# Lecture 4: A (very brief) intro to plotting R:

## **Types of graphical analysis**

### For explanatory analysis

* Getting to know my data
* Rough and ready plot of chunk rough

### For reporting

* Presenting my data to others
* Tells a standalone story
* polished plot of chunk polished

## **Base R graphics**

## Base graphics (*graphics* and *grDevices*):

* these packages are powerful graphing function to base R.

## Graphing process in base R

* Initialise a new plot
  + Plotting functions: *plot(x,y,...)*, *boxplot()*, *hist()*, *barplot(height, ...)*, *mosaicplot(x,...)*.
  + Improve my plot (find key plot parameters with *?par*):
* Add to the new plot (once I have it) +in the form of ablines: *abline()*, *lines()*, *points()*, *text()*

## Saving my plot

I can save my plot to a file by specifying the graphics device before I start plotting the figure [e.g., *png()*, *pdf()*]. The *pdf* is prefereable as it is saved in vector format. Remember to give a file name [e.g., *pdf('foo.pdf')*. For *pdf* outputs, I can also change the page orientation to landscape [e.g., \_pdf('foo.pdf, paper = 'a4R')]. If I open a device, I should remember to close it at the end [\_dev.off()], otherwise the file won't save. By default, the file will be saved to my working directory.

## The grammar of graphics and ggplot2

* Leland Wilkinson conceptualised a grammar for graphics that would move graphics passed mere chart types (*words*), like pie-charts, bar-charts, etcc.
* By providing a grammar for graphics, he heoped the provide the framework to build and unlimited array of graphic forms (*statements*).

## ggplot2

### Basic components of a ggplot2 plot:

* data (*in a dataframe*)
* Aesthetics (*how data are mapped to colour , size, etc*)
* Geoms (*gemetric objects like points, lines, shapes*)
* Coordinate system

## IN SWIRL:

* The *head(data.set)* function provides the firt few values of the data set, instead of the whole
* To save some typing, call the R command *with* using 2 arguments. The first will be the data set containing which ever variables, and the second will be the call to plot (whichever type, not only scatter as implied). Doing this allows me to avoid using the longer notation, [e.g., data.set$'variable'].
* The *par()* function is used to specify global graphics parameters that affect all plots in an R session. Use *dev.off* or *plot.new* to reset the defaults.
* The argument *type = n* tells R to set up the plot, but not to put the data in it.