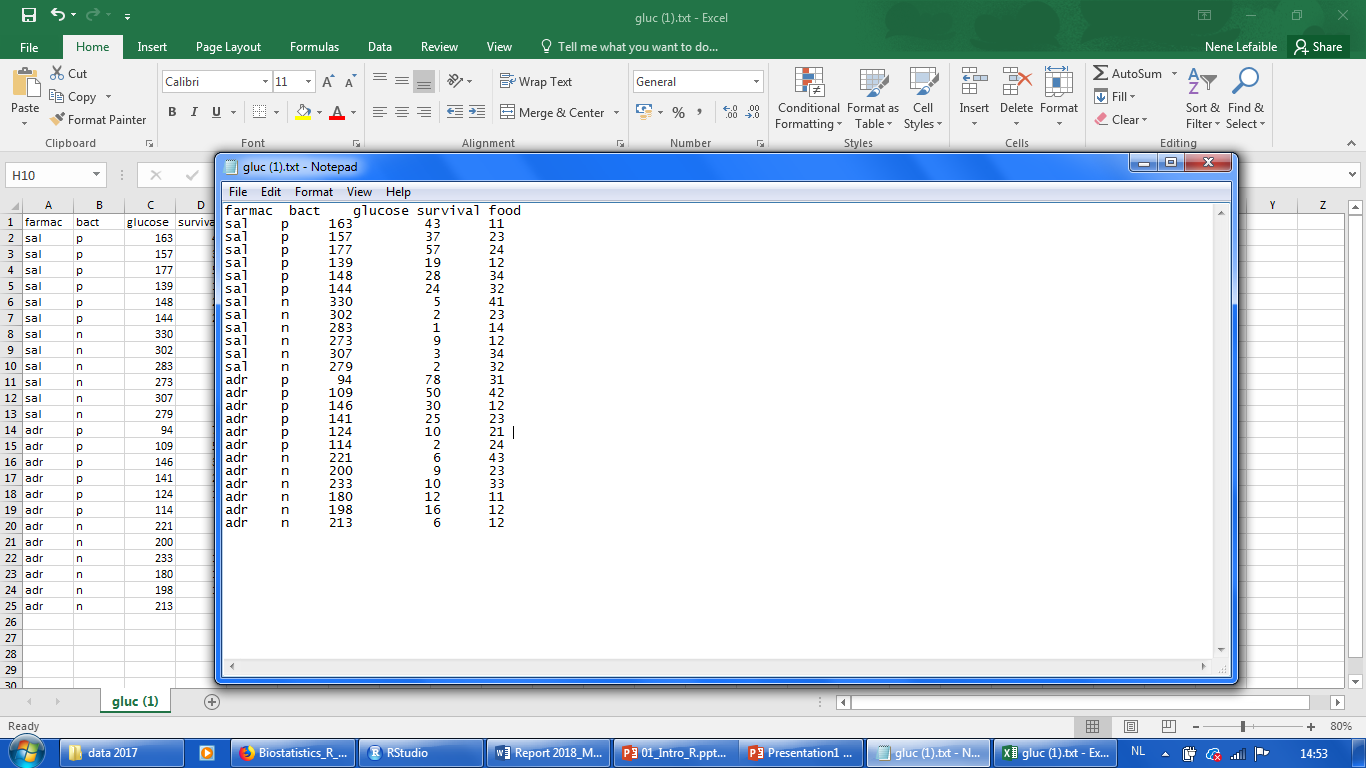
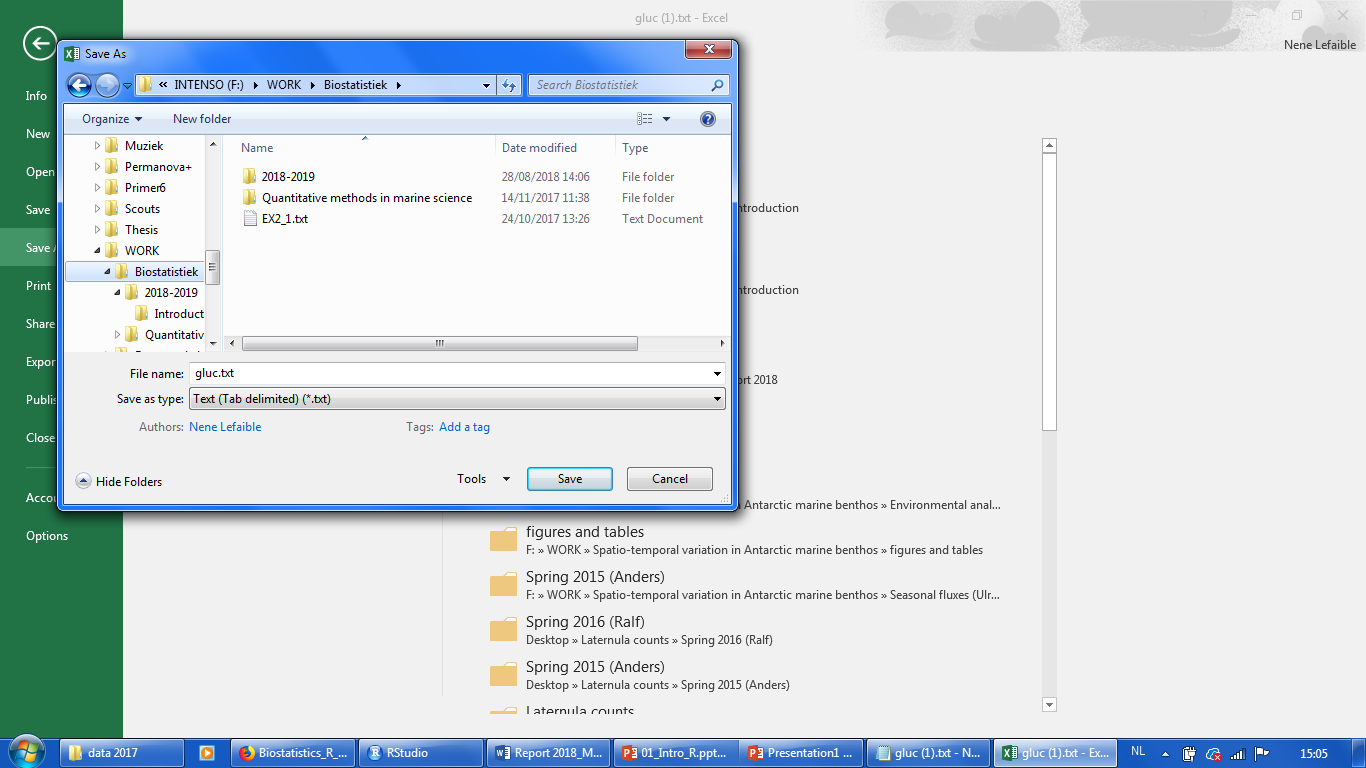
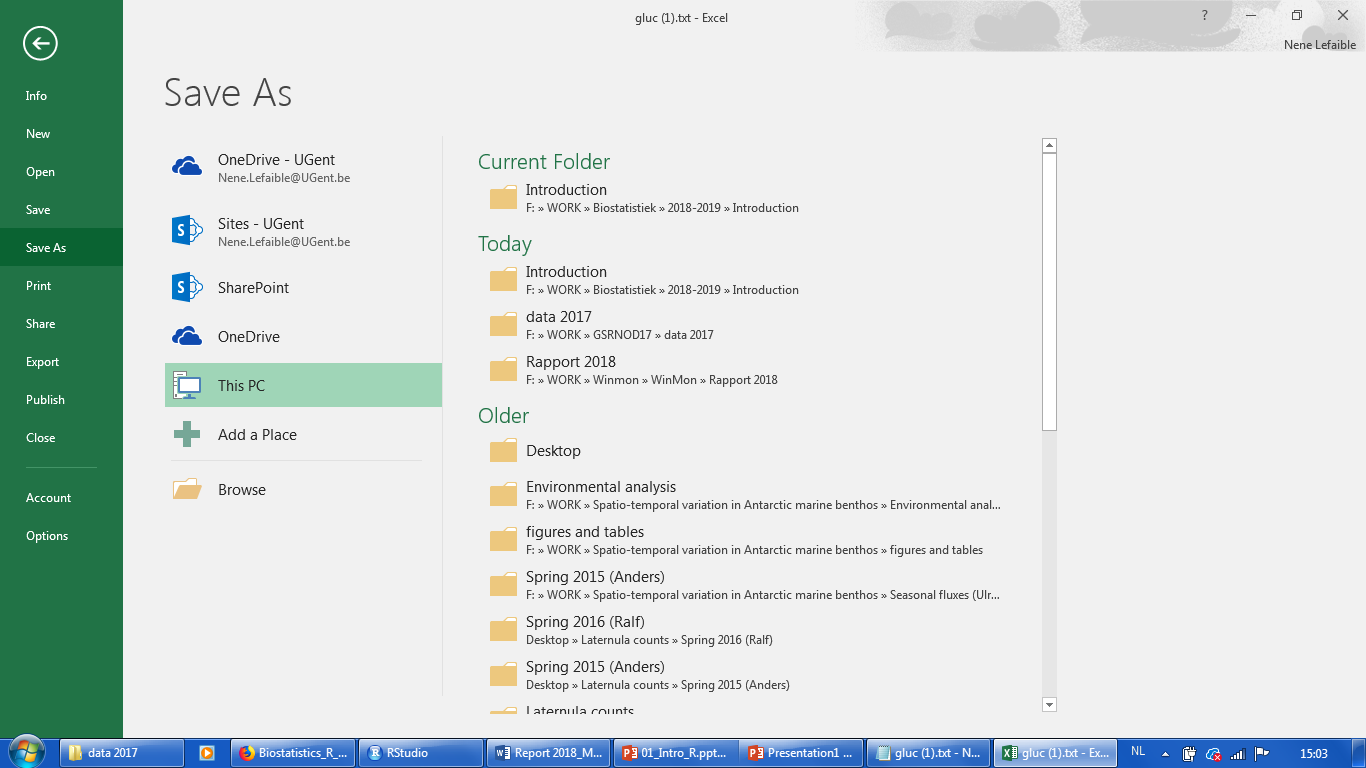
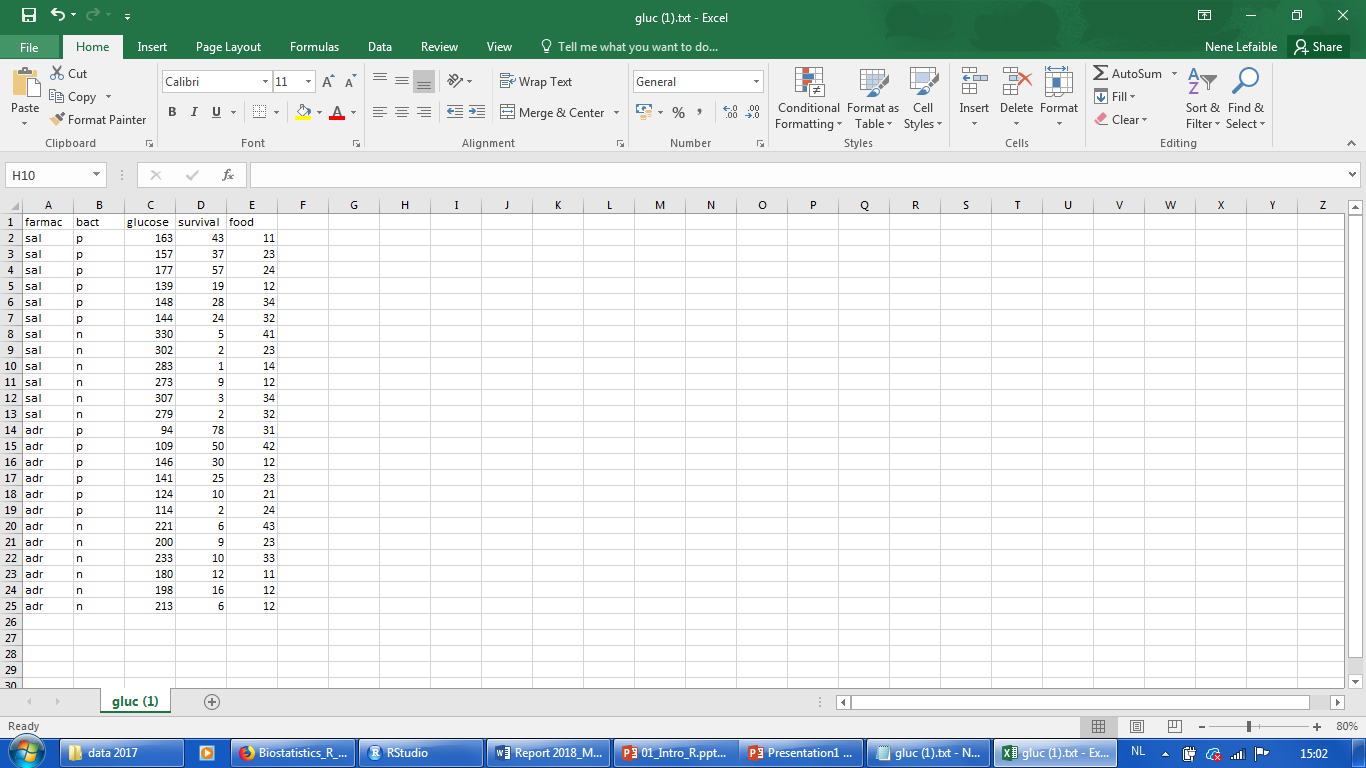
**R Introduction to data frames and packages**

**Working with data frames**

Key Points:

* R can only open datasets as “.txt” or “.csv” files
* R only recognizes “.” As a decimal sign
* The first row usually contains the column names/ names of the variables that you are investigating and testing
* Your column names cannot contain white spaces eg. “glucose\_level” instead of “glucose level”

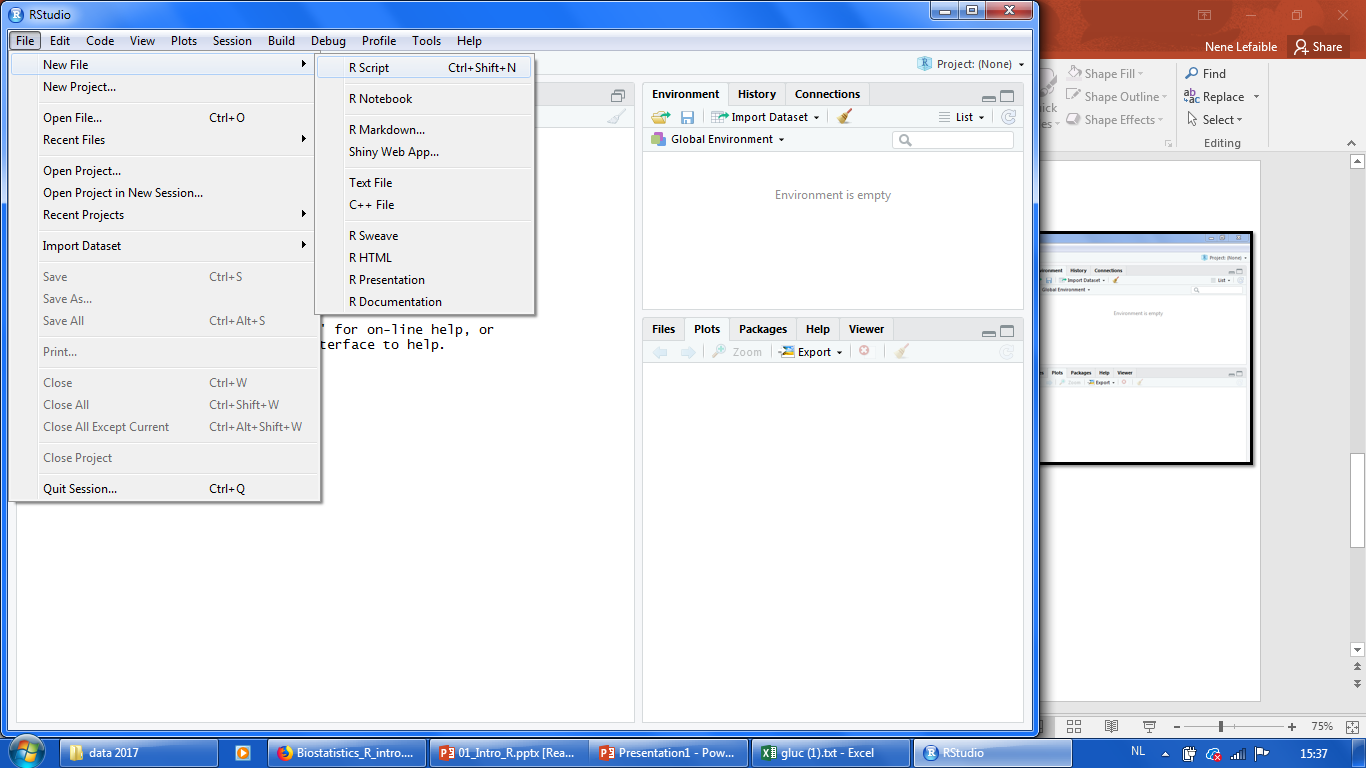
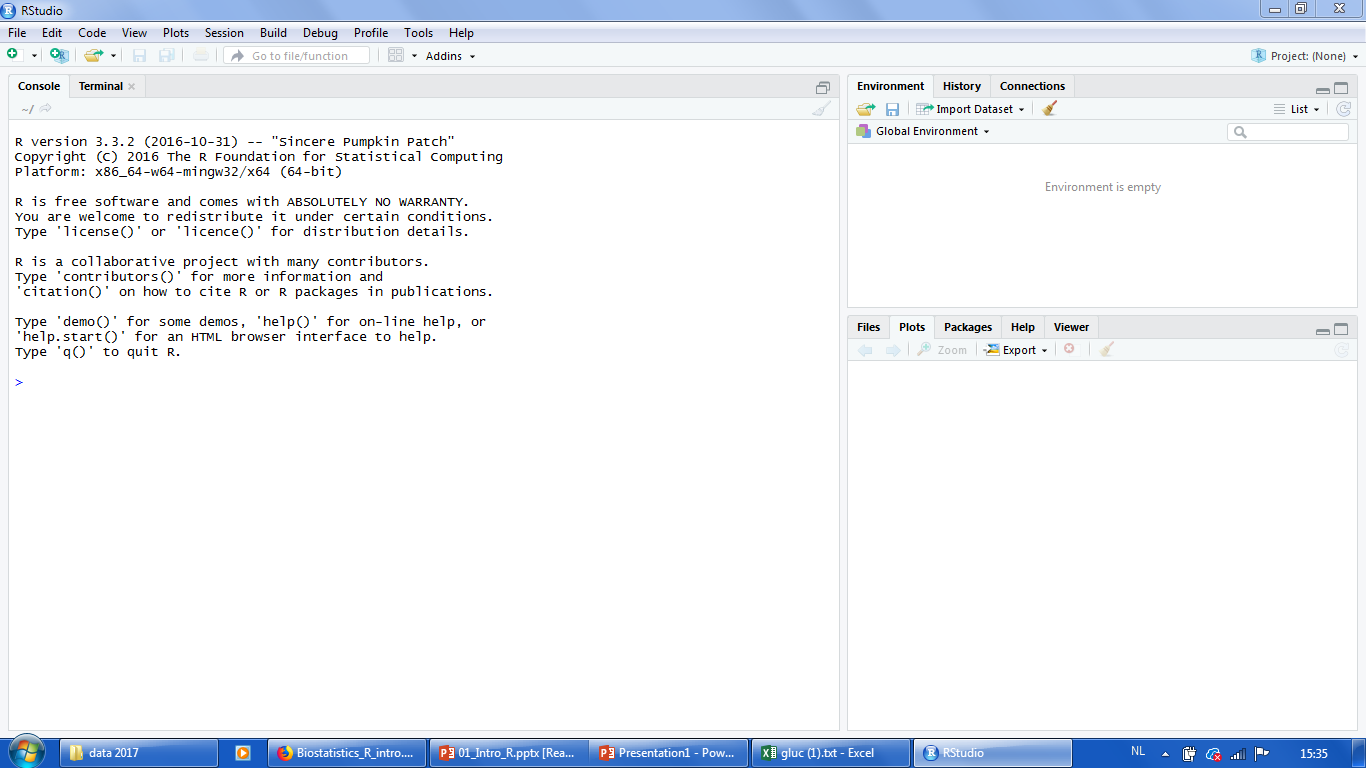
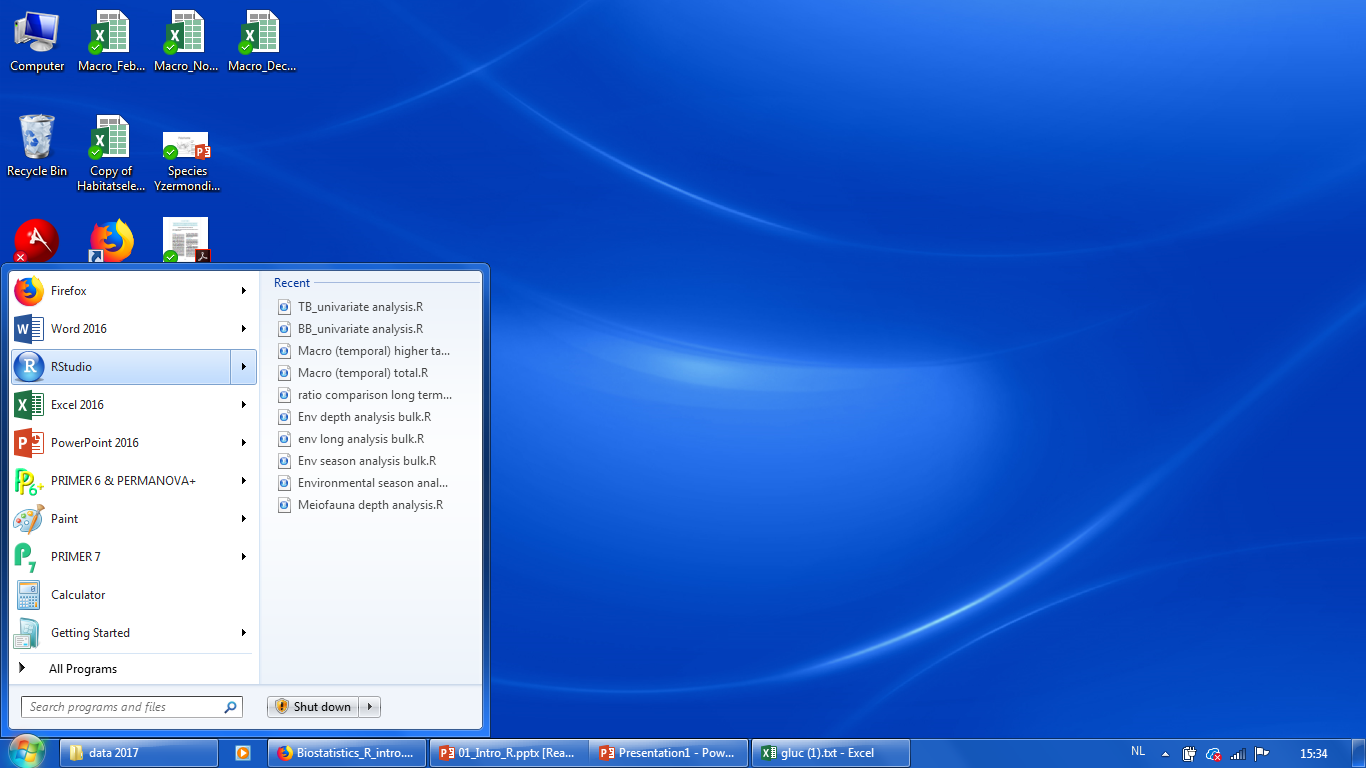
1. If you are working in excel, you can save your data into a text file as shown here:



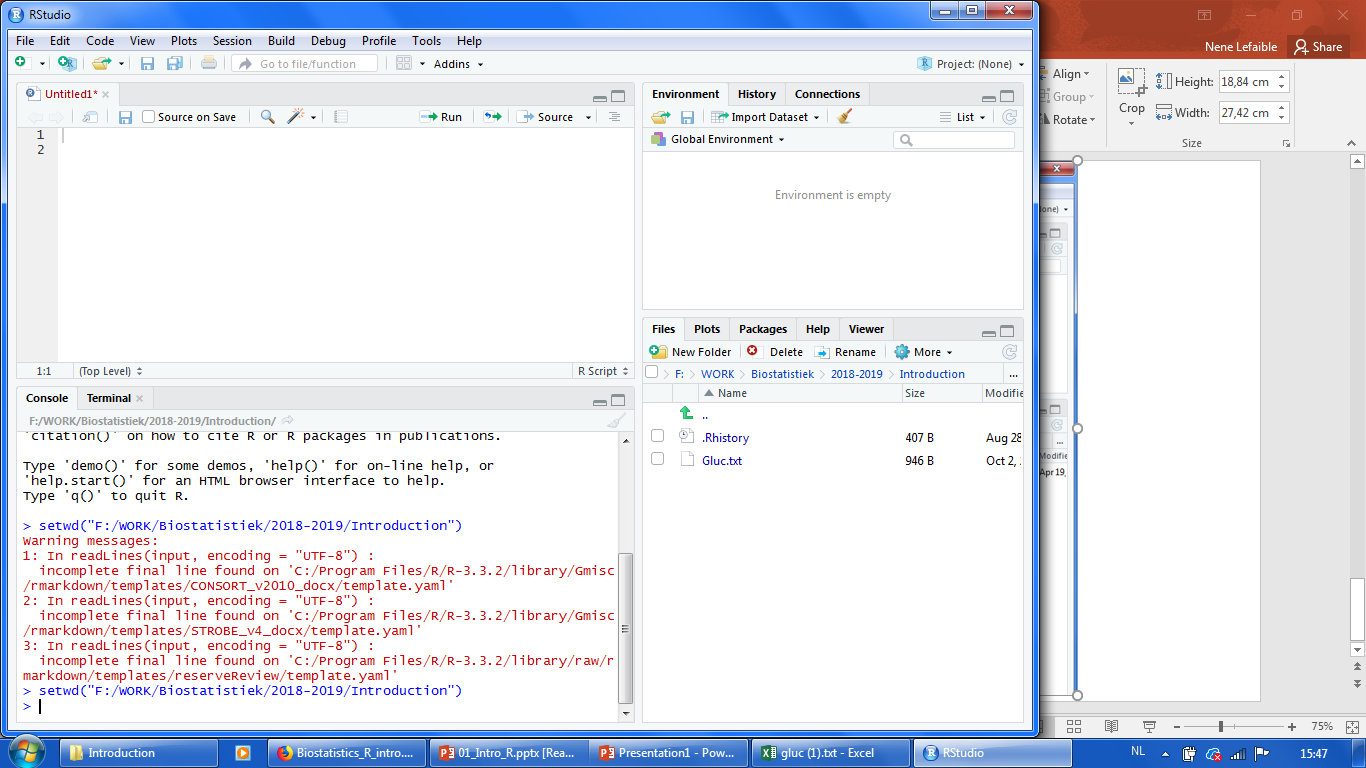
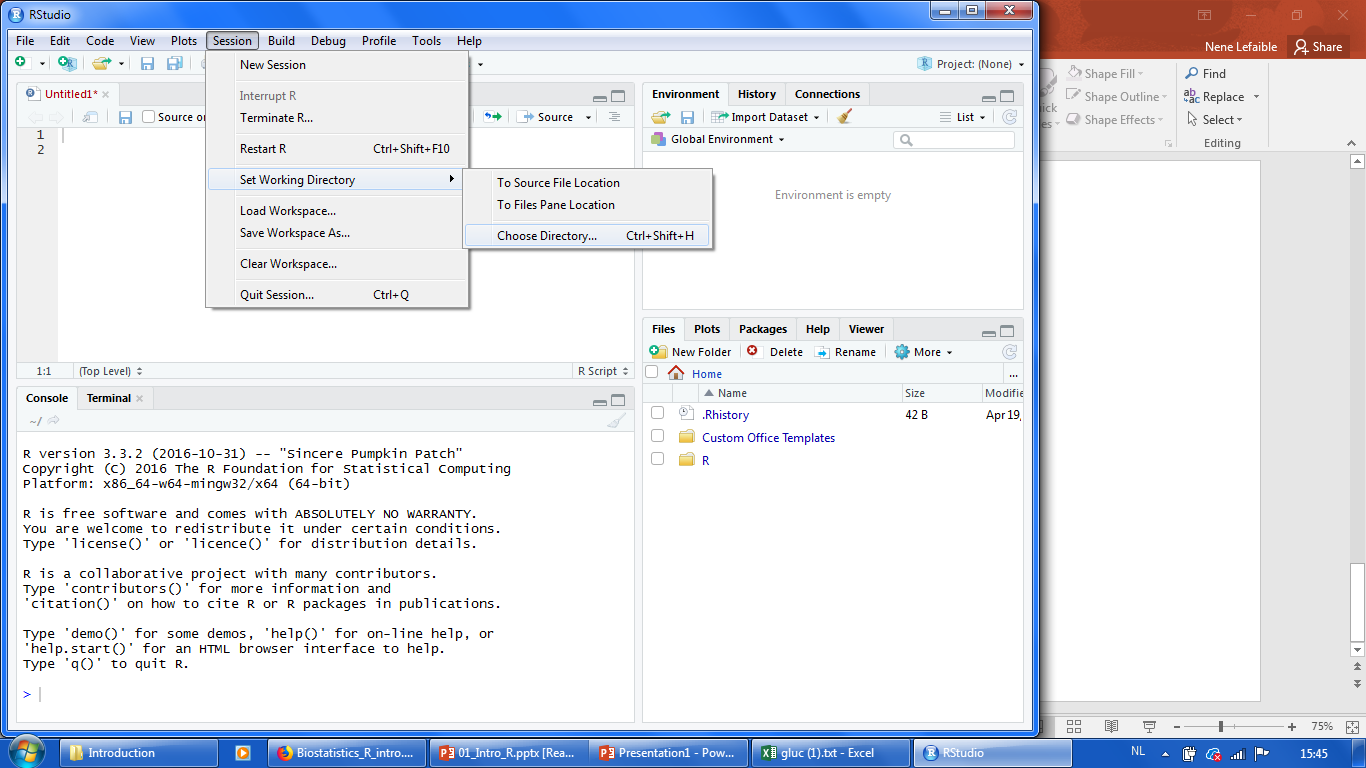
1. To read a data frame, first create a new R-script within RStudio as shown here:

R searches data in its working directory (place on computer where text file is stored)

You can specify it in several ways by clicking on “Session – Working directory”

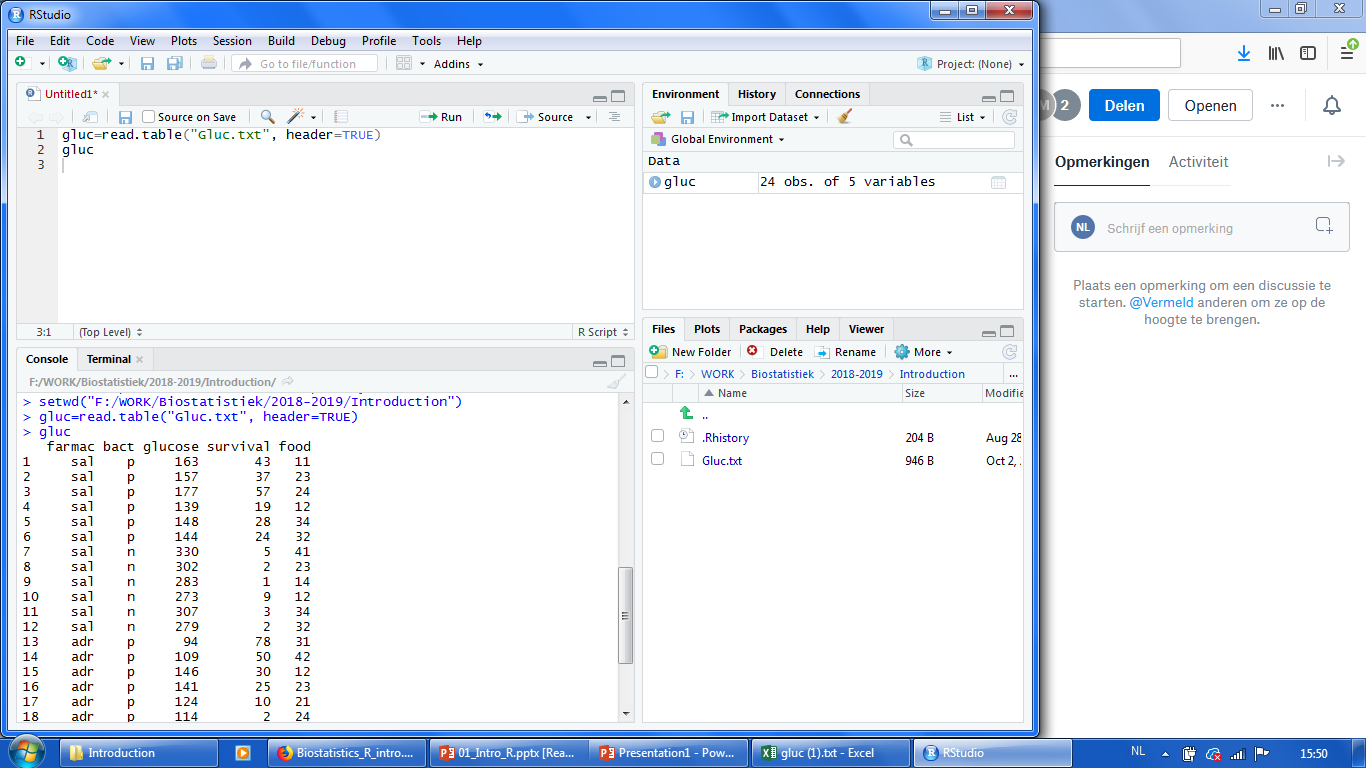


1. You can tell R where to find the data on your computer by setting the correct “working directory” as shown here and then browsing for the directory where you saved your files on your computer.



After choosing the directory, you will see your dataset on the right side within the “Files” folder.

1. Write the command line in the script to “read” your selected dataset as shown here. Make sure that your selected dataset is written exactly the same as the file you stored in the working directory including capitalization. So, if the file is stored as “Gluc.txt”, “gluc.txt” will give you an error message.



This is your “working environment” which now contains your dataset.

This expression confirms that the first row in the dataset contains column names

If you run the command, R will read in the dataset as shown here & will store it in your “working environment”

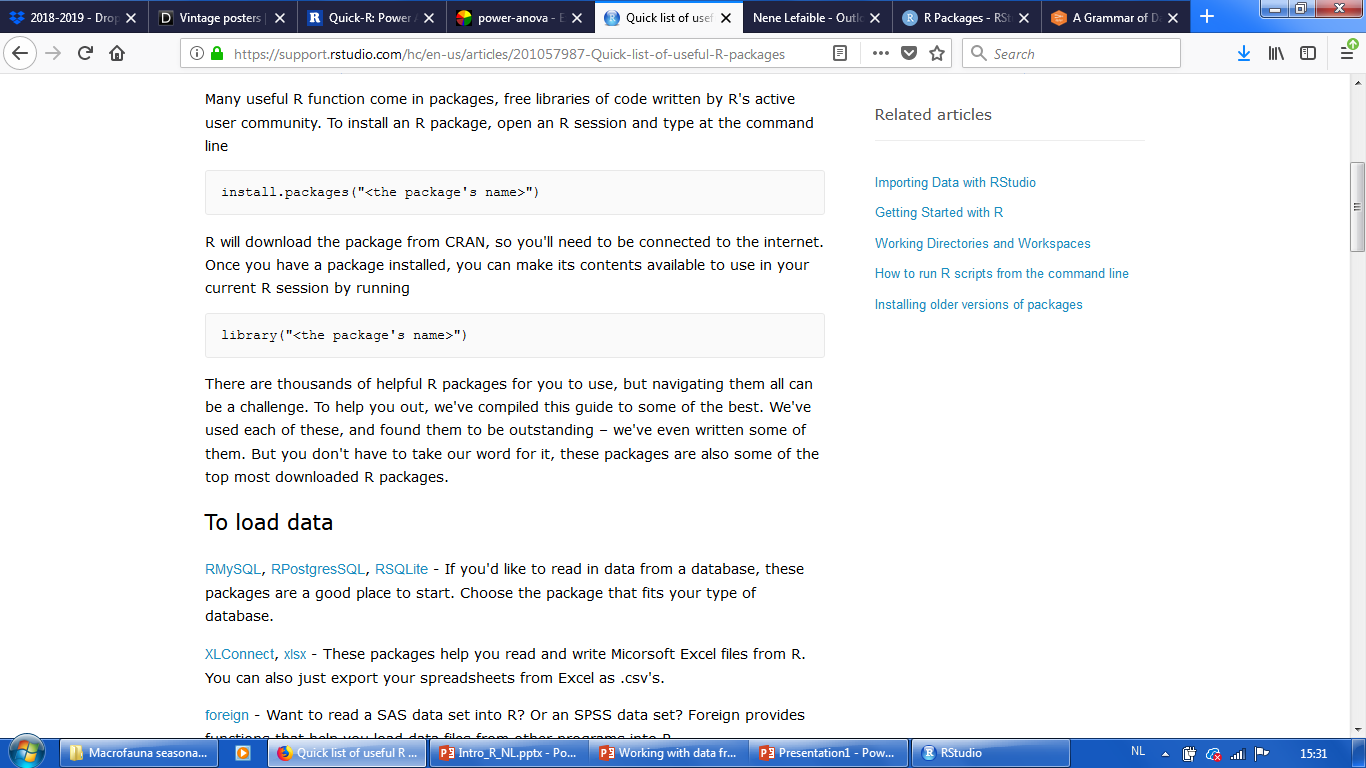
Your selected dataset.

1. Once you have read in your file, you can start writing commands (“code”) to perform several actions on your data.

Here are some examples of things you might want to do with your data

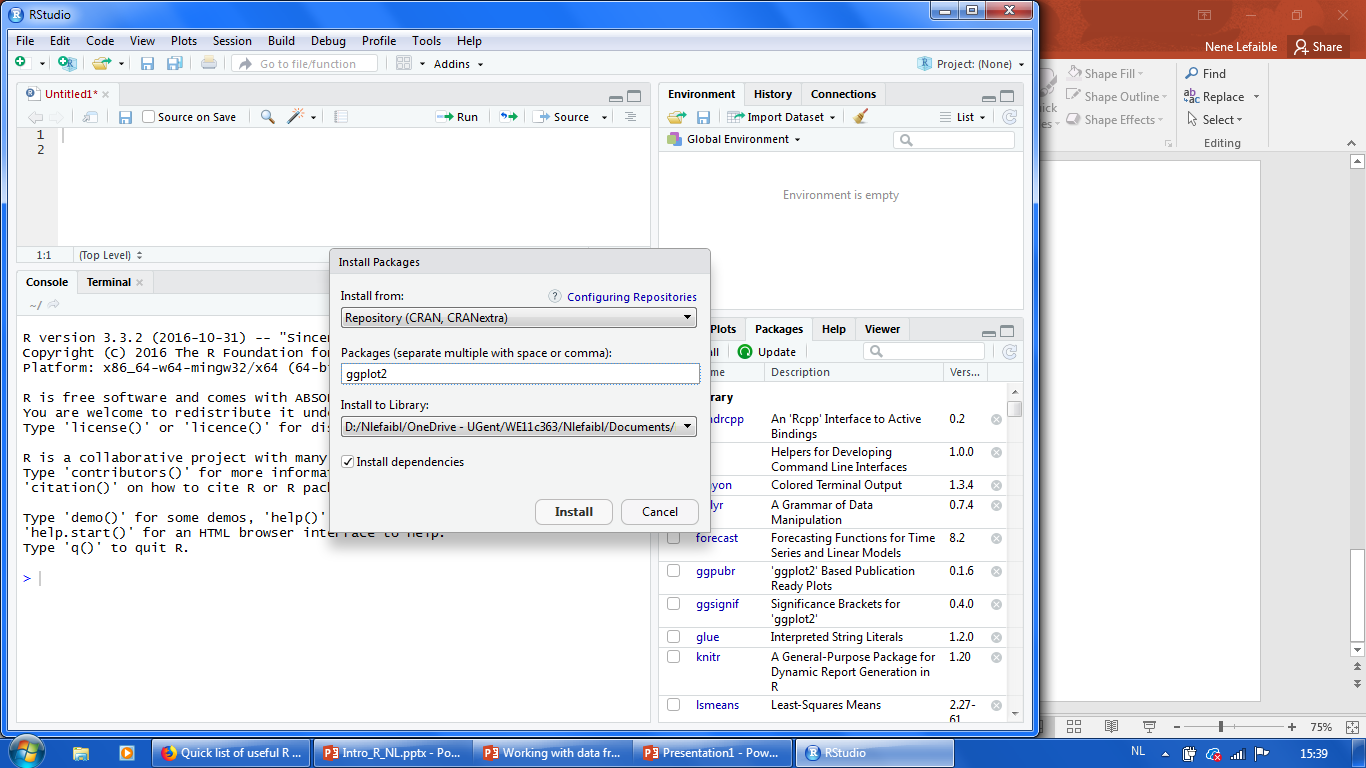
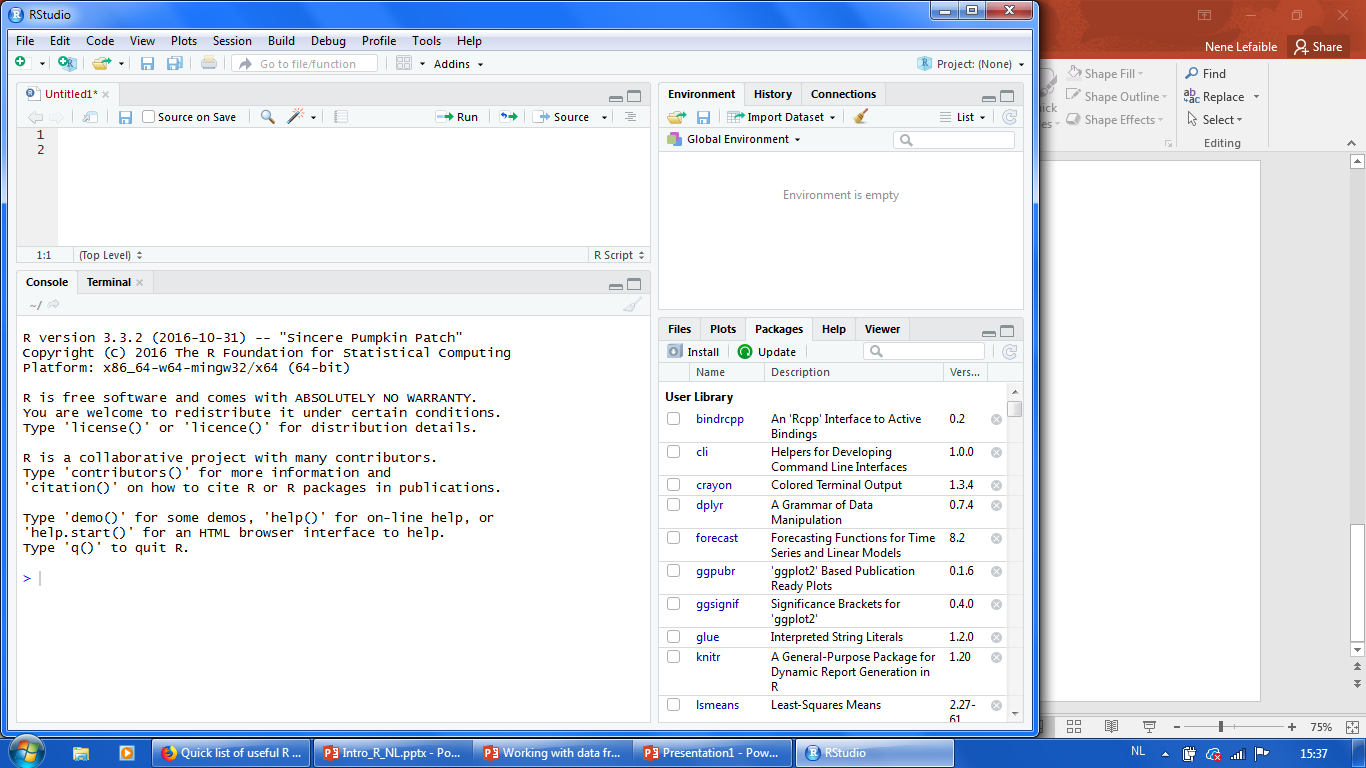
* 1. You can select specific parts of your dataset = subsets
  2. You can perform calculations such as finding the mean, the standard deviation or the standard error
  3. You can perform statistical analyses such as t-tests, ANOVA or regression analyses
  4. You can create a variety of graphs such as X-Y plots, barplots, boxplots or many others…

**Packages**



You can also install packages through the “packages panel on the right side of your interface as shown here:

After installing the package, you also have to check the box with the functions you want on the right here



**Here are some examples of useful packages for different purposes:**

To manipulate data

[dplyr](http://blog.rstudio.org/2014/01/17/introducing-dplyr/) - Essential shortcuts for subsetting, summarizing, rearranging, and joining together data sets. dplyr is our go to package for fast data manipulation.

[tidyr](http://blog.rstudio.org/2014/07/22/introducing-tidyr/) - Tools for changing the layout of your data sets. Use the gather and spread functions to convert your data into the [tidy format](http://www.jstatsoft.org/v59/i10/paper), the layout R likes best.

[stringr](http://journal.r-project.org/archive/2010-2/RJournal_2010-2_Wickham.pdf) - Easy to learn tools for regular expressions and character strings.

[lubridate](http://www.r-statistics.com/2012/03/do-more-with-dates-and-times-in-r-with-lubridate-1-1-0/) - Tools that make working with dates and times easier.

To visualize data

[ggplot2](http://docs.ggplot2.org/current/) - R's famous package for making beautiful graphics. ggplot2 lets you use the [grammar of graphics](http://vita.had.co.nz/papers/layered-grammar.pdf) to build layered, customizable plots.

[ggvis](http://ggvis.rstudio.com/) - Interactive, web based graphics built with the grammar of graphics.

[rgl](http://rgl.neoscientists.org/about.shtml) - Interactive 3D visualizations with R

[htmlwidgets](http://www.htmlwidgets.org/) - A fast way to build interactive (javascript based) visualizations with R. Packages that implement htmlwidgets include:

* [leaflet](http://rstudio.github.io/leaflet/) (maps)
* [dygraphs](http://rstudio.github.io/dygraphs) (time series)
* [DT](http://rstudio.github.io/DT/) (tables)
* [diagrammeR](http://rich-iannone.github.io/DiagrammeR/) (diagrams)
* [network3D](http://christophergandrud.github.io/networkD3/) (network graphs)
* [threeJS](https://github.com/bwlewis/rthreejs) (3D scatterplots and globes).

[googleVis](https://cran.rstudio.com/web/packages/googleVis) - Let's you use Google Chart tools to visualize data in R. Google Chart tools used to be called Gapminder, the graphing software Hans Rosling made famous in hie TED talk.

To model data

[car](http://www.rdocumentation.org/packages/car) - car's [Anova](http://www.rdocumentation.org/packages/car/functions/Anova) function is popular for making type II and type III Anova tables.

[mgcv](http://www.rdocumentation.org/packages/mgcv/functions/mgcv-package) - Generalized Additive Models

[lme4](http://www.rdocumentation.org/packages/lme4/functions/lme4-package)/[nlme](http://www.rdocumentation.org/packages/nlme/functions/nlme) - Linear and Non-linear mixed effects models

[randomForest](http://www.rdocumentation.org/packages/randomForest/functions/randomForest) - Random forest methods from machine learning

[multcomp](http://www.rdocumentation.org/packages/multcomp) - Tools for multiple comparison testing

[vcd](http://www.rdocumentation.org/packages/vcd) - Visualization tools and tests for categorical data

[glmnet](http://www.rdocumentation.org/packages/glmnet/functions/glmnet) - Lasso and elastic-net regression methods with cross validation

[survival](http://www.rdocumentation.org/packages/survival) - Tools for survival analysis

[caret](https://cran.rstudio.com/web/packages/caret) - Tools for training regression and classification models