

Postgraduate Research Methods and Analysis

Christopher J. Wilson

2020-09-03

Contents

1	Welcome to the module	5
2	Module Overview	7
2.1	Module timetable and delivery for 2020/2021	8
2.2	Learning and Teaching Strategies	8
2.3	How will the online sessions work?	8
3	Assessment	9
3.1	Formative assessment	9
3.2	Summative Assessments	10
3.3	Submitting work	11
3.4	A note about referencing	11
4	Statistical analysis software	13
5	Academic Support and Guidance	15
5.1	Module Team Contact Details	15
6	Appendix 1: Considering your thesis	17
7	Introduction to R and R Studio	19
7.1	By the end of this section, you should be able to:	19
7.2	Why learn / use R?	19
7.3	R has many advantages	20
7.4	Download R and R Studio	20
7.5	The R Studio environment	21
7.6	Working with a script	22
7.7	Installing and loading packages	24
8	Working with data in R	27
8.1	By the end of this section, you will be able to:	27
8.2	In this section, we will use the Tidyverse set of packages	27
8.3	Import data into R from excel, SPSS and csv files	28
8.4	Understanding objects in R	28

8.5	Identify different data structures and variable types	30
8.6	Working with dataframes	32

Chapter 1

Welcome to the module

Chapter 2

Module Overview

This Level 7 module for first year Doctorate in Clinical Psychology trainees aims to enable you to:

- Refresh and extend your knowledge, skills and critical understanding of advanced research methods using both qualitative and quantitative approaches;
- Creatively apply the principles of quantitative and qualitative research methods to clinical psychology research and practice;
- Refresh and extend your skills in project design, management, analysis and presentation.

The module is also designed to explicitly prepare you for the two Doctorate level research modules which occur in Years Two and Three of the programme, ensuring that you have the requisite knowledge and skills to successfully engage with those modules.

The key foci for this module include:

- critical review of established literature
- project design
- project management
- data analysis
- dissemination of research findings

The module is taught using a variety of techniques to best enhance your knowledge and understanding of the application of research theory and methods in the context of clinical psychology. These include lectures, seminars, guided statistical analysis and tutorials with the latter being used to provide individual guidance and formative feedback. The module has its own site on the University's Virtual Learning Environment <http://eat.tees.ac.uk> - known as Blackboard), with resources and literature designed to support learning.

2.1 Module timetable and delivery for 2020/2021

The timetable for this module appears on the Clinical Psychology Programme Site/Timetables. The majority of the sessions will take place on Monday mornings. Please note that the timetable should be checked on a regular basis.

Due to COVID19 pandemic, the sessions for this module will be delivered online in 2020/21

2.2 Learning and Teaching Strategies

The module is taught using a variety of techniques to best enhance your knowledge and understanding of the application of research theory and methods in the context of clinical psychology. This includes activities designed to encourage independent learning, a key skill for successful performance in research modules in Years Two and Three of the programme. Evidence of independent learning is expected in the assignments for this module. Specific links are made with research informed activity in practice.

You will be provided with two papers for critical review at the start of the module and asked to decide which one you will use for your summative critical review assignment; one of these papers is from a quantitative research tradition and the other is from a qualitative research tradition.

All presentations (with added annotations) are available, along with additional support materials, via an e-learning@tees on the VLE. E-learning is enabled through group activities on the VLE or Microsoft Teams where discussion and problem solving is undertaken in relation to tasks set during teaching sessions. The discussion boards or Microsoft Teams site will be used to ask and answer questions that arise from the taught material and also your independent work.

2.3 How will the online sessions work?

The video below explains how online sessions will work:

Chapter 3

Assessment

3.1 Formative assessment

Formative feedback is provided throughout the module through practical exercises and in seminars on trainee presentations.

By the end of year one, trainees are expected to have identified a thesis topic and have a completed research proposal. As such, there are a number of formative milestones across the year that will be monitored by the module team. Please see Appendix 1 of this guide for details.

The required format for thesis research proposals can be found in Appendix 2 of the DClinPsy Programme Research Handbook.

The formal formative assessment is of a presentation of the thesis research proposal to be presented during the research panels, which take place on 20, 21 and 22 July 2020. The presentation will be 20 minutes' long and will outline the thesis project that the trainee will develop in Years 2 and 3. There will also be 10 minutes allotted for questions from the panel which will have two academic members and one clinical member. This formative assessment is intended as a starting point for the Year 2 and 3 research methods modules. The timing is important as it should enable trainees to start the process of ethics approval for their dissertation. earlier. The trainees will hand in a printed copy of their slides with explanatory notes and references.

Formative Assessment Criteria

The following criteria will be used to assess the assignment:

- Effective justification for the study.
- Clearly defined research question.
- Comprehensive and critical review of the literature (within time constraints).

- Realistic research design.
- Effective consideration of ethical issues.
- Clear plan for writing up and dissemination.
- Fulfilment of professional research ethics requirements.
- Adherence to the relevant guidance for presentation as advised by the Module Tutor.

3.2 Summative Assessments

Assessment consists of an ICA and an ECA, each worth 50% of the overall module mark. The deadlines for these assessments can be found in the assessments section on Blackboard or in the programme assessment timetable.

ICA (50%) - A critical review of a published primary research paper (choice to be made by a trainee from three papers with different methodologies provided by the tutor). One of these papers is from a quantitative research approach, one from a qualitative research approach and the third will use mixed methods (2,000 words). *Learning outcomes: (KU 1-4, CIS 1-3, KTS 1-3)*

ICA Assessment Criteria (Critical Appraisal of Published Primary Research Paper)

The following criteria will be used to assess the assignment:

- Demonstrate a critical understanding of the role of the reviewed paper for clinical psychologists in service delivery and/or practice.
- Demonstrate a critical and comprehensive understanding of the relevant methodological issues.
- Systematically and critically evaluate stages of the research process.
- Demonstrate a comprehensive and critical understanding of the ethical issues involved in the research.
- Reach effectively argued conclusions.
- Demonstrate independent learning ability through reflection on the critical review process.
- Adhere to the American Psychological Association (APA) guidelines for presentation and referencing.

ECA (50%) - A research project proposal which both addresses limitations identified in the ICA critical review (1) and develops the research further with the use of an alternative methodology (2,000 words).

Learning outcomes: (KU 1-4, CIS 1-3, PPS 1-2, KTS 1-5)

ECA Assessment Criteria (Research Project Proposal)

The following criteria will be used to assess the assignment:

- Identify a project that demonstrates a detailed and critical understanding of the research evidence reviewed and wider methodological issues, including the role of the project for informing service delivery and/or practice.

- Provide detailed and appropriately justified solutions to the design of the research project.
- Consider both methodological and ethical issues in the design of the research project.
- Demonstrate a detailed and critical understanding of the data analysis required for the proposed study.
- Adhere to the American Psychological Association (APA) guidelines for presentation and referencing.

3.2.1 Word limits for assessment

Word limits are as stated above (note that there is no allowance of +10%). The word count refers to the assignment itself and does not include the reference list or tables/graphs. The references cited in the main body of text are included in the word count.

3.3 Submitting work

All work should be submitted electronically, using the appropriate links on the module Blackboard site. A printed hard copy of the assignment is **not** required and should not be submitted. **The university's policy is that all assessments must be submitted by 4 p.m. on the day of the deadline.**

3.4 A note about referencing

There is an expectation that all academic assignments conform to current American Psychological Association referencing and citation conventions. Poor referencing will be taken into consideration when marking. It is recommended that you use a digital reference management system (e.g., Refworks, Mendeley), which are freely available (and will save you time). The following online resources are also useful:

<http://reciteworks.com/> - good for checking fine details (e.g., missing references)

<http://www.apastyle.org> - detailed guidance for APA style

<https://owl.english.purdue.edu/owl/resource/560/01/> - additional advice for APA style

Chapter 4

Statistical analysis software

You may be familiar with SPSS from your undergraduate statistics teaching. Please note that we do not use SPSS for teaching and instead use R Statistics. The reason for this is that R is a free statistical package, meaning that it can be accessed in NHS settings that do not have funding for SPSS. This will enable you to run statistical analyses whilst on placement where required, and also enables you to conduct statistical analyses as a qualified Psychologist without incurring any software costs. R is also more flexible than SPSS and has greater functionality. During the teaching you will be shown how to set up and install R, and how to run statistical analyses in this software.

You can obtain R and R Studio from the following links:

<https://cran.r-project.org/>

<https://rstudio.com/>

Chapter 5

Academic Support and Guidance

Please contact the module team if you have any questions, concerns or any other areas you wish to discuss.

5.1 Module Team Contact Details

Module Leader: Dr Christopher Wilson: christopher.wilson@tees.ac.uk

Module Team: Dr Alan Bowman: A.Bowman@tees.ac.uk

Guest lecturers from Schools within the University and Local NHS clinicians also contribute to some teaching

Chapter 6

Appendix 1: Considering your thesis

Your doctoral thesis is one of the largest pieces of work you will undertake during your training and it is important to start thinking about it early on.

You are advised to read around the area of your thesis topic on a continuing basis, and make use of tutorials with your supervision team when needed.

It is also advised that you consider research governance and ethics as you develop your project. Please speak to your academic supervisor about this as they will be able to advise or direct you to someone with appropriate expertise to address queries.

As specified in the Research Handbook, **please note that a revised version of your research proposal forms one of your year two summative assignments.** The deadline for this assignment is **early on in the start of second year** and it is therefore advised that you work on your revised proposal as soon as you receive feedback from the panels and have discussed this with your academic supervisor.

Chapter 7

Introduction to R and R Studio

7.1 By the end of this section, you should be able to:

- Download R and R studio
- Identify the R script, R console, Data environment and file browser in R studio
- Write and run R code from a script
- Install and load R packages

7.2 Why learn / use R?

7.2.1 Some information about R

- R is developed and used by scientists and researchers around the world
- Open source = no cost
- Constant development
- Connects to other data science/research tools
- Worldwide community: training widely available
- Encourages transparency and reproducibility
- Publication-ready outputs

7.2.2 Moving from other software to R

- Workflow is different
 - Organise files and data differently
 - Workspace can contain data and outputs

- Can manage multiple datasets within a workspace
- Learning curve can be steep initially
 - e.g. Variables and coding, scripts
- Need to know what you want
 - e.g. building your regression model / ANOVA error terms

7.3 R has many advantages

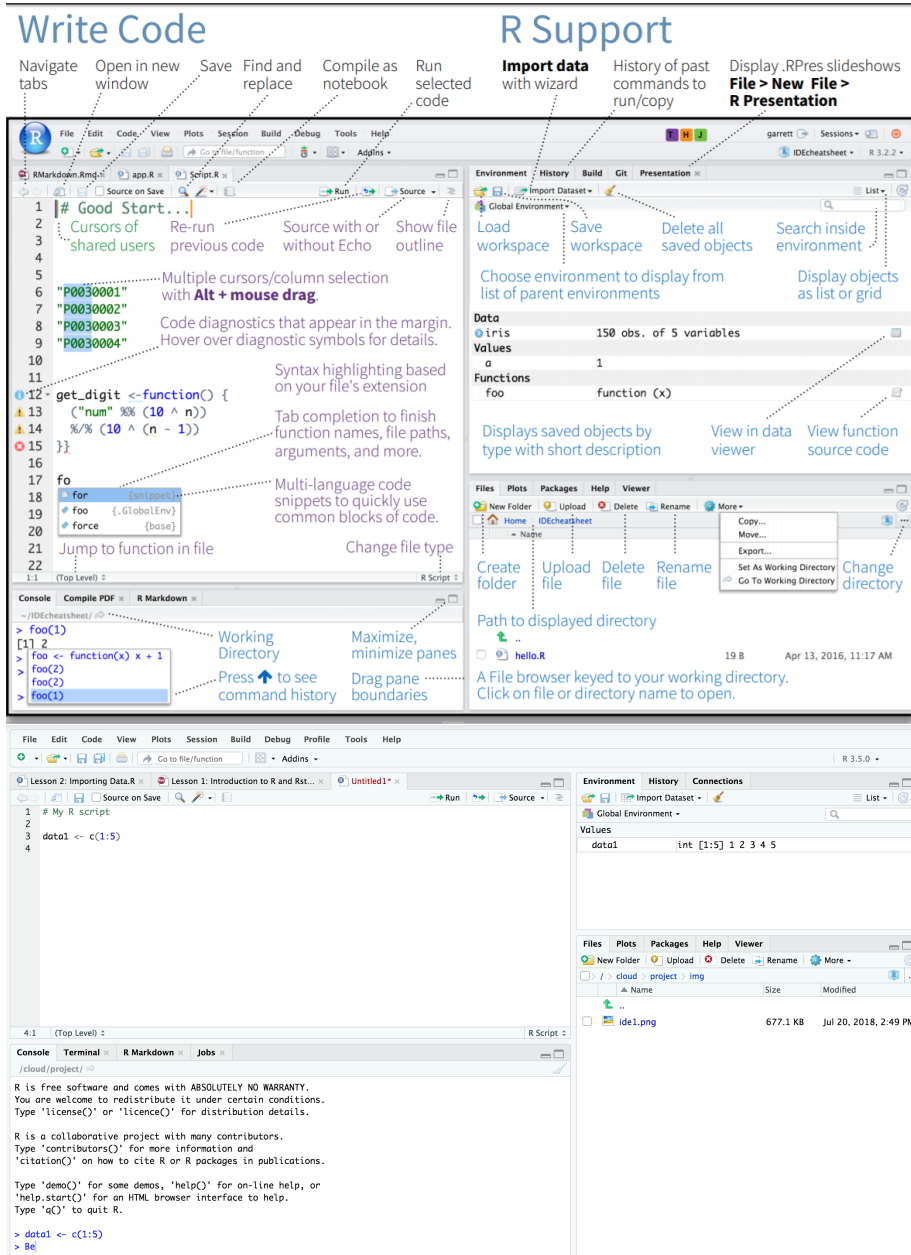
- Using scripts means analysis is easy to follow and reproduce
- R scripts are small, online collaboration, no SPSS “older version” problems
- Data can be organised and reorganised however you need it (tidyr)
- Packages are available for “cutting edge” analysis: e.g. Big Data & Machine Learning
- A robust language for precise plots and graphics (ggplot)
- R analysis code can be embedded into documents and presentations (R Markdown)

7.4 Download R and R Studio

Click on these links to download:

- [R project](#)
- [RStudio](#)

7.5 The R Studio environment



The interface for R Studio looks daunting at first. However, there are 4 main sections, 2 on the left and 2 on the right.

- MAIN TOP: R Script files or R Document Files

- Where we usually type our code as a script before we run it. Script files are usually saved so we can work on them and rerun the code again later (.R files).
- MAIN BOTTOM: Console
 - Shows the output of our R code. We can type R code directly into the console and the answer will output immediately. However, it is more convenient to use script files.
- RIGHT TOP: Environment
 - Contains all of the objects (e.g. data, analysis, equations, plots) that are currently stored in memory. We can save all of this to a file and load it later (.RData files).
- RIGHT BOTTOM: File Browser
 - The folder that R is working from is called ‘the working directory’ and it will automatically look for files there if we try to import something (e.g. a data file). Using the more button on the file browser allows you to set your desired working directory.

7.6 Working with a script

Scripts can be opened from the **File** menu.

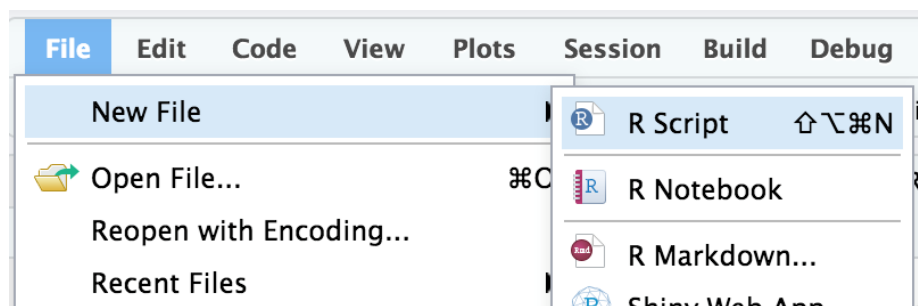


Figure 7.1: Creating a new script

The purpose of scripts is to allow you to type your analysis code and save it for use later. Scripts include, for example:

- Code for importing data into R
- Your analysis code (e.g. t-test or descriptive statistics)
- Code for graphs and tables
- Comments and notes (preceded by the ‘#’ symbol)

To run a script, you click the **Run** button. You can choose to:

- Run the whole script
- Run the selected line of code

When you run the script, you will normally see output in the **console**.

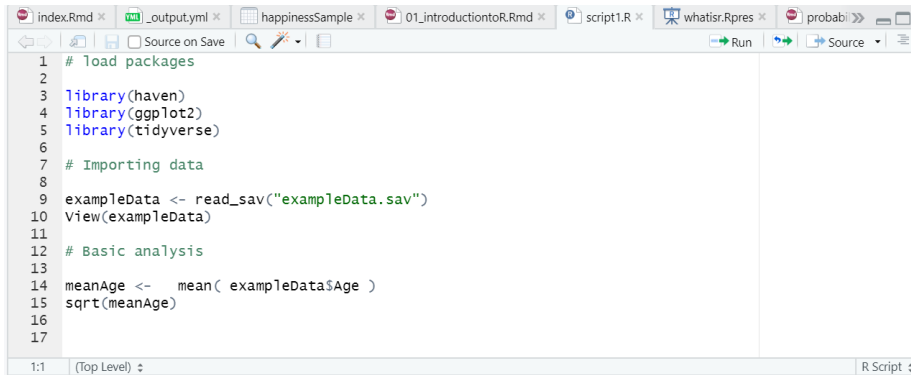


Figure 7.2: Example of an R script

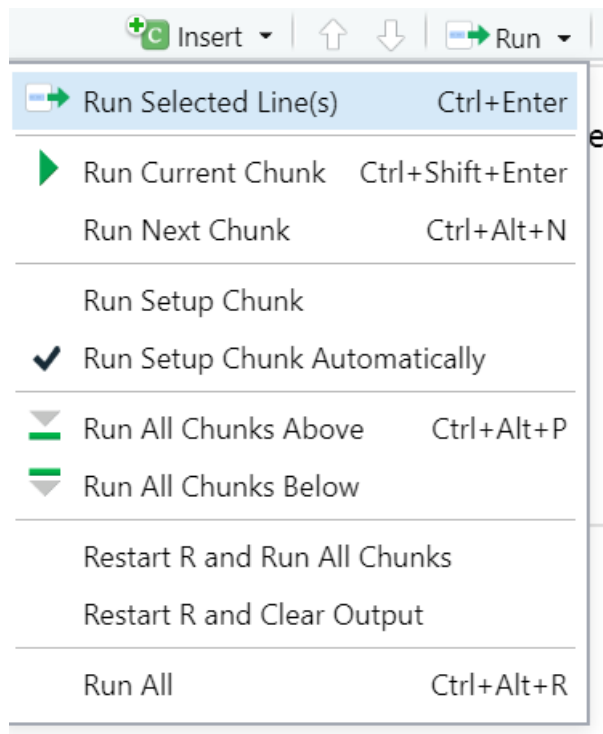


Figure 7.3: The run button

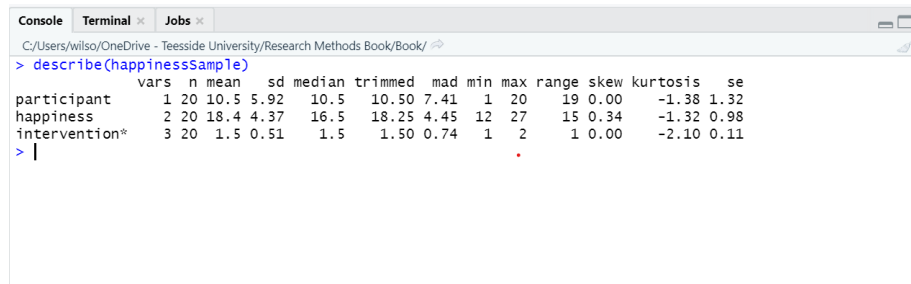


Figure 7.4: Output appears in the console

If your script contains code for a plot (graph), it will appear in the **Plots** window in the bottom right.

7.7 Installing and loading packages

install Packages from RStudio, Inc. on Vimeo.

Packages add functionality to R and allow us to do new types of analysis.

- They can be installed via the menu (Tools -> Install Packages)
- They can also be installed using code:

```
install.packages()
```

For example, TidyR is a package that contains functions for sorting and organising data. To install the package:

or use the code:

```
install.packages("tidyr")
```

Once a package has been installed, you need to load it using the *library()* command. For example:

```
library("tidyr")
```

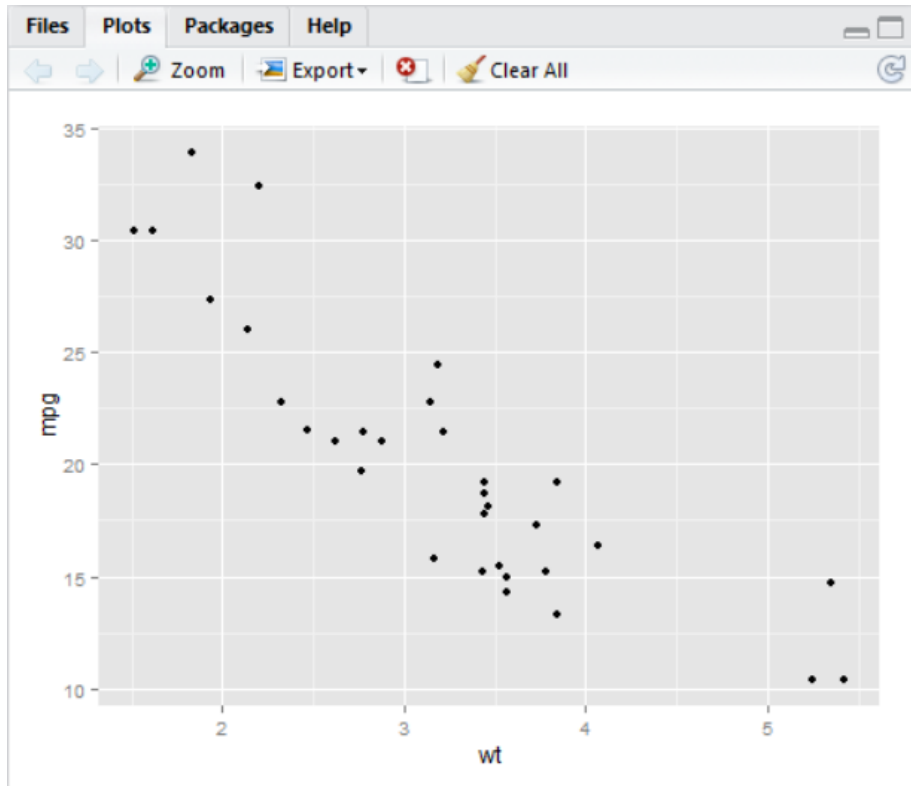



Figure 7.5: Plots appear in the plot window

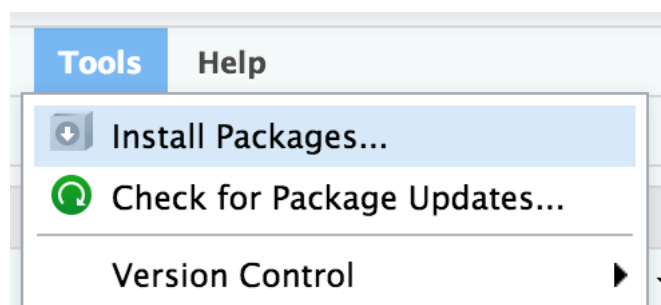


Figure 7.6: Installing a package in RStudio

Chapter 8

Working with data in R

8.1 By the end of this section, you will be able to:

- Import data into R from excel, SPSS and csv files
- Save data to objects
- Identify different data structures and variable types
- Convert variables from one type to another
- Order, filter and group data
- Summarise data
- Create new variables from data

8.2 In this section, we will use the Tidyverse set of packages

- A ‘toolkit’ of packages that are very useful for organising and manipulating data
- We will use the haven package to import SPSS files
- We will use the dplyr to organise data
- Also includes the ggplot2 and tidyR packages which we will use later

To install:

```
install.packages("tidyverse")
```

(See the previous section on installing packages)

8.3 Import data into R from excel, SPSS and csv files

We can import data from a range of sources using the **Import Dataset** button in the **Environment** tab:

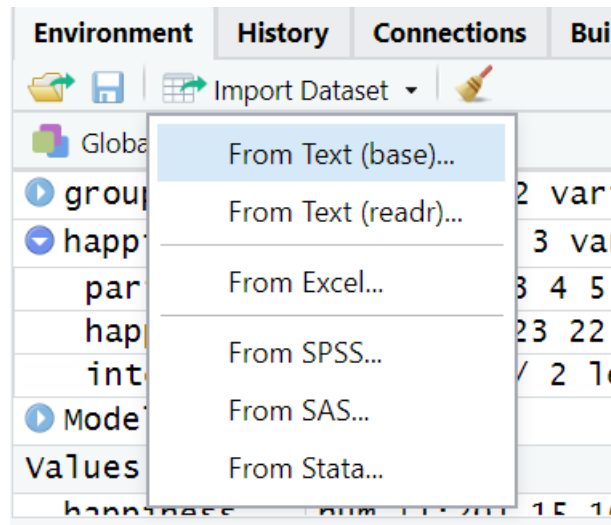


Figure 8.1: Importing data

It is also possible to import data using code, for example:

```
# importing a .csv file
library(readr)
studentData <- read_csv("Datasets/studentData.csv")
```

```
#importing an SPSS file
library(haven)
mySPSSData <- read_sav("Datasets/salesData.sav")
```

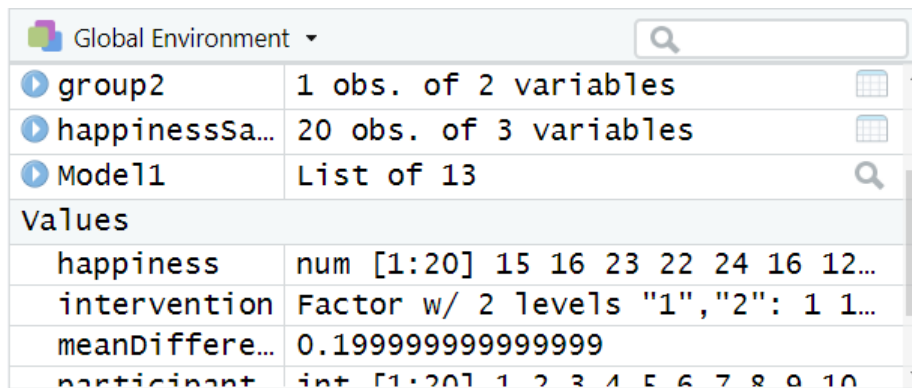
Once the data are imported, it will be visible in the environment:

8.4 Understanding objects in R

In R, an **object** is anything that is saved to memory. For example, we might do some analysis:

```
mean(happiness)
```

However, in the example above, the result would appear in the console but not be saved anywhere. To store the result for reuse later, we save it to an object:



The screenshot shows the R Global Environment window. At the top, there's a search bar and a dropdown menu set to 'Global Environment'. Below this, a list of objects is shown:

Object	Description
group2	1 obs. of 2 variables
happinessSa...	20 obs. of 3 variables
Model1	List of 13

Below the list, there's a section titled 'Values' which shows the details of the 'happinessSa...' object:

Variable	Value
happiness	num [1:20] 15 16 23 22 24 16 12...
intervention	Factor w/ 2 levels "1","2": 1 1...
meanDiffere...	0.199999999999999
participant	int [1:20] 1 2 3 4 5 6 7 8 9 10

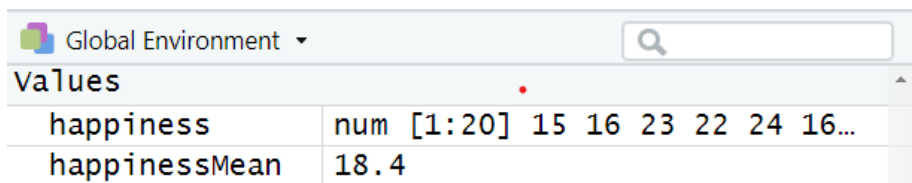
Figure 8.2: Imported data in the environment

```
happinessMean <- mean(happiness)
```

In the above code (reading left to right):

- We name the object “happinessMean”. This name can be anything we want.
- The arrow means that the result of the code on the right will be saved to the object on the left.
- The code on the right of the arrow calculates the mean of *happiness* data

When this code is run, *happinessMean* will be stored in the environment window:



The screenshot shows the R Global Environment window after running the code. The 'Values' section now includes the new object 'happinessMean'.

Variable	Value
happiness	num [1:20] 15 16 23 22 24 16...
happinessMean	18.4

Figure 8.3: Result of a calculation in the environment

To recall an object from the environment, we can simply type its name. For example:

```
happinessMean
```

It's important to note that anything can be stored as an object in R and recalled later. This includes, dataframes, the results of statistical calculations, plots etc.

8.5 Identify different data structures and variable types

8.5.1 Data structures

There are many different types of data that R can work with. The most common type of data for most people tends to be a **dataframe**. A **dataframe** is what you might consider a “normal” 2-dimensional dataset, with rows of data and columns of variables:

	participant	happiness	intervention
1	1	15	1
2	2	16	1
3	3	23	1
4	4	22	1
5	5	24	1
6	6	16	1
7	7	12	1

Figure 8.4: A dataframe example

R can also use other data types.

A vector is a one-dimensional set of values:

```
# a vector example
scores <- c(1,4,6,8,3,4,6,7)
```

A matrix is a multi-dimensional set of values. The below example is a 3-dimensional matrix, there are 2 groups of 2 rows and 3 columns:

```
## , , 1
##
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
##
## , , 2
##
##      [,1] [,2] [,3]
```

8.5. IDENTIFY DIFFERENT DATA STRUCTURES AND VARIABLE TYPES31

```
## [1,]    7    9   11
## [2,]    8   10   12
```

We will primarily work with dataframes (and sometimes vectors), as this is how the data in psychology research is usually structured.

8.5.2 Variable types

With numerical data, there are 4 key data types:

- Nominal (a category, group or factor)
- Ordinal (a ranking)
- Interval (scale data that can include negative values)
- Ratio (scale data that cannot include negative values)

Collecting data – main levels of data

- There are four different levels of numerical data:

Nominal	Ordinal	Interval	Ratio
<ul style="list-style-type: none">• Categories• Can be counted• Cannot be ranked• Cannot be measured• Male/Female. Old/Young, Yes/No	<ul style="list-style-type: none">• Ranks• Can be counted• Can be ranked• Cannot be measured• 1st, 2nd, 3rd	<ul style="list-style-type: none">• Scale with exact values• Can be counted• Can be ranked• Can be measured• Can go below zero• E.g. temperature or difference score	<ul style="list-style-type: none">• Scale with exact values• Can be counted• Can be ranked• Can be measured• <u>Cannot</u> go below zero• E.g. A real number (time, count)

R can use all of these variable types:

- **Nominal** variables are called **factors**
- **Ordinal** variables are called **ordered factors**
- **Interval and ratio** variables are called **numeric** data and can sometimes be called integers (if they are only whole numbers) or doubles (if they all have decimal points)

R can also use other data types such as text (**character**) data.

8.5.3 Convert variables from one type to another

When we first import data into R, it might not recognise the data types correctly. For example, in the below data, we can see the **intervention** variable :

```
##      participant intervention happiness
## 1             1             1  6.785931
## 2             20             1  6.878551
## 3             13             2  7.494531
## 4             14             2  7.549832
## 5             12             2  8.295756
## 6             10             1  8.986315
## 7              2             1  9.139099
## 8             16             2  9.236834
## 9              5             2  9.527681
## 10            9             2  9.592461
```

In the **intervention** variable, the numbers 1 and 2 refer to different intervention groups. Therefore, the variable is a **factor** variable. To ensure that R understands this, we can resave the intervention variable as a factor using the `as.factor()` function:

```
happinessSample$intervention <- as.factor(happinessSample$intervention)
```

8.6 Working with dataframes

Dataframes are the more standard data format that were are used to (think of how a dataset looks in SPSS or Excel).

In a dataframe, variables are columns and each row usually represents one measurement or one participant.

8.6.1 View dataframe

To view a dataframe, we can click on it in the environment window and it will display:

8.6.2 Refer to variables (columns) in a dataframe

Columns in a dataframe are accessed using the “\$” sign. For example, to access the *happiness* column in the *happinessSample* dataframe, we would type:

```
happinessSample$happiness
```

```
## [1]  6.785931  9.139099 11.319182 11.873362  9.527681  9.950592  9.841057 10.58425
## [10]  8.986315 12.965060  8.295756  7.494531  7.549832 11.049785  9.236834 11.61314
## [19] 12.127610  6.878551
```

As we can see above, the result is then displayed.

Global Environment		
group2	1 obs. of 2 variables	
happinessSa...	20 obs. of 3 variables	
Model1	List of 13	
Values		
happiness	num [1:20] 15 16 23 22 24 16 12...	
intervention	Factor w/ 2 levels "1","2": 1 1...	
meanDiffere...	0.1999999999999999	
participant	int [1:20] 1 2 3 4 5 6 7 8 9 10	

Figure 8.5: Clicking on datasets in hte environment will open them up for viewing

	participant	happiness	intervention
1	1	15	1
2	2	16	1
3	3	23	1
4	4	22	1
5	5	24	1
6	6	16	1
7	7	12	1

Figure 8.6: Viewing a dataframe

8.6.3 Order, filter and group data

If you have the **tidyverse** package loaded, it is easy to organise and filter data.

```
arrange(happinessSample, happiness)
```

```
##      participant intervention happiness
## 1             1             1  6.785931
## 2            20             1  6.878551
## 3            13             2  7.494531
## 4            14             2  7.549832
## 5            12             2  8.295756
## 6            10             1  8.986315
## 7             2             1  9.139099
## 8            16             2  9.236834
## 9             5             2  9.527681
## 10           9             2  9.592461
## 11           18             2  9.635646
## 12            7             2  9.841057
## 13            6             1  9.950592
## 14            8             1 10.584258
## 15           15             1 11.049785
## 16            3             2 11.319182
## 17           17             2 11.613146
## 18            4             1 11.873362
## 19           19             2 12.127610
## 20           11             2 12.965060
```

```
arrange(happinessSample, desc(happiness)) # Arrange in descending order
```

```
##      participant intervention happiness
## 1             11             2 12.965060
## 2            19             2 12.127610
## 3             4             1 11.873362
## 4            17             2 11.613146
## 5             3             2 11.319182
## 6            15             1 11.049785
## 7             8             1 10.584258
## 8             6             1  9.950592
## 9             7             2  9.841057
## 10           18             2  9.635646
## 11            9             2  9.592461
## 12            5             2  9.527681
## 13           16             2  9.236834
## 14            2             1  9.139099
## 15           10             1  8.986315
## 16           12             2  8.295756
```

```
## 17      14      2  7.549832
## 18      13      2  7.494531
## 19      20      1  6.878551
## 20       1      1  6.785931
```

- Show clients with a happiness score of less than 4

```
filter(happinessSample, happiness < 4)
```

```
## [1] participant intervention happiness
## <0 rows> (or 0-length row.names)
```

- Show Intervention group 2 with happiness scores above 7

```
filter(happinessSample, happiness > 7 & intervention == 2)
```

```
##      participant intervention happiness
## 1           3           2 11.319182
## 2           5           2  9.527681
## 3           7           2  9.841057
## 4           9           2  9.592461
## 5          11           2 12.965060
## 6          12           2  8.295756
## 7          13           2  7.494531
## 8          14           2  7.549832
## 9          16           2  9.236834
## 10         17           2 11.613146
## 11         18           2  9.635646
## 12         19           2 12.127610
```

- Group by intervention and show the mean happiness score

```
happinessSample %>% group_by(intervention) %>% summarise(mean = mean(happiness))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   intervention mean
##   <fct>         <dbl>
## 1 1           9.41
## 2 2           9.93
```

8.6.4 Create new variables from data

To create new variables from data, we can use the **mutate()** function.

For example, let's say we wanted to calculate the difference between each person's happiness score and the mean happiness score.

We could do the following:

```
happinessSample %>% mutate(difference = happiness - mean(happiness))
```

##	participant	intervention	happiness	difference
## 1	1	1	6.785931	-2.93640295
## 2	2	1	9.139099	-0.58323556
## 3	3	2	11.319182	1.59684776
## 4	4	1	11.873362	2.15102748
## 5	5	2	9.527681	-0.19465333
## 6	6	1	9.950592	0.22825761
## 7	7	2	9.841057	0.11872229
## 8	8	1	10.584258	0.86192386
## 9	9	2	9.592461	-0.12987337
## 10	10	1	8.986315	-0.73601959
## 11	11	2	12.965060	3.24272524
## 12	12	2	8.295756	-1.42657888
## 13	13	2	7.494531	-2.22780305
## 14	14	2	7.549832	-2.17250279
## 15	15	1	11.049785	1.32745030
## 16	16	2	9.236834	-0.48550020
## 17	17	2	11.613146	1.89081116
## 18	18	2	9.635646	-0.08668862
## 19	19	2	12.127610	2.40527587
## 20	20	1	6.878551	-2.84378324