normal full line (default)

– `lty =2` Dashed line

– `lty =3` Dotted line

– `lty =4` Dash-dot line

```
x=rnorm(10)
y=rnorm(10)
plot(x,y)
plot(x,y,xlim=c(-4,4),ylim=c(-4,4))
abline(v =0 , lty =2 )    # add a vertical dotted line (here the y-axis) to the plot
abline(h=0  ,lty =3)    # add a horizontal dotted line (here the x-axis) to the plot
abline(a=0,b=1,col="green") # add a line to your plot with intercept "a" and slope "b"
```

31.2 Changing your plot character

To change the plot character (the symbol for each covariate, we supply an additional argument to the plot() function. This argument is formulated as pch=n where n is some number. Additionally we change the colour of the characters, by specifying a colour in the col argument.

```
plot(x,y,pch=15,col="red") #Square plot symbols
plot(x,y,pch=16,col="green") #Orb plot symbols
plot(x,y,pch=17,col="mauve") #Triangular plot symbols
plot(x,y,pch=36 ,col="amber") #Dollar sign plot symbols
```

Recall that we can add new variates to an existing scatterplot using the points() function. Remember to set the vertical and horizontal limits accordingly.

```
y1 = rnorm(10); y2 = rnorm(10)
plot(x,y1, pch=8,col="purple" ,xlim=c(-5,5),ylim=c(-5,5))
points(x,y2,pch=12,col="green")
```

31.3 Adding the regression model line

The `abline()` function can be used to add a regression model line by supplying as an argument the `coef()` values for intercept and slope estimates .These estimates can be inputted directly by using both functions in conjunction.

```
Fit1 =lm(y1~x);  coef(Fit1)
abline(coef(Fit1))
```

31.4 Adding a title

It is good practice to label your scatterplots properly. You can specify the following argument

- main="Scatterplot Example", This provides the plot with a title
- sub="Subtitle", This adds a subtitle
- xlab="X variable ", This command labels the x axis
- ylab="y variable ", This command labels the y-axis

We can also add text to each margin, using the `mtext()` command. We simply require the number of the side. (1 = bottom, 2=left,3=top,4=right). We can change the colour using the col argument.

```
plot(x,y,main="Scatterplot Example",   sub="subtitle",    xlab="X variable ", ylab="y variable ")
mtext("Enhanced Scatterplot", side=4,col="red ")
```

Alternatively , we can also use the command title() to add a title to an existing scatterplot.

```
title(main="Scatterplot Example)
```

32 Combining plots

It is possible to combine two plots. We used the graphical parameters command `par()` to create an array. Often we just require two plots side by side or above and below. We simply specify the numbers of rows and columns of this array using the `mfrow` argument, passed as a vector.

```
par(mfrow=c(1,2))
plot(x,y1) # draw first plot
plot(x,y2) # draw second plot
par(mfrow=c(1,1)) # reset to default setting.
```

33 Plot of single vectors

If only one vector is specified i.e. `plot(x)`, the plot created will simply be a scatter-plot of the values of x against their indices.

$plot(x)$ Suppose we wish to examine a trend that these points represent. We can connect each covariate using a line.

$plot(x, type = "l")$ If we wish to have both lines and points, we would input the following code. This is quite useful if we wish to see how a trend develops over time. $plot(x, type = "b")$

34 Exercise

The following are measurements (in mm) of a critical dimension on a sample of twelve engine crankshafts:

```
224.120  224.001  224.017  223.982  223.989  223.961
223.960  224.089  223.987  223.976  223.902  223.980
```

(a) Calculate the mean and standard deviation for these data. (b) The process mean is supposed to be ? = 224mm. Is this the case? Give reasons for your answer. (c) Construct a 99% confidence interval for these data and interpret. (d) Check that the normality assumption is valid using 2 suitable plots.

```
> x<-c(224.120,224.001,224.017,223.982 ,223.989 ,223.961,
+ 223.960 ,224.089 ,223.987 ,223.976 , 223.902 ,223.980)
>
> mean(x)
[1] 223.997
>
> sd(x)
[1] 0.05785405
```

```
>
> t.test(x,mu=224,conf.level=0.99)

        One Sample t-test

data:  x
t = -0.1796, df = 11, p-value = 0.8607
alternative hypothesis: true mean is not equal to 224
99 percent confidence interval:
 223.9451 224.0489
sample estimates:
mean of x
  223.997
```

35 Exercise 2

The height of 12 Americans and 10 Japanese was measured. Test for a difference in the heights of both populations.

```
Americans
174.68    169.87      165.07     165.95  204.99  177.61
170.11    170.71      181.52  167.68  158.62  182.90
Japanese
158.76   168.85   159.64   180.02   164.24
161.91   163.99   152.71   157.32   147.20
```

```
> t.test(A,J)
        Welch Two Sample t-test
data:  A and J
t = 2.8398, df = 19.815, p-value = 0.01018
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  3.360121 21.996879
sample estimates:
mean of x mean of y
 174.1425  161.4640
```

36 Exercise 3

A large group of students each took two exams. The marks obtained in both exams by a sample of eight students is given below

```
Student 1 2 3 4 5 6 7 8
```

```
Exam 1 57 76 47 39 62 56 49 81
Exam 2 67 81 62 49 57 61 59 71
```

Test the hypothesis that in the group as a whole the mean mark gained did not vary according to the exam against the hypothesis that the mean mark in the second exam was higher

```
>
> Ex1<-c(57,76,47,39,62,56,49,81)
> Ex2<-c(67,81,62,49,57,61,59,71)
> t.test(Ex1-Ex2)

        One Sample t-test

data:  Ex1 - Ex2
t = -1.6733, df = 7, p-value = 0.1382
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -12.065666   2.065666
sample estimates:
mean of x
       -5
```

37 Exercise 4

A poll on social issues interviewed 1025 people randomly selected from the United States. 450 of people said that they do not get enough time to themselves. A report claims that over 41% of the population are not satisfied with personal time. Is this the case?

```
> prop.test(450,1025,p=0.40,alternative="greater")

        1-sample proportions test with continuity correction

data:  450 out of 1025, null probability 0.4
X-squared = 6.3425, df = 1, p-value = 0.005894
alternative hypothesis: true p is greater than 0.4
95 percent confidence interval:
 0.413238 1.000000
sample estimates:
        p
0.4390244
```

Exercise 23b: A company wants to investigate the proportion of males and females promoted in the last year. 45 out of 400 female candidates were promoted, while 520 out of 3270 male candidates were promoted. Is there evidence of sexism in the company?

```
> x.vec=c(45,520)
> n.vec=c(400,3270)
>  prop.test(x.vec,n.vec)

 2-sample test for equality of proportions with continuity correction

data:  x.vec out of n.vec
X-squared = 5.5702, df = 1, p-value = 0.01827
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.08133043 -0.01171238
sample estimates:
   prop 1    prop 2
0.1125000 0.1590214
```

?

38 Exercise

Generate a histogram for data set 'scores', with an accompanying box-and-whisker plot. The colour of the histogram's bar should be yellow. The orientation for the boxplot should be horizontal.

```
scores <-c(23,19,22,22,19,20,25,26,26,19,24,23,17,21,28,26)

par(mfrow=c(2,1))  # two rows , one column

hist(scores,main="Distribution of scores",xlab="scores",col="yellow")

boxplot(scores ,horizontal=TRUE)

par(mfrow =c(1,1))  #reset
```

39 The R Programming Language

The R Programming Language is a statistical , data analysis , etc

R is a free software environment for statistical computing and graphics.

40 Writing R scripts

Editing your R script "R Editor".

- On the menu of the R console, click on file.

- Select open script or new script as appropriate.
- Navigate to your working directory and select your `.R` file
- A new dialogue box "`the R editor`" will open up.
- Input or select code you wish to compile.
- To compile this code, highlight it. Click the edit button on the menu.
- Select either "Run Line" or "Run Selection or All".
- Your code should now compile.
- To save your code, clink on "file" and then "`save as`".
- Save the file with the "`.R`" extension to your working directory.

41 Vector types

`R` operates on named data structures. The simplest such structure is the vector, which is a single entity consisting of an ordered collection of Numbers or characters.

- Numeric vectors
- Character vectors
- Logical vectors
- (also complex number vectors and colour vectors)

To create a vector, use the assignment operator and the concatenate function. For numeric vectors, the values are simply numbers.

```
># week8.r
>NumVec<-c(10.4,5.6,3.1,6.4)
```

Alternatively we can use the `assign()` command

For character vectors, the values are simply characters, specified with quotation marks.A logical vectors is a vector whose elements are TRUE, FALSE or NA

```
>CharVec<-c(‘‘blue", ‘‘green", ‘‘yellow")
>LogVec<-c(TRUE, FALSE)
```

42 Graphical data entry interface

`Data.entry()` is a useful command for inputting or editing data sets. Any changes are saved automatically (i.e. dont need to use the assignment operator). We can also used the `edit()` command, which calls the `R Editor`.

```
>data.entry(NumVec)
>NumVec <- edit(NumVec)
```

Another method of creating vectors is to use the following

```
numeric (length = n)
character (length = n)
logical (length = n)
```

These commands create empty vectors, of the appropriate kind, of length n. You can then use the graphical data entry interface to populate your data sets.

42.0.1 Accessing specified elements of a vector

The nth element of vector "Vec" can be accessed by specifying its index when calling "Vec".

```
>Vec[n]
```

A sequence of elements of vector "Vec" can be accessed by specifying its index when calling "Vec".

```
>Vec[l:u]
```

Omitting and deleting the nth element of vector "Vec"

```
>Vec[-n]
>Vec <- Vec[-n]
```

43 Reading data

43.1 inputting data

Concatenation

43.2 using help

?mean

43.3 Adding comments

43.4 Packages

The capabilities of R are extended through user-submitted packages, which allow specialized statistical techniques, graphical devices, as well as and import/export capabilities to many external data formats.

44 Managing Precision

- floor() -
- ceiling() -
- round() -
- as.integer() -

```
> pi
[1] 3.141593
> floor(pi)
[1] 3
> ceiling(pi)
[1] 4
> round(pi,3)
[1] 3.142
> as.integer(pi)
[1] 3
```

45 Basic Operations

45.1 Complex numbers

45.2 Trigonometric functions

46 Matrices

46.0.1 exponentials, powers and logarithms

```
>x^y
>exp(x)
>log(x)
>log(y)
#determining the square root of x
>sqrt(x)
```

46.1 vectors

```
R handles vector objects quite easily and intuitively.

> x<-c(1,3,2,10,5)    #create a vector x with 5 components
> x
[1]  1  3  2 10  5
> y<-1:5              #create a vector of consecutive integers
> y
[1] 1 2 3 4 5
> y+2                 #scalar addition
[1] 3 4 5 6 7
> 2*y                 #scalar multiplication
[1]  2  4  6  8 10
> y^2                 #raise each component to the second power
[1]  1  4  9 16 25
> 2^y                 #raise 2 to the first through fifth power
[1]  2  4  8 16 32
> y                   #y itself has not been unchanged
[1] 1 2 3 4 5
> y<-y*2
> y                   #it is now changed
[1]  2  4  6  8 10
```

46.1.1 Misc

`seq()` and `rep()` are useful commands for constructing vectors with a certain pattern.

46.2 Matrices

A matrix refers to a numeric array of rows and columns.

One of the easiest ways to create a matrix is to combine vectors of equal length using cbind(), meaning ”column bind”. Alternatively one can use rbind(), meaning “row bind”.

46.2.1 Matrices Inversion

46.2.2 Matrices Multiplication

46.3 Data frame

A Data frame is

Descriptive Statistics

47 Basic Statistics

1cm

```
> X=c(1,4,5,7,8,9,5,8,9)
> mean(X);median(X)       #mean and median of vector
[1] 6.222222
[2] 7
> sd(X)                   #standard deviation of Vector
[1] 2.682246
> length(X)               #sample size of vector
[1] 9
> sum(X)
[1] 56
> X^2
[1]  1 16 25 49 64 81 25 64 81
> rev(X)
[1] 9 8 5 9 8 7 5 4 1
> sort(X)                 #items in ascending order
[1] 1 4 5 5 7 8 8 9 9
> X[1:5]
[1] 1 4 5 7 8
```

48 Summary Statistics

The R command **summary()** returns a summary statistics for a simple dataset. The R command **fivenum()** returns a summary statistics for a simple dataset, but without the mean. Also, the quartiles are computed a different way.

1cm

```
> summary(mtcars$mpg)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  10.40   15.43   19.20   20.09   22.80   33.90
>
> fivenum(mtcars$mpg)
[1] 10.40 15.35 19.20 22.80 33.90
```

49 Bivariate Data

1cm

```
> Y=mtcars$mpg
> X=mtcars$wt
>
> cor(X,Y)          #Correlation
[1] -0.8676594
```

```
>
> cov(X,Y)          #Covariance
[1] -5.116685
```

50 Histograms

Histograms can be created using the `hist()` command. To create a histogram of the car weights from the Cars93 data set 1cm

```
hist(mtcars$mpg, main="Histogram of MPG (Data: MTCARS) ")
```

`R` automatically chooses the number and width of the bars. We can change this by specifying the location of the break points. 1cm

```
hist(Cars93$Weight, breaks=c(1500, 2050, 2300, 2350, 2400,
2500, 3000, 3500, 3570, 4000, 4500), xlab="Weight",
main="Histogram of Weight")
```

51 Boxplot

Boxplots can be used to identify outliers.

By default, the `boxplot()` command sets the orientation as vertical. By adding the argument `horizontal=TRUE`, the orientation can be changed to horizontal.

```
boxplot(mtcars$mpg, horizontal=TRUE, xlab="Miles Per Gallon",
main="Boxplot of MPG")
```

Advanced R code

52 Data frame

A Data frame is

52.1 Merging Data frames

53 Functions

Syntax to define functions

```
myfct <- function(arg1, arg2, ...) { function_body }
```

The value returned by a function is the value of the function body, which is usually an unassigned final expression, e.g.: return()

Syntax to call functions

```
myfct(arg1=..., arg2=...)
```

53.1 Time and Date

It is useful . The length of time a program takes is interesting.

```
date() # returns the current system date and time
```

54 The Apply family

Sometimes want to apply a function to each element of a vector/data frame/list/array.
Four members: lapply, sapply, tapply, apply
lapply: takes any structure and gives a list of results (hence the 'l')
sapply: like lapply, but tries to simplify the result to a vector or matrix if possible (hence the 's')
apply: only used for arrays/matrices
tapply: allows you to create tables (hence the 't') of values from subgroups defined by one or more factors.

55 Plots

This section is an introduction for producing simple graphs with the R Programming Language.

- Line Charts
- Bar Charts
- Histograms
- Pie Charts
- Dotcharts

55.0.1 Comparison of variances

Even though it is possible in R to perform the two-sample t test without the assumption that the variances are the same, you may still be interested in testing that assumption, and R provides the var.test function for that purpose, implementing an F test on the ratio of the group variances. It is called the same way as `t.test`:.

```
> var.test(expend~stature)
```

-
-

```
> code here
```

55.1 Charts

1cm

```
# Define 2 vectors cars <- c(1, 3, 6, 4, 9) trucks <- c(2, 5, 4,
5, 12)

# Calculate range from 0 to max value of cars and trucks g_range
<- range(0, cars, trucks)

# Graph autos using y axis that ranges from 0 to max # value in
cars or trucks vector.  Turn off axes and # annotations (axis
labels) so we can specify them ourself plot(cars, type="o",
col="blue", ylim=g_range,
   axes=FALSE, ann=FALSE)

# Make x axis using Mon-Fri labels axis(1, at=1:5,
lab=c("Mon","Tue","Wed","Thu","Fri"))

# Make y axis with horizontal labels that display ticks at # every
4 marks. 4*0:g_range[2] is equivalent to c(0,4,8,12). axis(2,
las=1, at=4*0:g_range[2])
```

```
# Create box around plot box()

# Graph trucks with red dashed line and square points
lines(trucks, type="o", pch=22, lty=2, col="red")

# Create a title with a red, bold/italic font title(main="Autos",
col.main="red", font.main=4)

# Label the x and y axes with dark green text title(xlab="Days",
col.lab=rgb(0,0.5,0)) title(ylab="Total", col.lab=rgb(0,0.5,0))

# Create a legend at (1, g_range[2]) that is slightly smaller #
(cex) and uses the same line colors and points used by # the
actual plots legend(1, g_range[2], c("cars","trucks"), cex=0.8,
   col=c("blue","red"), pch=21:22, lty=1:2);
```

55.2 Bar charts

1cm

```
# Define the cars vector with 7 values
cars <- c(1, 3, 6, 4, 9, 5, 7)
# Graph cars
barplot(cars)
```

55.3 Boxplots

55.4 Setting graphical parameters

55.5 Miscellaneous

The following code can be used to make variations of the plots.
1cm

```
# Make an empty chart
plot(1, 1, xlim=c(1,5.5), ylim=c(0,7), type="n", ann=FALSE)

# Plot digits 0-4 with increasing size and color
text(1:5, rep(6,5), labels=c(0:4), cex=1:5, col=1:5)

# Plot symbols 0-4 with increasing size and color
points(1:5, rep(5,5), cex=1:5, col=1:5, pch=0:4)
text((1:5)+0.4, rep(5,5), cex=0.6, (0:4))

# Plot symbols 5-9 with labels
```

```
points(1:5, rep(4,5), cex=2, pch=(5:9))
text((1:5)+0.4, rep(4,5), cex=0.6, (5:9))

# Plot symbols 10-14 with labels
points(1:5, rep(3,5), cex=2, pch=(10:14))
text((1:5)+0.4, rep(3,5), cex=0.6, (10:14))

# Plot symbols 15-19 with labels
points(1:5, rep(2,5), cex=2, pch=(15:19))
text((1:5)+0.4, rep(2,5), cex=0.6, (15:19))

# Plot symbols 20-25 with labels
points((1:6)*0.8+0.2, rep(1,6), cex=2, pch=(20:25))
text((1:6)*0.8+0.5, rep(1,6), cex=0.6, (20:25))
```

55.6 Lattice Graphs

55.7 setting up

Execute the following command: 1cm

```
library(lattice)
```

For information on lattice, type: 1cm

```
help(package = lattice)
```

The examples in this section are generally drawn from the R documentation and Murrell (2006).

Murrell gives three reasons for using Lattice Graphics:

They usually look better. They can be extended in powerful ways. The resulting output can be annotated, edited, and saved

55.8 3 Dimensional Graphs

How to do a 3-d graph

Statistical Analysis using R

56 Confidence Intervals

56.1 Confidence Intervals for Large Samples

56.2 Confidence Intervals for Small Samples

57 Linear Models

The Slope and Intercept 1cm

58 ANOVA

Normality Assumptions and Outliers

58.0.1 Grubbs Test for outliers

58.1 Anderson Darling Test

58.2 Normal Probability plots

58.2.1 Kolmogorov Smirnov Test

59 Matrices

59.1 Creating a matrix

Matrices can be created using the `matrix()` command. The arguments to be supplied are 1) vector of values to be entered 2) Dimensions of the matrix, specifying either the numbers of rows or columns.

Additionally you can specify if the values are to be allocated by row or column. By default they are allocated by column.

```
Vec1=c(1,4,5,6,4,5,5,7,9) # 9 elements
A=matrix(Vec,nrow=3) #3 by 3 matrix. Values assigned by column.
A
B= matrix(c(1,6,7,0.6,0.5,0.3,1,2,1),ncol=3,byrow =TRUE)
B           #3 by 3 matrix. Values assigned by row.
```

If you have assigned values by column, but require that they are assigned by row, you can use the transpose function

```
t().
t(A) # Transpose
A=t(A)
```

Another methods of creating a matrix is to "bind" a number of vectors together, either by row or by column. The commands are rbind() and cbind() respectively.

```
x1 =c(1,2) ; x2 = c(3,6)
rbind(x1,x2)
cbind(x1,x2)
```

Particular rows and columns can be accessed by specifying the row number or column number, leaving the other value blank.

```
A[1,]   # access first row of A
B[,2]   # access first column of B
```

Addition and subtractions For matrices, addition and subtraction works on an element-wise basis. The first elements of the respective matrices are added, and so on. A+B A-B

Matrix Multiplication To multiply matrices, we require a special operator for matrices; "If we just used the normal multiplication, we would get an element-wise multiplication. A * B Element-wise multiplication A We can compute crossproducts using the crossprod () command. If only one matrix is used it

```
crossprod(A,B)  # A'B
crossprod(A)  # A'A
```

Diagonals The diag() command is a very versatile function for using matrices. It can be used to create a diagonal matrix with elements of a vector in the principal diagonal. For an existing matrix, it can be used to return a vector containing the elements of the principal diagonal.

Most importantly, if k is a scalar, diag() will create a k x k identity matrix.

```
Vec2=c(1,2,3)
diag(Vec2) # Constructs a diagonal matrix based on values of Vec2
diag(A) #       Returns diagonal elements of A as a vector
diag(3) # creates a 3 x 3 identity matrix
diag(diag(A)) #   Diagonal matrix D of matrix A ( Jacobi Method)
```

Determinants, Inverse Matrices and solving Linear systems To compute the determinant of a square matrix, we simply use the det() command det(A) det(B) To find the inverse of a square matrix, we use the solve() command, specifying only the matrix in question solve(A)

To solve a system of linear equations in the form Ax=b , where A is a square matrix, and b is a column vector of known values, we use the solve() command to determine the values of the unknown vector x. b=vec2 from before solve(A, b) Row and Column Statistics. Statistic on the rows and columns can easily be computed if required.

59.2 Eigenvalues and Eigenvectors

The eigenvalues and eigenvectors can be computed using the eigen() function. A data object is created. This is a very important type of matrix analysis, and many will encounter it again in future modules.

Y = eigen(A) names(Y) " $yval are the eigenvalues of A$" yvec are the eigenvectors of A

? Part 2 Revision on Earlier Material " Accessing a column of a data frame " Accessing a row of a data frame

A particular row can be accessed by specifying the row index , while leaving the column index empty

Info [4,] Fourth row of "Info" is called

A sequence of rows can be accessed by specifying a sequence of rows as follows.

Info [10:15,] tenth row to fifteenth row of "Info" is called

59.3 Subsetting datasets by rows

Suppose we wish to divide a data frame into two different section. The simplest approach we can take is to create two new data sets, each assigned data from the relevant rows of the original data set.

Suppose our dataset "Info" has the dimensions of 200 rows and 4 columns. We wish to separate "Info" into two subsets , with the first and second 100 rows respectively. (We call these new subsets "Info.1" and "Info.2".)

```
Info.1 = Info[1:100,] #assigning "info" rows 1 to 100
Info.2 = Info[101:200,] #assigning "info" rows 101 to 200
```

More useful commands such as rbind() and cbind() can be used to manipulate vectors.

Part 2 Strategies for Data project

– Exploratory Data Analysis

  The first part of your report should contain some descriptive statistics and summary values. Also include some tests for normality.

– Regression You should have a data set with multiple columns, suitable for regression analysis. Familiarize yourself with the data, and decide which variable is the dependent variable.

  Also determine the independent variables that you will use as part of your analysis.

– Correlation Analysis Compute the Pearson correlation for the dependent variable with the respective independent variables. As part of your report, mention the confidence interval for the correlation estimate Choose the independent variables with the highest correlation as your candidate variables. For these independent variables, perform a series of simple linear regression procedures.

  ```
  lm(y~x1)
  lm(y~x2)
  ```

  Comment on the slope and intercept estimates and their respective p-values. Also comment on the coefficient of determination (multiple R squared). Remember to write the regression equations. Perform a series of multiple linear regressions, using pairs of candidate independent variables.

  ```
  lm(y~x1 +x2)
  lm(y~x2 +x3)
  ```

  Again, comment on the slope and intercept estimates, and their respective p-values. In this instance, compare each of the models using the coefficient of determinations. Which model explains the data best?

59.4 Analysis of residuals

Perform an analysis of regression residuals (you can pick the best regression model from last section). Are the residuals normally distributed? Histogram / Boxplot / QQ plot / Shapiro Wilk Test Also you can plot the residuals to check that there is constant variance.

```
y=rnorm(10)
x=rnorm(10)
fit1=lm(y~x)
res.fit1 = resid(fit1)
plot(res.fit1)
```

Probability Distributions

60 Generating a set of random numbers

1cm

```
rnorm(10)
```

61 The Poisson Distribution

62 The Binomial Distribution

63 Using probability distributions for simulations

64 Probability Distributions

64.1 Generate random numbers

MTCARSmpgwt.png

Figure 2: Scatterplot

Graphical methods

65 Scatterplots

66 Adding titles, lines, points to plots

```
library(MASS)
# Colour points and choose plotting symbols according to a levels of a factor
plot(Cars93$Weight, Cars93$EngineSize, col=as.numeric(Cars93$Type),
pch=as.numeric(Cars93$Type))

# Adds x and y axes labels and a title.
plot(Cars93$Weight, Cars93$EngineSize, ylab="Engine Size",
xlab="Weight", main="My plot")
# Add lines to the plot.
lines(x=c(min(Cars93$Weight), max(Cars93$Weight)), y=c(min(Cars93$EngineSize),
max(Cars93$EngineSize)), lwd=4, lty=3, col="green")
abline(h=3, lty=2)
abline(v=1999, lty=4)
# Add points to the plot.
```

Programming

67 Writing Functions

A simple function can be constructed as follows:

```
function_name <- function(arg1, arg2, ...){
commands
output
}
```

You decide on the name of the function. The function command shows R that you are writing a function. Inside the parenthesis you outline the input objects required and decide what to call them. The commands occur inside the .
The name of whatever output you want goes at the end of the function. Comments lines (usually a description of what the function does is placed at the beginning) are denoted by "#".

```
sf1 <- function(x){
x^2
}
```

This function is called sf1. It has one argument, called x. Whatever value is inputted for x will be squared and the result outputted to the screen. This function must be loaded into `R` and can then be called. We can call the function using:

```
sf1(x = 3)
#sf1(3)
[1] 9
To store the result into a variable x.sq
x.sq <- sf1(x = 3)
x.sq <- sf1(3)
> x.sq
[1] 9
```

Example

```
sf2 <- function(a1, a2, a3){
x <- sqrt(a1^2 + a2^2 + a3^2)
return(x)
}
```

This function is called sf2 with 3 arguments. The values inputted for a1, a2, a3 will be squared, summed and the square root of the sum calculated and stored in x. (There will be no output to the screen as in the last example.) The return command specifies what the function returns, here the value of x. We will not be able to view the result of the function unless we store it.

```
sf2(a1=2, a2=3, a3=4)
sf2(2, 3, 4) # Can't see result.
res <- sf2(a1=2, a2=3, a3=4)
res <- sf2(2, 3, 4) # Need to use this.
res
[1] 5.385165
```

We can also give some/all arguments default values.

```
mypower <- function(x, pow=2){
x^pow
}
```

If a value for the argument pow is not specified in the function call, a value of 2 is used.

```
mypower(4)
[1] 16
```

If a value for "pow" is specified, that value is used.

```
mypower(4, 3)
[1] 64
mypower(pow=5, x=2)
[1] 32
```

67.0.1 Two Sample t test

The two-sample t test is used to test the hypothesis that two samples may be assumed to come from distributions with the same mean.
The theory for the two-sample t test is not very different in principle from that of the one-sample test. Data are now from two groups, $x_{11}, ..., x_{1n1}$ and $x_{21}, ..., x_{2n2}$, which we assume are sampled from the normal distributions $N(_1, \sigma_2^1)$ and $N(_2, \sigma_2^2)$, and it is desired to test the null hypothesis $\mu_1 = \mu_2$. You then calculate

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S.E.(\bar{X}_1 - \bar{X}_2)}$$

67.0.2 slidename

- *
- *

```
> code here
```

67.0.3 slidename

- *
- *

```
> code here
```

67.0.4 slide234

The TS are ¡equation here¿ The p-values for both of these tests are 0 and so there is enough evidence to reject H_0 and conclude that both 0 and 1 are not 0, i.e. there is a significant linear relationship between x and y. Also given are the R^2 and R^2 adjusted values. Here $R^2 = SSR/SST = 0.8813$ and so 88.13% of the variation in y is being explained by x. The final line gives the result of using the ANOVA table to assess the model t.

67.0.5 slide235

In SLR, the ANOVA table tests ¡EQN¿The TS is the F value and the critical value and p-values are found in the F tables with (p - 1) and (n - p) degrees of freedom.

This output gives the p-value = 0, therefore there is enough evidence to reject H0 and conclude that there is a signicant linear relationship between y and x. The full ANOVA table can be accessed using :
¡TABLE HERE¿

67.0.6 slide236

Once the model has been tted, must then check the residuals. The residuals should be independent and normally distributed with mean of 0 and constant variance. A Q-Q plot checks the assumption of normality (can also use a histogram as in MINITAB) while a, plot of the residuals versus fitted values gives an indication as to whether the assumption of constant variance holds.
¡HISTOGRAM¿

67.0.7 slidename

```
> xbar <- 83
> sigma <- 12
> n <- 5
> sem <- sigma/sqrt(n)
> sem
[1] 5.366563
> xbar + sem * qnorm(0.025)
[1] 72.48173
> xbar + sem * qnorm(0.975)
[1] 93.51827
```

67.0.8 Testing the slope (II)

You can compute a t test for that hypothesis simply by dividing the estimate by its standard error

$$t = \frac{\hat{\beta}}{S.E.(\hat{\beta})} \tag{2}$$

which follows a t distribution on n - 2 degrees of freedom if the true β is zero.

- ∗ The standard χ^2 test in chisq.test works with data in matrix form, like fisher.test does.
- ∗ For a 2 by 2 table, the test is exactly equivalent to prop.test.

```
> chisq.test(lewitt.machin)
```

67.0.9 Chi-squared Test

A chi^2 test is carried out on tabular data containing counts, e.g. the number of animals that died, the number of days of rain, the number of stocks that grew in value, etc.

Usually have two qualitative variables, each with a number of levels, and want to determine if there is a relationship between the two variables, e.g. hair colour and eye colour, social status and crime rates, house price and house size, gender and left/right handedness.

The data are presented in a contingency table: right-handed left-handed TOTAL

	right-handed	left-handed	TOTAL
Male	43	9	52
Female	44	4	48
TOTAL	87	13	100

The hypothesis to be tested is $H0$:There is no relationship between gender and left/right-handedness $H1$:There is a relationship between gender and left/right-handedness The values that we collect from our sample are called the observed (O) frequencies (counts). Now need to calculate the expected (E) frequencies, i.e. the values we would expect to see in the table, if H0 was true.

67.0.10 Two Sample Tests

All of the previous hypothesis tests and confidence intervals can be extended to the two-sample case.

The same assumptions apply, i.e. data are normally distributed in each population and we may want to test if the mean in one population is the same as the mean in the other population, etc.

Normality can be checked using histograms, boxplots and Q-Q plots as before. The Anderson-Darling test can be used on each group of data also.

67.0.11 Implementation

This can be carried out in R by hand:

```
>obs.vals <- matrix(c(43,9,44,4), nrow=2, byrow=T)
>row.tots <- apply(obs.vals, 1, sum)
>col.tots <- apply(obs.vals, 2, sum)
>exp.vals <- row.tots%o%col.tots/sum(obs.vals)
>TS <- sum((obs.vals-exp.vals)^2/exp.vals)
>TS
>[1] 1.777415
```

R Graphics

68 E

nhancing your scatter plots

68.1 Adding lines

Previously we have used scatter plots to plot bivariate data. They were constructed using the plot() command. Recall that we can use the arguments `xlim` and `ylim` to control the vertical and horizontal range of the plots, by specifying a two element vector (min and max) for each.

Using the `abline()` command, we can add lines to our scatter plots. We specify the argument according to the type of line required. A demonstration of three types of line is provided below. Additionally we change the colour of the added lines, by specifying a colour in the `col` argument. We can also change the line type to one of four possible types, using the `lty` argument.

The line types are follows

∗ `lty =1` Normal full line (default)

∗ `lty =2` Dashed line

∗ `lty =3` Dotted line

∗ `lty =4` Dash-dot line

```
x=rnorm(10)
y=rnorm(10)
plot(x,y)
plot(x,y,xlim=c(-4,4),ylim=c(-4,4))
abline(v =0 , lty =2 )    # add a vertical dotted line (here the y-axis) to the plot
abline(h=0  ,lty =3)    # add a horizontal dotted line (here the x-axis) to the plot
abline(a=0,b=1,col="green") # add a line to your plot with intercept "a" and slope "b"
```

68.2 Changing your plot character

To change the plot character (the symbol for each covariate, we supply an additional argument to the plot() function. This argument is formulated as pch=n where n is some number. Additionally we change the colour of the characters, by specifying a colour in the col argument.

```
plot(x,y,pch=15,col="red") #Square plot symbols
plot(x,y,pch=16,col="green") #Orb plot symbols
plot(x,y,pch=17,col="mauve") #Triangular plot symbols
plot(x,y,pch=36 ,col="amber") #Dollar sign plot symbols
```

Recall that we can add new variates to an existing scatterplot using the points() function. Remember to set the vertical and horizontal limits accordingly.

```
y1 = rnorm(10); y2 = rnorm(10)
plot(x,y1, pch=8,col="purple" ,xlim=c(-5,5),ylim=c(-5,5))
points(x,y2,pch=12,col="green")
```

68.3 Adding the regression model line

The `abline()` function can be used to add a regression model line by supplying as an argument the `coef()` values for intercept and slope estimates .These estimates can be inputted directly by using both functions in conjunction.

```
Fit1 =lm(y1~x);  coef(Fit1)
abline(coef(Fit1))
```

68.4 Adding a title

It is good practice to label your scatterplots properly. You can specify the following argument

- main="Scatterplot Example", This provides the plot with a title
- sub="Subtitle", This adds a subtitle
- xlab="X variable ", This command labels the x axis
- ylab="y variable ", This command labels the y-axis

We can also add text to each margin, using the `mtext()` command. We simply require the number of the side. (1 = bottom, 2=left,3=top,4=right). We can change the colour using the col argument.

```
plot(x,y,main="Scatterplot Example",   sub="subtitle",    xlab="X variable ", ylab="y variabl
mtext("Enhanced Scatterplot", side=4,col="red ")
```

Alternatively , we can also use the command title() to add a title to an existing scatterplot.

```
title(main="Scatterplot Example)
```

69 Combining plots

It is possible to combine two plots. We used the graphical parameters command `par()` to create an array. Often we just require two plots side by side or above and below. We simply specify the numbers of rows and columns of this array using the `mfrow` argument, passed as a vector.

```
par(mfrow=c(1,2))
plot(x,y1) # draw first plot
plot(x,y2) # draw second plot
par(mfrow=c(1,1)) # reset to default setting.
```

70 Plot of single vectors

If only one vector is specified i.e. `plot(x)`, the plot created will simply be a scatterplot of the values of x against their indices.

plot(x) Suppose we wish to examine a trend that these points represent. We can connect each covariate using a line.

plot$(x, type = "l")$ If we wish to have both lines and points, we would input the following code. This is quite useful if we wish to see how a trend develops over time. *plot*$(x, type = "b")$

71 Exercise

The following are measurements (in mm) of a critical dimension on a sample of twelve engine crankshafts:

```
224.120  224.001  224.017  223.982  223.989  223.961
223.960  224.089  223.987  223.976  223.902  223.980
```

(a) Calculate the mean and standard deviation for these data. (b) The process mean is supposed to be ? = 224mm. Is this the case? Give reasons for your answer. (c) Construct a 99% confidence interval for these data and interpret. (d) Check that the normality assumption is valid using 2 suitable plots.

```
> x<-c(224.120,224.001,224.017,223.982 ,223.989 ,223.961,
+ 223.960 ,224.089 ,223.987 ,223.976 , 223.902 ,223.980)
>
> mean(x)
[1] 223.997
>
> sd(x)
[1] 0.05785405
>
> t.test(x,mu=224,conf.level=0.99)

        One Sample t-test

data:  x
t = -0.1796, df = 11, p-value = 0.8607
alternative hypothesis: true mean is not equal to 224
99 percent confidence interval:
 223.9451 224.0489
sample estimates:
mean of x
  223.997
```

72 Exercise 2

The height of 12 Americans and 10 Japanese was measured. Test for a difference in the heights of both populations.

```
Americans
```

```
174.68    169.87      165.07     165.95  204.99  177.61
170.11    170.71      181.52  167.68  158.62  182.90
Japanese
158.76   168.85   159.64   180.02   164.24
161.91   163.99   152.71   157.32   147.20
```

```
> t.test(A,J)
        Welch Two Sample t-test
data:  A and J
t = 2.8398, df = 19.815, p-value = 0.01018
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  3.360121 21.996879
sample estimates:
mean of x mean of y
 174.1425  161.4640
```

73 Exercise 3

A large group of students each took two exams. The marks obtained in both exams by a sample of eight students is given below

```
Student 1 2 3 4 5 6 7 8
Exam 1 57 76 47 39 62 56 49 81
Exam 2 67 81 62 49 57 61 59 71
```

Test the hypothesis that in the group as a whole the mean mark gained did not vary according to the exam against the hypothesis that the mean mark in the second exam was higher

```
>
> Ex1<-c(57,76,47,39,62,56,49,81)
> Ex2<-c(67,81,62,49,57,61,59,71)
> t.test(Ex1-Ex2)

        One Sample t-test

data:  Ex1 - Ex2
t = -1.6733, df = 7, p-value = 0.1382
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -12.065666   2.065666
sample estimates:
mean of x
       -5
```

74 Exercise 4

A poll on social issues interviewed 1025 people randomly selected from the United States. 450 of people said that they do not get enough time to themselves. A report claims that over 41% of the population are not satisfied with personal time. Is this the case?

```
> prop.test(450,1025,p=0.40,alternative="greater")

        1-sample proportions test with continuity correction

data:  450 out of 1025, null probability 0.4
X-squared = 6.3425, df = 1, p-value = 0.005894
alternative hypothesis: true p is greater than 0.4
95 percent confidence interval:
 0.413238 1.000000
sample estimates:
        p
0.4390244
```

Exercise 23b: A company wants to investigate the proportion of males and females promoted in the last year. 45 out of 400 female candidates were promoted, while 520 out of 3270 male candidates were promoted. Is there evidence of sexism in the company?

```
> x.vec=c(45,520)
> n.vec=c(400,3270)
>  prop.test(x.vec,n.vec)

 2-sample test for equality of proportions with continuity correction

data:  x.vec out of n.vec
X-squared = 5.5702, df = 1, p-value = 0.01827
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.08133043 -0.01171238
sample estimates:
   prop 1    prop 2
0.1125000 0.1590214
```

?

75 Exercise

Generate a histogram for data set 'scores', with an accompanying box-and-whisker plot. The colour of the histogram's bar should be yellow. The orientation for the boxplot should be horizontal.

```
scores <-c(23,19,22,22,19,20,25,26,26,19,24,23,17,21,28,26)

par(mfrow=c(2,1))  # two rows , one column

hist(scores,main="Distribution of scores",xlab="scores",col="yellow")

boxplot(scores ,horizontal=TRUE)

par(mfrow =c(1,1))  #reset
```

76 Introduction to R

R consists of a base package and many additional packages R was originally designed as a command language. Commands were typed into a text-based input area on the computer screen and the program responded with a response to each command. The R console opens with information and then a prompt mark "¿" it is ready to accept commands R is an open source software package, meaning that the code written to implement the various functions can be freely examined and modified. R can be installed free of charge from the R-project website.

77 Vector Operations

* *R* operates on named data structures. The simplest such structure is the vector, which is a single entity consisting of an ordered collection of numbers or characters.
* The most common types of vectors are:
  - Numeric vectors
  - Character vectors
  - Logical vectors
* There are, of course, other types of vectors.
  - Colour vectors - potentially useful later on.
  - Order vectors - The rankings of items in a vector.
  - Complex number vectors - not part of this course.

77.1 Ordering Vector Operations

```
sort(x)  # sort x into ascending order
rev(x)
rev(sort(x))
```

```
x=c(15, 34, 7, 12, 18, 9, 1, 42, 56, 28, 13, 24, 35)

length(x)          # How many items in x
median(x)          # median of data set x
sort(x)[7]         # 7th item when x is in ascending order
quantile(x,0.75)   # Compute the third quartile
quantile(x,0.25)   # Compute the first quartile
IQR(x)
fivenum(x)

# code is committed
```

78 Some Useful Operations

78.1 Sampling

The `sample()` function.

78.2 Set Theory Operations

78.3 Controlling Precision and Integerization

```
pi
round(pi,3)
round(pi,2)
floor(pi)
ceiling(pi)
```

79 Important Introductory Topics

79.1 The `head()` and `tail()` functions

79.2 Randomly Generated Numbers

With a and b as the lower and upper bound of the continous uniform distribution.

$$X \sim U(a,b)$$

79.3 The `as` and `is` families of functions

79.4 The `apply` family of functions

79.5 Writing your own function

80 Matrices

80.1 Creating a matrix

Matrices can be created using the `matrix()` command. The arguments to be supplied are

1. vector of values to be entered
2. Dimensions of the matrix, specifying either the numbers of rows or columns.

Additionally you can specify if the values are to be allocated by row or column. By default they are allocated by column.

```
Vec1=c(1,4,5,6,4,5,5,7,9) # 9 elements
A=matrix(Vec,nrow=3) #3 by 3 matrix. Values assigned by column.
A
B= matrix(c(1,6,7,0.6,0.5,0.3,1,2,1),ncol=3,byrow =TRUE)
B           #3 by 3 matrix. Values assigned by row.
```

If you have assigned values by column, but require that they are assigned by row, you can use the transpose function `t()`.

```
t(A) # Transpose
A=t(A)
```

Another method of creating a matrix is to bind a number of vectors together, either by row or by column. The commands are `rbind()` .and `cbind()` respectively.

```
x1 =c(1,2) ; x2 = c(3,6)
rbind(x1,x2)
cbind(x1,x2)
```

81 Lists and Data Frames

81.1 Lists

81.2 Named Components

81.3 Data Frames

```
framename = data.frame()
```

82 Important Graphical Procedures

1. Histograms
2. Box-plots
3. Scatter-plots

83 Testing The Assumption of Normality

The null hypothesis of both the ‘Anderson-Darling and ‘Shapiro-Wilk tests is that the population is normally distributed, and the alternative hypothesis is that the data is not normally distributed.

83.1 Anderson-Darling Test

83.2 Shapiro-Wilk Test

As with the Anderson-Darling test, the null and alternative hypothesis are :
H_0 : The data set is normally distributed.
H_1 : The data set is **not** normally distributed.

```
#Generate 100 normally distributed random numbers

NormDat = rnorm(100)

shapiro.test(NormDat)
```

Sample output, using the randomly generated `NormDat` data set, is as follows:

```
> shapiro.test(NormDat)

        Shapiro-Wilk normality test
```

```
data:  NormDat
W = 0.9864, p-value = 0.4003
```

Here, the p-value is well above the 0.05 threshold. Hence we **fail to reject** the null hypothesis, and may proceed to treat the `NormDat` data set as normally distributed.

83.3 The Normal Probability (QQ) plot

```
#Generate 100 normally distributed random numbers

NormDat = rnorm(100)

qqnorm(NormDat)
qqline(NormDat)
```

83.4 Transforming the Data

Sometimes when we get non-normal data, we can change the scale of our data i.e. transform it to get a normal distribution. One transformation that often works for positively skewed data is the natural logarithm (ln) transformation.

In such a case, we work with the natural logarithms of the data set, rather than the data itself.

84 Inference Procedures

The two main types of inference procedures are **Hypothesis Tests** and **Confidence Intervals**.

There are two ways of conducting a hypothesis test. One method is to compute the test statistic, and compare to the critical values.

The second method is to compute the probability value (i.e. p-value), and compare it to the significance level. Nearly all computer programs use the p-value approach. In this course we will focus on the p-value approach.

84.1 Significance Level

In hypothesis testing, the significance level is the criterion used for rejecting the null hypothesis. The significance level is used in hypothesis testing as follows: First, the difference between the results of the experiment and the null hypothesis is determined. Then, assuming the null hypothesis is true, the probability of a difference that large or larger is computed . Finally, this probability is compared to the significance level. If the probability is less than or equal to the significance level, then the null hypothesis is rejected and the outcome is said to be statistically significant.

Traditionally, experimenters have used either the 0.05 level (sometimes called the 5% level) or the 0.01 level (1% level), although the choice of levels is largely subjective. The lower the significance level, the more the data must diverge from the null hypothesis to be significant. Therefore, the 0.01 level is more conservative than the 0.05 level. The Greek letter alpha (α) is sometimes used to indicate the significance level

84.2 The probability value

The probability value (sometimes called the $p-value$) is the probability of obtaining a statistic as different from or more different from the parameter specified in the null hypothesis as the statistic obtained in the experiment. The precise meaning of the p-value

There is often confusion about the precise meaning of the probability computed in a significance test. The convention in hypothesis testing is that the null hypothesis (Ho) is assumed to be true.

The difference between the statistic computed in the sample and the parameter specified by the null hypothesis is computed and the probability of obtaining a difference this large or large is calculated. This probability value is the probability of obtaining data as extreme or more extreme than the current data (assuming the null hypothesis is true).

It is not the probability of the null hypothesis itself. Thus, if the probability value is 0.005, this does not mean that the probability that the null hypothesis is either true or false is .005. It means that the probability of obtaining data as different or more different from the null hypothesis as those obtained in the experiment is 0.005.

The inferential step to conclude that the null hypothesis is false goes as follows: The data (or data more extreme) are very unlikely given that the null hypothesis is true. This means that:

(1) a very unlikely event occurred or

(2) the null hypothesis is false.

The inference usually made is that the null hypothesis is false. (Importantly it doesnt prove the null hypothesis to be false)

84.3 Using P-values to reject the null hypothesis

According to one view of hypothesis testing, the significance level should be specified before any statistical calculations are performed. Then, when the p-value is computed from a significance test, it is compared with the significance level. The null hypothesis is rejected if p-value is at or below the significance level; it is not rejected if p-value is above the significance level. The degree to which p ends up being above or below the significance level does not matter. The null hypothesis either is or is not rejected at the previously stated significance level.

Thus, if an experimenter originally stated that he or she was using the $= 0.05$ significance level and p-value was subsequently calculated to be 0.042, then the person would reject the null hypothesis at the 0.05 level. If p-value had been 0.0001 instead of 0.042 then the null hypothesis would still be rejected at the 0.05 significance level.

The experimenter would not have any basis to be more confident that the null hypothesis was false with a p-value of 0.0001 than with a p-value of 0.041. Similarly, if the p had been 0.051 then the experimenter would fail to reject the null hypothesis He or she would have no more basis to doubt the validity of the null hypothesis than if p-value had been 0.482. The conclusion would be that the null hypothesis could not be rejected at the 0.05 level.

In short, this approach is to specify the significance level in advance and use p-value only to determine whether or not the null hypothesis can be rejected at the stated significance level.

Many statisticians and researchers find this approach to hypothesis testing not only too rigid, but basically illogical. It is very reasonable to have more confidence that the null hypothesis is false with a p-value of 0.0001 then with a p-value of 0.042? The less likely the obtained results (or more extreme results) under the null hypothesis, the more confident one should be that the null hypothesis is false.

The null hypothesis should not be rejected once and for all. The possibility that it was falsely rejected is always present, and, all else being equal, the lower the p-value, the lower this possibility.

According to this view, research reports should not contain the p-value, only whether or not the values were significant (at or below the significance level).

However it is much more reasonable to just report the p-values. That way each reader can make up his or her mind about just how convinced they are that the null hypothesis is false.

84.4 Sample Size

For Students t distribution, statistical tables such (e.g. Murdoch Barnes and State Examinations Commision tables) only tabulate quantiles with degrees of freedom of less than 30. This restraint has given rise to the convention that a sample of size greater than 30 is a 'large sample and in this case the standard normal distribution should be used.

However there is a disparity between the Z value and the correct t value. For a sample size of 61 (i.e. degrees of freedom =60), the 97.5% t-quantiles of Student's t distribution is 2.003, and not 1.96.

However, statistical software is free from this restraint. The correct distribution will be automatically used. The Students t distribution will be used in all appropriate cases. As the sample size increases the Student t distribution converges with the standard normal distribution.

84.5 Commonly Used Inference Procedures

- Hypothesis test for the mean of a single sample
- Hypothesis test for the mean of two independent samples
- Hypothesis test for the proportion of a single group
- Hypothesis test for the proportions of two independent samples

84.6 Hypothesis test for the mean of a single sample

This procedure is used to assess whether the population mean μ has a specified value, based on the sample mean. The hypotheses are conventionally written in a form similar to below (here the hypothesized population mean is simply zero).

Ho : $\mu = 0$
Ha : $\mu neq 0$

There are two hypothesis test for the mean of a single sample.

1) The sample is of a normally-distributed variable for which the population standard deviation (σ) is known.
2) The sample is of a normally-distributed variable where σ is estimated by the sample standard deviation (s).

In practice, the population parameter values is rarely known. For this reason, we will consider the second case only in this course.

84.7 Hypothesis test for the means of two independent samples

The procedure associated with testing a hypothesis concerning the difference between two population means is similar to that for testing a hypothesis concerning the value of one population mean. The procedure differs only in that the standard error of the difference between the means is used to determine the test statistic associated with the sample result. For two tailed tests, the null hypothesis states that the population means are the same, with the alternative stating that the population means are not equal.

Ho : $\mu_1 = \mu_2$
Ha : $\mu_1 \neq \mu_2$

84.8 Hypothesis test of Proportion

This procedure is used to assess whether an assumed proportion is supported by evidence. For two tailed tests, the null hypothesis states that the population proportion has a specified value, with the alternative stating that has a different value.
The hypotheses are typically as follows:

Ho : $\pi = 0.50$
Ha : $\pi \neq 0.50$

84.8.1 Example

A manufacturer is interested in whether people can tell the difference between a new formulation of a soft drink and the original formulation. The new formulation is cheaper to produce so if people cannot tell the difference, the new formulation will be manufactured.
A sample of 100 people is taken. Each person is given a taste of both formulations and asked to identify the original. Sixty-two percent of the subjects correctly identified the new formulation. Is this proportion significantly different from 50%?
The first step in hypothesis testing is to specify the null hypothesis and an alternative hypothesis. In testing proportions, the null hypothesis is that π, the proportion in the population, is equal to 0.5. The alternate hypothesis is $\pi \neq 0.5$.
The computed p-values is compared to the pre-specified significance level of 5%. Since the p-value (0.0214) is less than the significance level of 0.05, the effect is statistically significant.

```
> prop.test(62,100,0.5)

        1-sample proportions test with continuity correction

data:  62 out of 100, null probability 0.5
X-squared = 5.29, df = 1, p-value = 0.02145
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.5170589 0.7136053
sample estimates:
   p
0.62
```

Since the effect is significant, the null hypothesis is rejected. It is concluded that the proportion of people choosing the original formulation is greater than 0.50.

This result might be described in a report as follows:

> The proportion of subjects choosing the original formulation (0.62) was significantly greater than 0.50, with p-value = 0.021.

84.9 Correlation and Regression tests

84.9.1 Correlation Coefficient

Strength of a linear relationship between X and Y

```
M=1000
CorrData=numeric(M)
for (i in 1:M)
{
CorrData[i] = cor(rnorm(10),rnorm(10))
}
```

The null hypothesis is that the correlation coefficient is zero.

The alternative hypothesis is that the correlation coefficients is greater than zero.

The slope and intercept estimates

These tests are given in the "Two Tailed" format. The one tailed format compares a null hypothesis where the parameter of interest has a true value of less than or equalt to one versus an alternative hypothesis stating that it has a value greater than zero.

85 Outliers

85.1 Grubb's Test for Outliers

```
library(outliers)
grubbs.test(X)
```

85.2 Dixon Test for Outliers

85.3 Outliers on Boxplots

Boxplots can used to determine an outlie (we will refer to them as "boxplot outliers")

```
boxplot(X, horizontal = TRUE)
```

86 Inference Procedures

86.1 Confidence Interval

A confidence interval gives an estimated range of values which is likely to include an unknown population parameter, the estimated range being calculated from a given set of sample data. If independent samples are taken repeatedly from the same population, and a confidence interval calculated for each sample, then a certain percentage (confidence level) of the intervals will include the unknown population parameter.

Confidence intervals are usually calculated so that this percentage is 95%, but we can produce 90%, 99%, 99.9% (or whatever) confidence intervals for the unknown parameter. The width of the confidence interval gives us some idea about how uncertain we are about the unknown parameter. A very wide interval may indicate that more data should be collected before anything very definite can be said about the parameter.

86.2 Power

The power of a statistical hypothesis test measures the test's ability to reject the null hypothesis when it is actually false - that is, to make a correct decision. In other words, the power of a hypothesis test is the probability of not committing a type II error. It is calculated by subtracting the probability of a type II error from 1, usually expressed as:

$$\text{Power} = 1 - \text{P(type II error)} \ = 1 - \beta$$

The maximum power a test can have is 1, the minimum is 0. Ideally we want a test to have high power, close to 1.

87 Single Sample Inference Procedures

If we have a single sample we might want to answer several questions:

* What is the mean value?
* Is the mean value significantly different from current theory? (Hypothesis test)
* What is the level of uncertainty associated with our estimate of the mean value? (Confidence interval)
* (Last week : confidence interval for a mean)
* Revision: For large samples ($n > 30$) and/or if the population standard deviation (σ) is known, the usual test statistic is given by:

$$Z = \frac{\bar{X} - \mu}{SE(\bar{X})}$$

* $S.E.(\bar{X}) = \frac{\sigma}{\sqrt{n}}$ or $\frac{s}{\sqrt{n}}$.

* For small samples, use the $t-$distribution with $n-1$ degrees of freedom.
* Critical value from tables.
* Compare test statistics and critical values.

To ensure that our analysis is correct we need to check for outliers in the data (i.e. boxplots) and we also need to check whether the data are normally distributed or not.

```
> t.test(X,mu=10)

        One Sample t-test

data:  X
t = 14.1421, df = 4, p-value = 0.0001451
alternative hypothesis: true mean is not equal to 10
95 percent confidence interval:
 10.08037 10.11963
sample estimates:
mean of x
     10.1
```

88 Test for Equality of Variance and Means

* Test for Equality of Test (`var.test()`)
* Welch Two Sample *t*-test (`t.test()`)
* Independent Two Sample *t*-test (`t.test(var.equal=TRUE)`)

88.1 Bartlett's test for Homogeneity of Variances

Equal variances across samples is called homogeneity of variances. Bartlett's test is used to test if multiple samples have equal variances.

Some statistical tests, such as the analysis of variance, assume that variances are equal across groups or samples. The Bartlett test can be used to verify that assumption.

* The null hypothesis is that each of the samples have equal variance.
* The alternative hypothesis states that at least one sample has a significantly different variance.

89 Non-Parametric Inference Procedures

Nonparametric procedures were developed to be used in cases when the distribution of the variable of interest in the population is known to be not-normal, and furthermore the distribution is undetermined (hence the name nonparametric).

Nonparametric tests are also referred to as ***distribution-free*** tests. These tests have the obvious advantage of not requiring the assumption of normality or the assumption of homogeneity of variance. They compare medians rather than means and, as a result, if the data have one or two outliers, their influence is negated.

Parametric tests are preferred because, in general, for the same number of observations, they are more likely to lead to the rejection of a false hull hypothesis. That is, they have more power. This greater power stems from the fact that if the data have been collected at an interval or ratio level, information is lost in the conversion to ranked data (i.e., merely ordering the data from the lowest to the highest value).

∗ Kolmogorov- Smirnov Test (`ks.test()`)

∗ Wilcoxon test (`wilcox.test()`)

89.1 Kolmogorov-Smirnov Test

For a single sample of data, the Kolmogorov-Smirnov test is used to test whether or not the sample of data is consistent with a specified distribution function. (Not part of this course) When there are two samples of data, it is used to test whether or not these two samples may reasonably be assumed to come from the same distribution. The null and alternative hypotheses are as follows:
H0: The two data sets are from the same distribution
H1: The data sets are not from the same distribution

Consider two sample data sets X and Y that are bothnormally distributed with similar means and variances.

```
> X=rnorm(16,mean=20,sd=5)
> Y=rnorm(18,mean=21,sd=4)
> ks.test(X,Y)

        Two-sample Kolmogorov-Smirnov test

data:  X and Y
D = 0.2153, p-value = 0.7348
alternative hypothesis: two-sided
```

Remark: It doesnt not suffice that both datasets are from the same distribution. They must have the same value for the defining parameters. Consider the case of data sets; X and Z. Both are normally distributed, but with different mean values.

```
> X=rnorm(16,mean=20,sd=5)
> Z=rnorm(16,mean=14,sd=5)
> ks.test(X,Z)

        Two-sample Kolmogorov-Smirnov test
```

```
data:  X and Z
D = 0.5625, p-value = 0.0112
alternative hypothesis: two-sided
```

89.2 Wilcoxon Mann-Whitney Test

The Wilcoxon Mann-Whitney Test is one of the most powerful of the nonparametric tests for comparing two populations. It is used to test the null hypothesis that two populations have identical distribution functions against the alternative hypothesis that the two distribution functions differ only with respect to ***location*** (i.e. median), if at all. The Wilcoxon Mann-Whitney test does not require the assumption that the differences between the two samples are normally distributed. In many applications, the Wilcoxon Mann-Whitney Test is used in place of the two sample t-test when the normality assumption is questionable. This test can also be applied when the observations in a sample of data are ranks, that is, ordinal data rather than direct measurements.

90 Programming Paradigms

90.1 While Loops

The while loop can be used if the number of iterations required is not known beforehand. For example, if we want to continue looping until a certain condition is met, a while loop is useful.
The following is the syntax for a while loop:

```
while (condition){
   command
   command
}
```

The loop continues while `condition == TRUE`.
Note: `sample()` takes a sample of the specified size (here just one) from a range of values (here integers 1 to 100).

```
#initialise a counter to zero
niter = 0
#initialize an empty vector
numvec = numeric()


num = sample(1:100, 1)

#while loop
while(num != 20)
   {
   num = sample(1:100, 1)
   niter = niter + 1
   numvec = c(numvec,num)
   }
numvec
niter
```

90.2 Nested Loops

90.3 Sums of two dice rolls

```
#Set Up an Empty Matrix of 6 rows and 6 columns
Dice = matrix(0,6,6)
```

```
#Main Loop
for(i in 1:6)
{
    #Nested Loop
for(j in 1:6)
{
Dice[i,j] = i+j
}
}
Dice   # Print your Results
```

90.4 Correlation Structure Example

91 Correlation and Simple Regression Models

91.1 Correlation

A correlation coefficient is a number between -1 and 1 which measures the degree to which two variables are linearly related. If there is perfect linear relationship with positive slope between the two variables, we have a correlation coefficient of 1; if there is positive correlation, whenever one variable has a high (low) value, so does the other.

If there is a perfect linear relationship with negative slope between the two variables, we have a correlation coefficient of -1; if there is negative correlation, whenever one variable has a high (low) value, the other has a low (high) value. A correlation coefficient of 0 means that there is no linear relationship between the variables.

We can determine the Pearson Correlation coefficient in R using the `cor()` command. To get a more complete statistical analysis, with formal tests, we can use the command `cor.test()` The interpretation of the output from the cor.test()procedure is very similar to procedures we have already encountered. The null hypothesis is that the correlation coefficient is equal to zero. This is equivalent to saying that there is no linear relationship between variables.

```
C=c(0,2,4,6,8,10,12)
F=c(2.1,5.0,9.0,12.6,17.3,21.0,24.7)
cor.test(C,F)
```

```
        Pearson's product-moment correlation

data:  C and F
t = 47.1967, df = 5, p-value = 8.066e-08
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9920730 0.9998421
sample estimates:
      cor
0.9988796
```

91.2 Spearman and Kendall Correlation

Spearman and Kendall correlations are both ***rank correlations***. To implement Spearman and Kendall correlation, simply specify the type in the `method=" "` argument.

```
> cor(G,D)
[1] 0.3167869
>
> cor(G,D,method="spearman")
```

```
[1] 0.1785714
>
> cor(G,D,method="kendall")
[1] 0.1428571
>
```

The interpretation is very similar, but there are no confidence intervals for the estimates.

91.3 Fitting a Regression Model

A regression model is fitted using the `lm()` command.
Consider the response variable F and predictor variable C.

```
C=c(0,2,4,6,8,10,12)
F=c(2.1,5.0,9.0,12.6,17.3,21.0,24.7)
Fit1=lm(F~C)
```

91.4 Confidence Interval for Regression Estimate

To compute the confidence intervals for both estimates, we use the `confint()` command, specifying the name of the fitted model.

```
coef(Fit1)
# (Intercept)        Conc
     1.517857    1.930357

confint(Fit1)
#                2.5 %   97.5 %
# (Intercept) 0.75970 2.276014
# Conc        1.82522 2.035495
```

91.5 Confidence and Prediction Intervals for Fitted Values

Recall that a fitted value $\hat{Y}$ is a estimate for the response variable, as determined by a linear model. The difference between the observed value and the corresponding fitted value is known as the residual.

The ***residual standard error*** is the conditional standard deviation of the dependent variable Y given a value of the independent variable X. The calculation of this standard error follows from the definition of the residuals.

The residual standard error is often called the root mean square error (RMSE), and is a measure of the differences between values predicted by a model or an estimator and the values actually observed from the thing being modelled or estimated.

Since the residual standard error is a good measure of accuracy, it is ideal if it is small.

91.5.1 Prediction Intervals

In contrast to a confidence interval, which is concerned with estimating a population parameter, a prediction interval is concerned with estimating an individual value and is therefore a type of probability interval.

The complete standard error for a prediction interval is called the standard error of forecast, and it includes the uncertainty associated with the vertical scatter about the regression line plus the uncertainty associated with the position of the regression line value itself.

92 Multiple Linear Regression and Variable Selection

92.1 Multiple Linear Regression

In your future studies, you will come across multiple linear regression (MLR). This is a linear model uses multiple independent variables to explain a single dependent variable.The implementation is very similar to simple linear regression (SLR). All that is required is to specify the additional independent variables.

Multiple regression analysis is an extension of simple regression analysis, as described previously, to applications involving the use of two or more independent variables (predictors) to estimate the value of the dependent variable (response variable).

The basic model for multiple regression analysis is

$$y = b_0 + b_1x_1 + \cdots + b_kx_k + e$$

92.2 The Coefficient of Determination

The coefficient of determination, R^2, is a measure of the proportion of variability explained by, or due to the regression (linear relationship) in a sample of bivariate (i.e. X v Y) data. It is a number between zero and one and a value close to zero suggests a poor model.

A very high value of R^2 can arise even though the relationship between the two variables is non-linear. The fit of a model should never simply be judged from the R^2 value.

In the case of simple linear regression (i.e. bivariate data) the coefficient of determination is equivalent to the square of the correlation coefficient of X and Y. In the case of MLR, the coefficient of determination is derived from ***Sums of Squares Identities***.

The R^2 value is presented as part of the output of the summary command for a fitted model.

92.3 Adjusted R^2

Adjusted R^2 is used to compensate for the addition of independent variables to the model. As more independent variables are added to the regression model, unadjusted R^2 will generally increase but there will never be a decrease. This will occur even when the additional variables do little to help explain the dependent variable.

To compensate for this, adjusted R^2 is corrected for the number of independent variables in the model. The result is an adjusted R2 than can go up or down depending on whether the addition of another variable adds or does not add to the explanatory power of the model. Adjusted R2 will always be lower than unadjusted.

The adjusted R^2 is also presented in the output of the summary of a fitted model. It has become standard practice to report the adjusted R^2, especially when there are multiple models presented with varying numbers of independent variables.

92.4 Akaike Information Criterion

The Akaike's information criterion (AIC), is a model selection metric. For a series of candidate fitted models, the model with a lowest AIC value is treated the best. To compute the AIC for a candidate model in `R`, simply specify the name of the model as an argument to the `AIC()` function.

```
AIC(fit1)
AIC(fit2)
```

92.5 Influence Analysis

92.5.1 Outlier

In linear regression, an outlier is an observation with large residual. In other words, it is an observation whose dependent-variable value is unusual given its values on the predictor variables. An outlier may indicate a sample peculiarity or may indicate a data entry error or other problem.

92.5.2 Leverage

An observation with an extreme value on a predictor variable is a point with high leverage. Leverage is a measure of how far an independent variable deviates from its mean. These leverage points can have an effect on the estimate of regression coefficients.

92.5.3 Influence

An observation is said to be influential if removing the observation substantially changes the estimate of coefficients. Influence can be thought of as the product of leverage and outlier-ness.

92.6 Diagnostic Plots

Homoscedascity (constant variance) is one of the assumptions required in a regression analysis in order to make valid statistical inferences about population relationships. Homoscedasticity requires that the variance of the residuals are constant for all fitted values, indicated by a uniform scatter or dispersion of data points about the trend line (i.e. "The Zero Line"). From the above plot, we can conclude that the constant variance assumption is valid. We can also see that the mean value of the residuals is zero.

```
plot(fit1)
#Four Diagnostic Plots are printed to screen sequentially.
```

93 Working with Categorical Data

93.1 Chi-Square

The table below shows the relationship between gender and party identification in a US state.

Test for association between gender and party affiliation at two appropriate levels and comment on your results.

Set out the null hypothesis that there is no association between method of computation and gender against the alternative, that there is. Be careful to get these the correct way round!

H0: There is no association. H1: There is an association.

Work out the expected values. For example, you should work out the expected value for the number of males who use no aids from the following: (95/195) 22 = 10.7.

94 Probability Distributions

94.1 Discrete Probability Distribution

* Poisson Distribution
* Binomial Distribution
* Geometric Distribution

The two most accessible discrete distributions are the binomial and ***Poisson*** distributions The binomial distribution is characterized by the number of trials, which in `R` is denoted as `size` rather than n, and the probability of success `prob`.

```
rbinom(n=5,size=100,prob=0.25)          #generate five random numbers
```

The Poisson distribution is characterized the by the arrival rate lambda.

```
rpois(n=5,lambda=4)          #generate five random numbers
```

94.1.1 Simple population study

Suppose a small island has population 1,000 at the start of a decade. The birth rate on this island is expected to 25 births per annum, while there is on average 10 deaths. Forecast the population after five years.

```
Base = 1000
Births =rpois(5,25)
Deaths=rpois(5,10)
```

```
Yrly.Incr =Births - Deaths
Increase =cumsum(Yrly.Incr)
Popn = Base + Births +Deaths
```

94.2 Continuous Probability Distributions

The continuous uniform distribution is commonly used in simulation. The Normal distribution

The normal distribution is perhaps the most widely known distribution.

```
rnorm(n=15)       #15 random numbers, mean  = 0 , std. deviation = 1
rnorm(n=15,mean= 17)       #set the mean to 17
rnorm(n=15,mean= 17,sd=4)       #set the standard deviation to 4
rnorm(15,17,4)                #argument matching : default positions
```

The exponential distribution