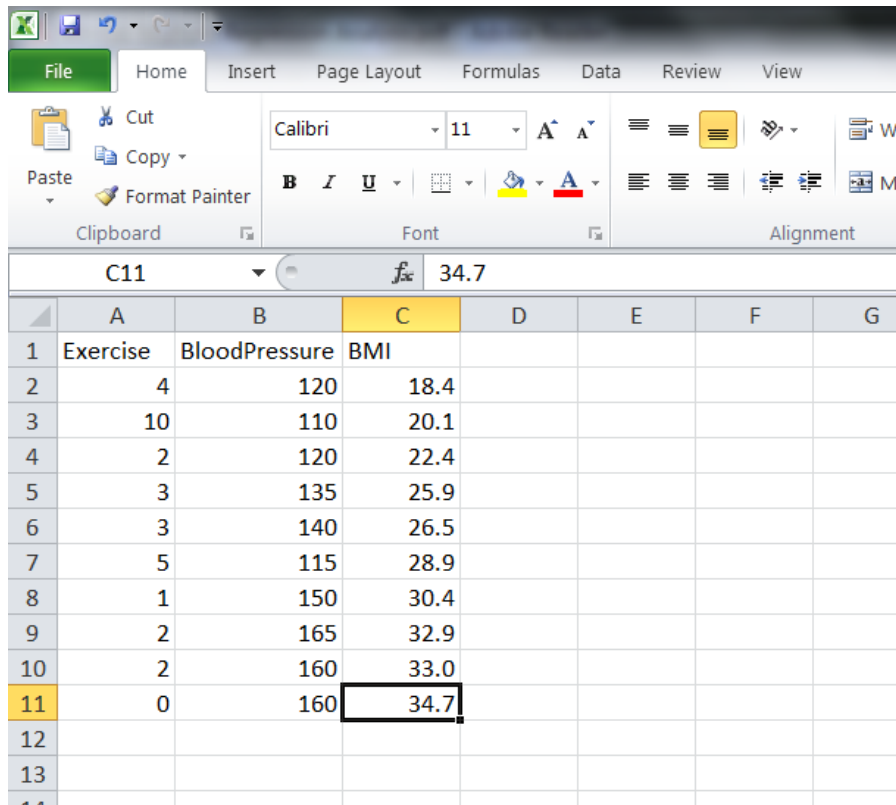


R is a free statistical software environment that can run a wide variety of tests based on different downloadable packages and can produce a wide variety of simple to more complex graphs based on the data provided. R can seem daunting to use at first because it does not possess a user interface as friendly as many other software packages, but it more than makes up for it with its structured, easy-to-use command structure and flexible environment.

The first step with using a statistical program to solve a program is to enter the data for use in the program, and R is no different. There are many different ways to enter and revise data in R, but this tutorial will cover what CASE considers to be one of the easiest methods. This step-by-step process involves using Excel (or any other spreadsheet based program) and R.

Importing Data into R Using Excel

Step 1: Enter data into an Excel spreadsheet. Label each column with the name of the variable. Any character can be used, but do not use spaces in the name (ex. Blood Pressure should be BloodPressure or Blood_Pressure). If spaces are used an error message will appear when importing the data to R. (If you need additional support for entering data in spreadsheet formats, please see the CASE “Introduction to Minitab” tutorial listed under [Minitab Tutorials](#) for step-by-step directions.)



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G
1	Exercise	BloodPressure	BMI				
2	4	120	18.4				
3	10	110	20.1				
4	2	120	22.4				
5	3	135	25.9				
6	3	140	26.5				
7	5	115	28.9				
8	1	150	30.4				
9	2	165	32.9				
10	2	160	33.0				
11	0	160	34.7				
12							
13							
14							

Step 2: Save the file as an Excel file for future editing, adding more data, changing data, to preserve any formulas you were using, etc.

Step 3: Click/select **Save As**.

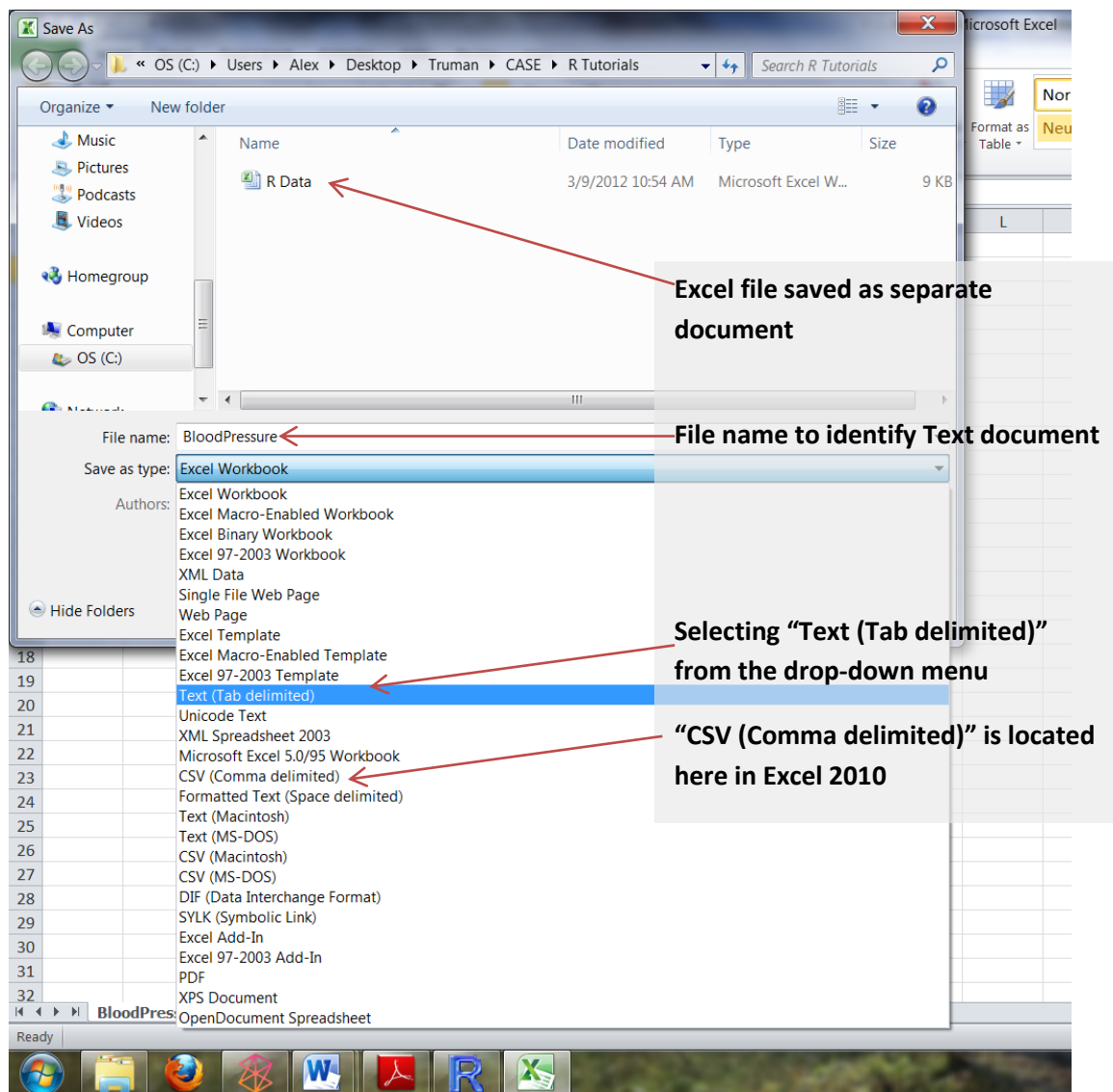
For the **Save as type:** select the **Text (Tab delimited)** or **CSV (Comma delimited)** option from the drop-down menu depending on preference and capabilities of the computer. Name the file something that can be identified for future use.

Before clicking **Save**, note the folder the files are being saved into. Then click **Save**.

A notification window will appear. Click **OK**.

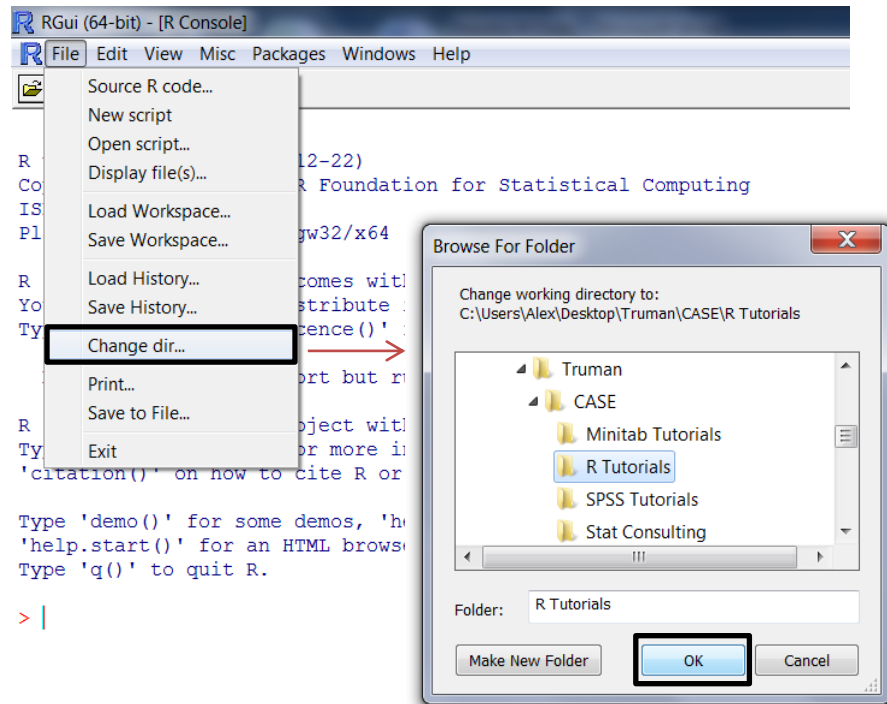
Another notification window will appear with the alert that saving in this format may alter you data, click **Yes**.

The Excel file is now saved as a .txt or .csv file that can be imported to R. If more changes need to be made to this file just repeat this process and save over the file or create a new .txt or .csv file.



Step 4: Open R. The Windows version of R will open to the RGui program by default that allows access to the command-line tools used by R. For Linux or Mac be sure to open the correct GUI interface for your system.

Step 5: To find the file just created select **File**, then select **Change dir....** Find the folder the file is saved to on the computer and click **OK**.



Step 6: Enter the following code into R at the > symbol to import the data in R:

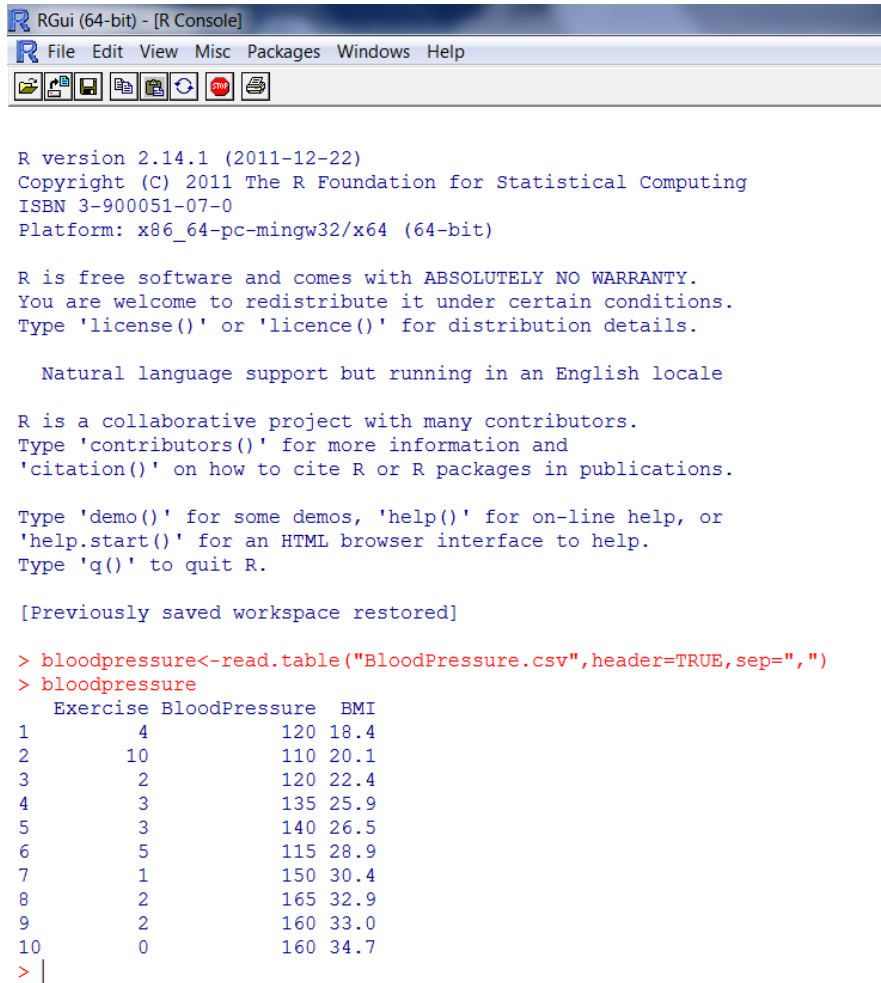
For .txt file: `bloodpressure<-read.table("BloodPressure.txt",header=TRUE)`

For .csv file: `bloodpressure<-read.table("BloodPressure.csv",header=TRUE,sep=",")`

Press **Enter** on the keyboard. If the file is imported correctly it should just appear with a new > symbol on the line below the code waiting for further instructions. The word **bloodpressure** above can be replaced with nearly any word or collection of letters, such as BP, Blood_pressure, or abc123, to identify the dataset being used. **bloodpressure** serves as an object to store the information so the computer can easily understand what the user wants. From this point on R recognizes that "**bloodpressure**" holds the table of data contained in the .txt or .csv file created from the Excel document.

Step 7: To check the data to ensure it uploaded all cases, type *bloodpressure* at the > and the dataset should appear on the screen as it was entered into Excel.

The following screenshot shows the data entry code from step 6 (for a .csv file) and checking the data in step 7:



```
R version 2.14.1 (2011-12-22)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> bloodpressure<-read.table("BloodPressure.csv",header=TRUE,sep=",")
> bloodpressure
  Exercise BloodPressure  BMI
1         4           120 18.4
2        10           110 20.1
3         2           120 22.4
4         3           135 25.9
5         3           140 26.5
6         5           115 28.9
7         1           150 30.4
8         2           165 32.9
9         2           160 33.0
10        0           160 34.7
> |
```

At this point the data has been successfully uploaded and can now be used to run statistical tests, produce graphics, or complete any other task that code can be found for in R.

Editing Minor Data Misentries/Mistakes in R

R can edit the data set uploaded in the previous steps in a spreadsheet environment. This is useful for correcting small errors or adding a few cases. Any changes should also be made to the master Excel document created to originally store the data so that it remains consistent. If major changes need to be made, it may be worth going back into Excel and repeating the steps listed above to just import the updated data set.

If the data set changes are minor R has a useful tool for easily updating the entries. Click **Edit** and then select **Data editor...** A window will appear that asks for “Name of data frame or matrix”. In the space provided enter the name of the object entered above, in the case of this example, **bloodpressure**. Click **OK**.

The following screen appears:

	Exercise	BloodPressure	BMI	var4	var5	var6	var7	var8
1	4	120	18.4					
2	10	110	20.1					
3	2	120	22.4					
4	3	135	25.9					
5	3	140	26.5					
6	5	115	28.9					
7	1	150	30.4					
8	2	165	32.9					
9	2	160	33					
10	0	160	34.7					

Like in Excel, the numbers or column names can be changed by simply clicking on the cell and typing in the updated entry. Any changes made here will permanently affect the data as it is used in R, but will not change the .txt or .csv file used to import the data. Therefore, any changes made in R also need to be made to the master Excel document, otherwise the same data corrections will need to be made every time a new .txt or .csv file is created from the Excel document and imported into R.

Once all changes have been made the window must be closed in order to continue using R.

Exporting Data in R Back to .txt or .csv File

If changes are made to your data, it is possible to update the file from inside R instead of having to open the file separately. To save your data to a .txt or .csv file from R use the following code:

For .txt file: `write.table(bloodpressure,file="BloodPressure.txt",sep=" ")`

For .csv file: `write.csv(bloodpressure,file="BloodPressure.csv")`

where **bloodpressure** represents the name of the dataset you have in R. The file name should be something you can easily recognize with the dataset. If you are updating an already existing file, it should be the exact file name. .txt files require the `sep=" "` from the code to identify that spaces are used as the separator of the different parts of data in the exported file.

Closing R

To save all data imported or entered into R as well as any other variables that have been created from other commands click **File** then **Save Workspace...** and name the file. To return to this data set and workspace simply open the file in the folder it was saved in. R should save all workspace information, so opening the file in R after closing the program should result in the same objects (such as **bloodpressure** representing the imported dataset) being the same.

If you want to use the GUI to enter a command to save the image just use the following code:

`save.image(file="BloodPressureFile.RData")`

where **BloodPressureFile** can be whatever name you wish to have in your folder for the R data.