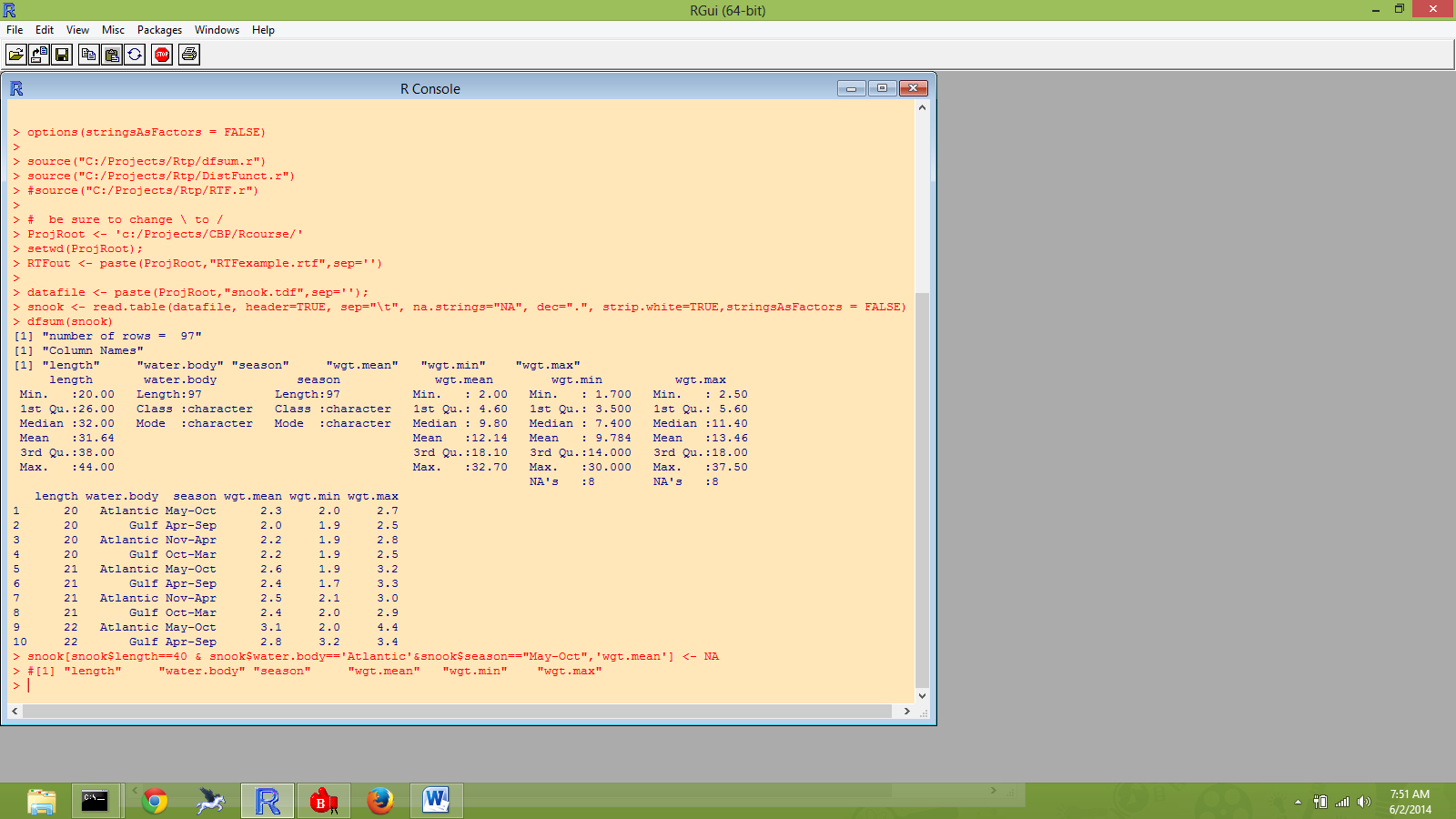
Rbit003

**Functions for some Common Data Analyses**

In this Rbit, we cover some functions to conduct a variety of common data analyses. None of these are covered in depth. My goal is to get you a set of tools that will enable you to start some productive work as quickly as possible and then start filling in the gaps with more depth. To avoid the issue of MSword substituting characters that the R-language cannot interpret, I have provided the R-scripts for these examples in a separate file - CommonAnalysesRbit003.r.

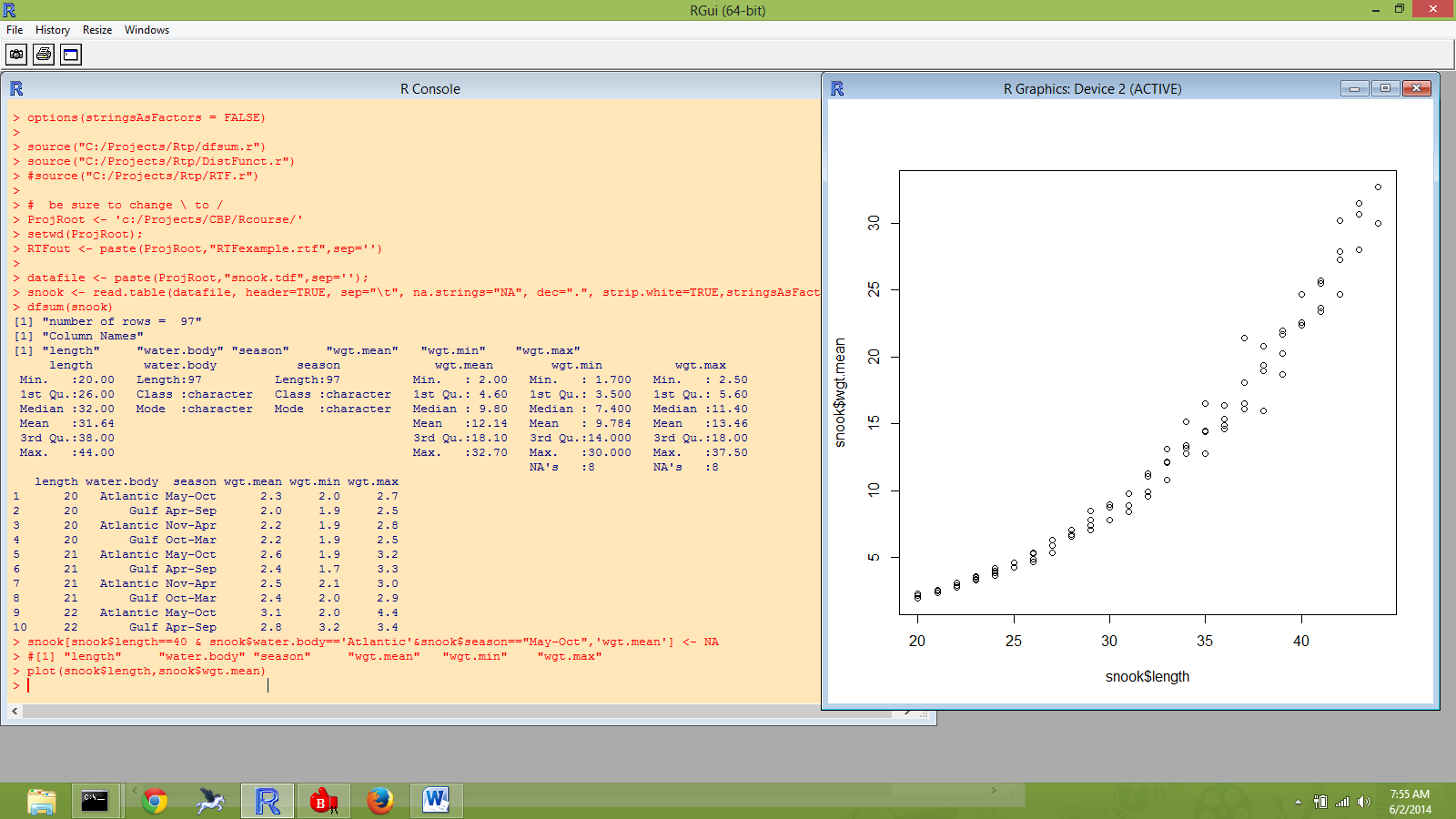
First read in the Snook data –



**Scatter Plot**

Scatter plots we have seen before. This simplest set of arguments for the plot function is just to provide the x-variable and the y-variable like this:

plot(snook$length,snook$wgt.mean)



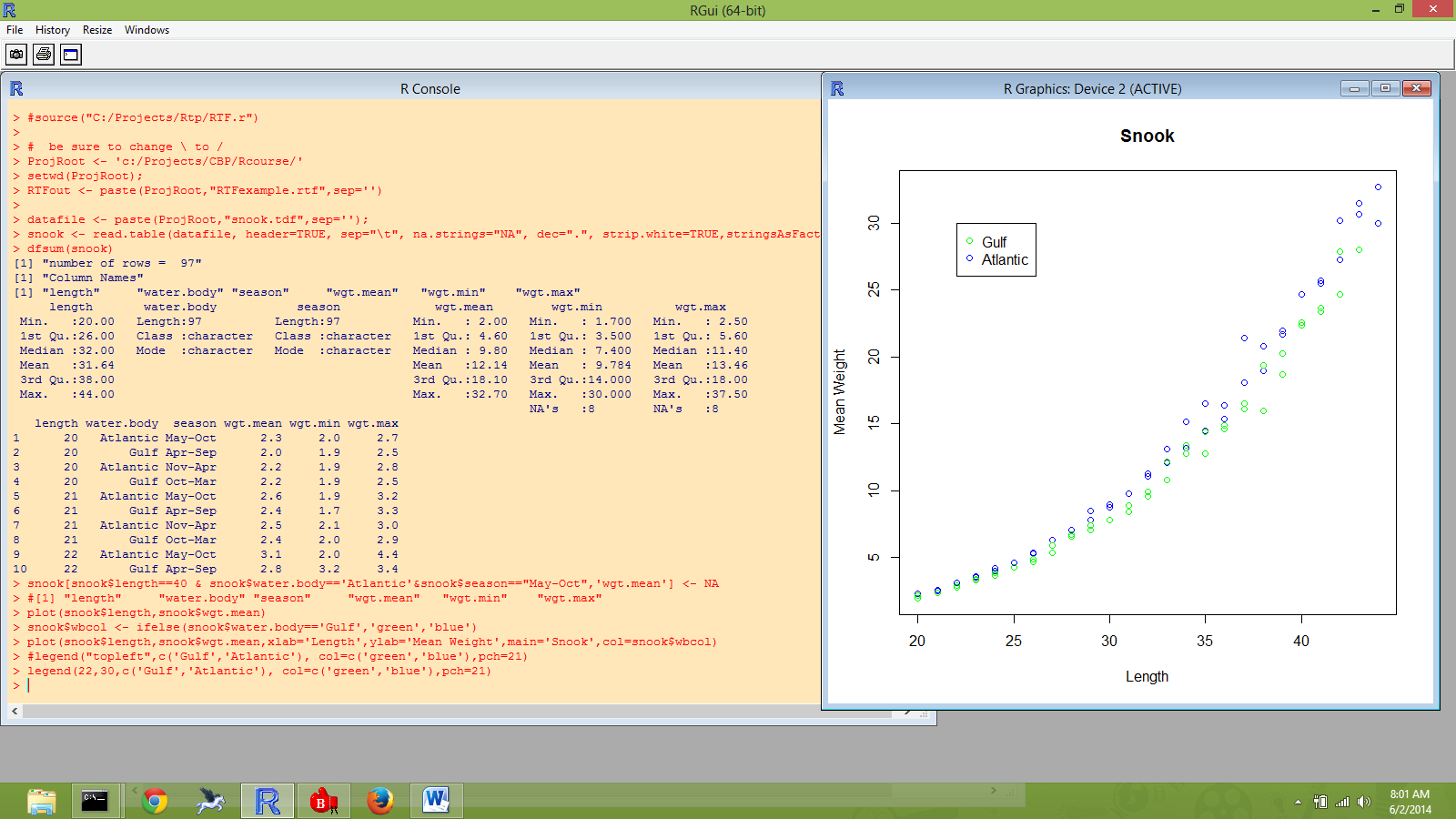
Next is an example using additional arguments to add some dressing to the plot and subsequently add a legend using the legend function. First I create a new column in the data frame, wbcol (water body color), that I then refer to in the plot call with col=snook$wbcol. This option is what produce the two colors for the two waterbodies. Check out ?plot if you have questions about other options. Note that I first positioned the legend using the “topleft” of legend, and then decided to place the legend by explicitly giving the (X,Y) coordinates.

snook$wbcol <- ifelse(snook$water.body=='Gulf','green','blue')

plot(snook$length,snook$wgt.mean,xlab='Length',ylab='Mean Weight',main='Snook',col=snook$wbcol)

#legend("topleft",c('Gulf','Atlantic'), col=c('green','blue'),pch=21)

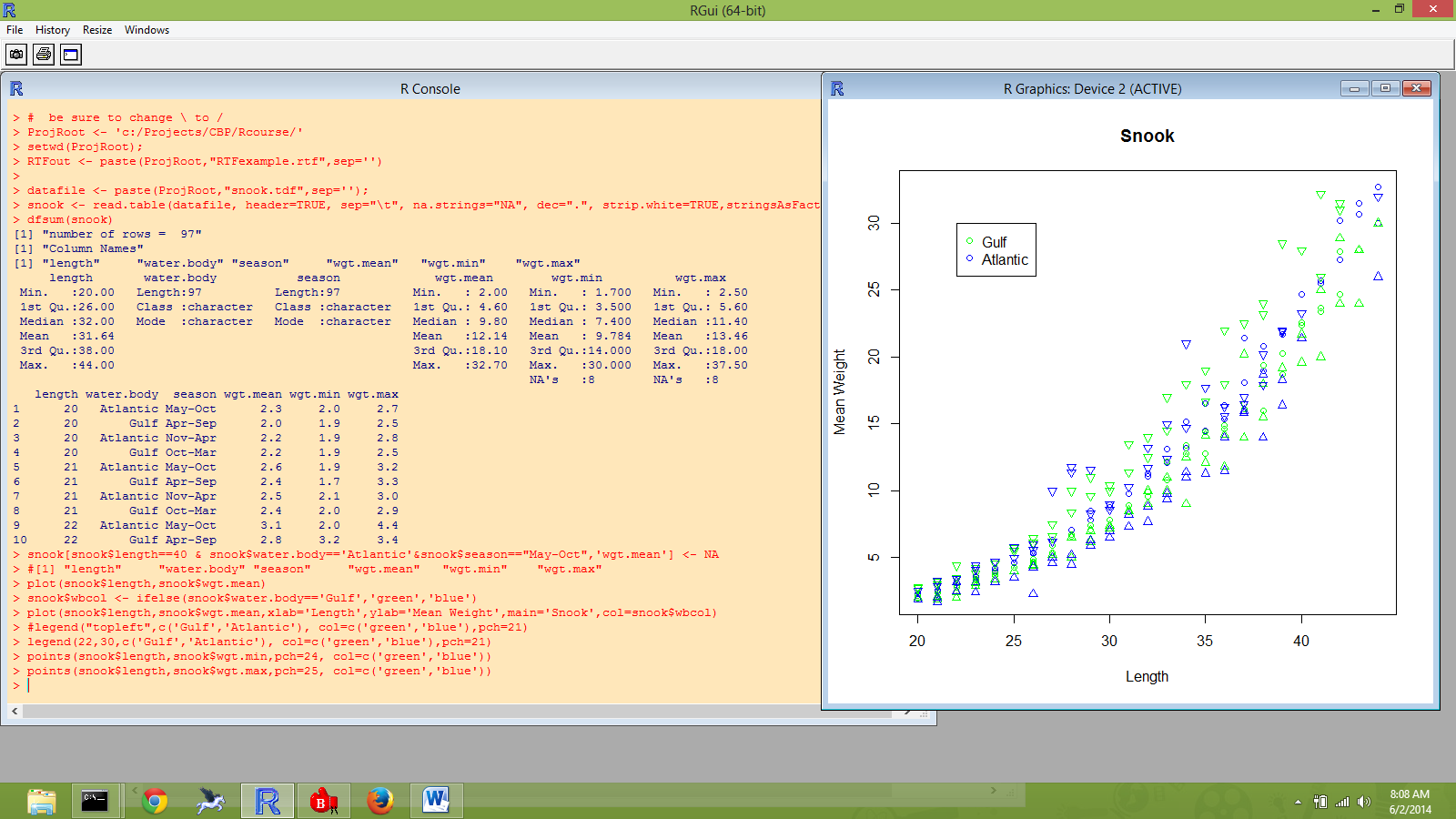
legend(22,30,c('Gulf','Atlantic'), col=c('green','blue'),pch=21)



You can subsequently add information to a plot using the points and lines functions. Here I add the max and min for each length and waterbody. The result is so busy that it is difficult to interpret.

points(snook$length,snook$wgt.min,pch=24, col=c('green','blue'))

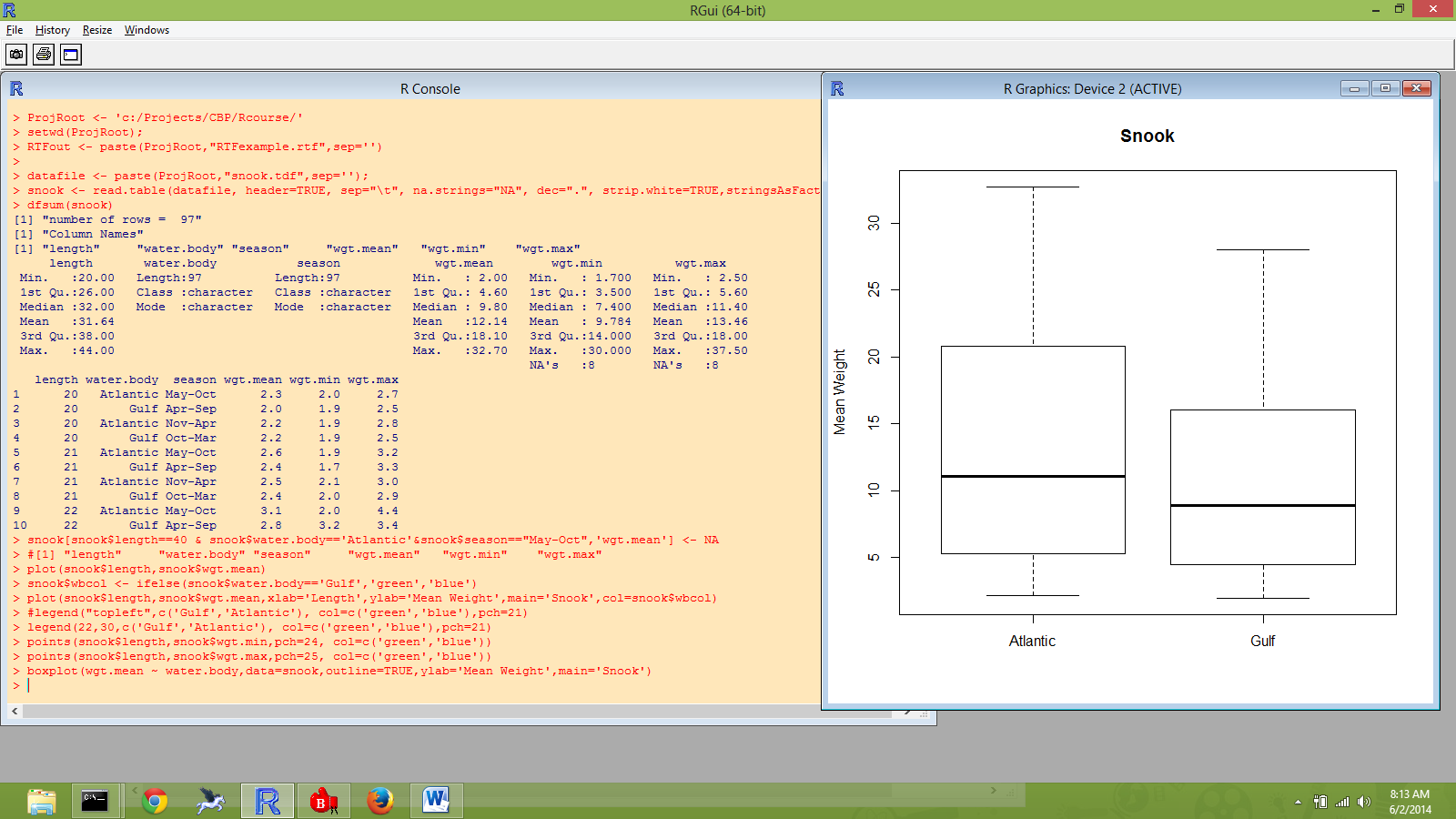
points(snook$length,snook$wgt.max,pch=25, col=c('green','blue'))



Box Plot

If you have continuous response variables grouped by categories, the box plot is often a useful display. Here I call boxplot to show mean.wgt by waterbody.

boxplot(wgt.mean ~ water.body,data=snook,outline=TRUE,ylab='Mean Weight',main='Snook')

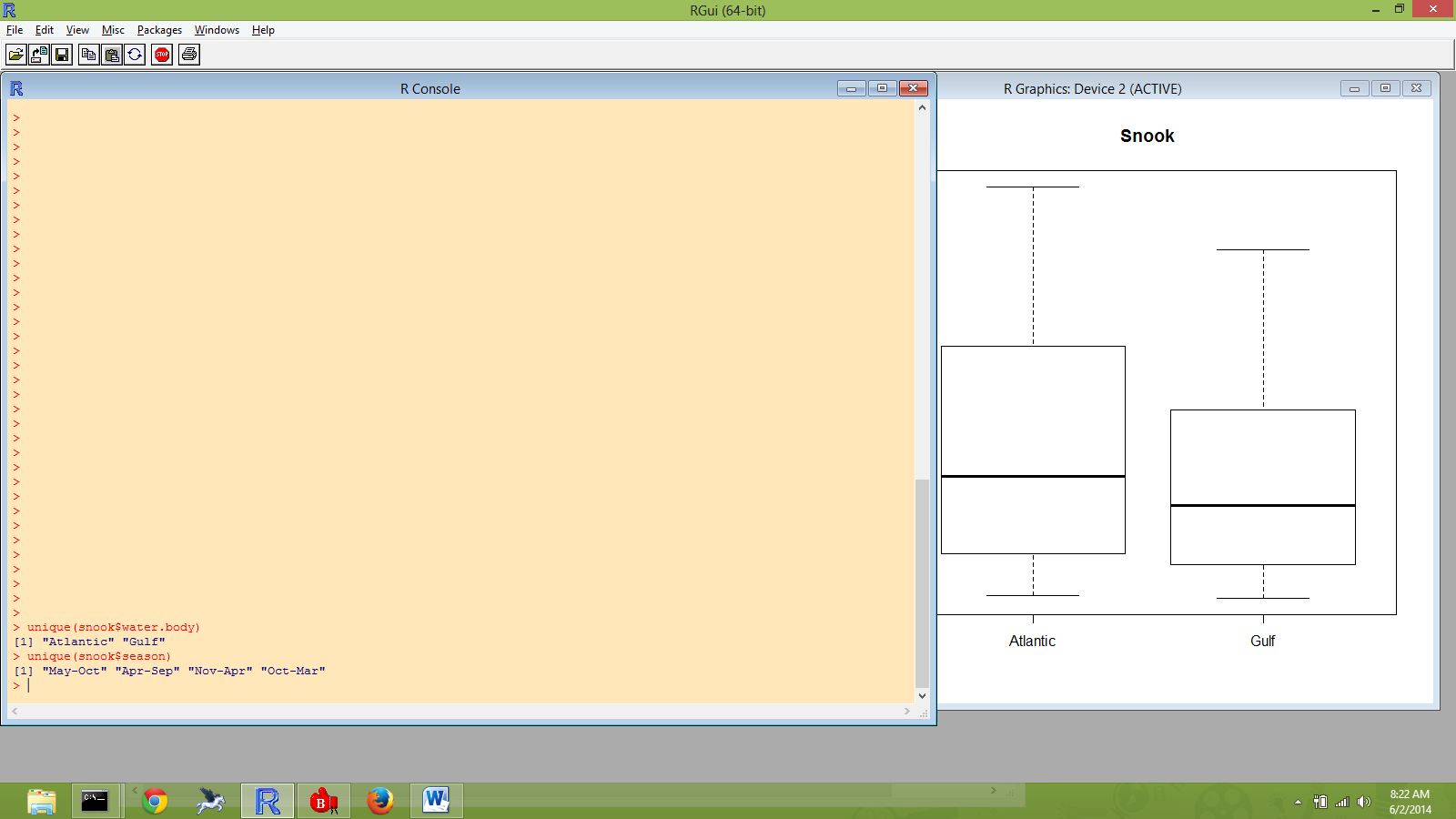


Next I turn to computing simple statistics of data. In the R-language, these are vector argument functions. This means that they accept a column of numbers as their argument. In this case the column of numbers is a column out of a data frame, but it could be a column of numbers assigned to a single variable. We will cover this idea of vector variables later. If you are anxious to learn about it, the introductory guide covers this well.

The unique function is handy for checking the number of categories present in data.

unique(snook$water.body)

unique(snook$season)



Here are a select few of the many descriptive statistics available in the R-language.

mean(snook$wgt.mean)

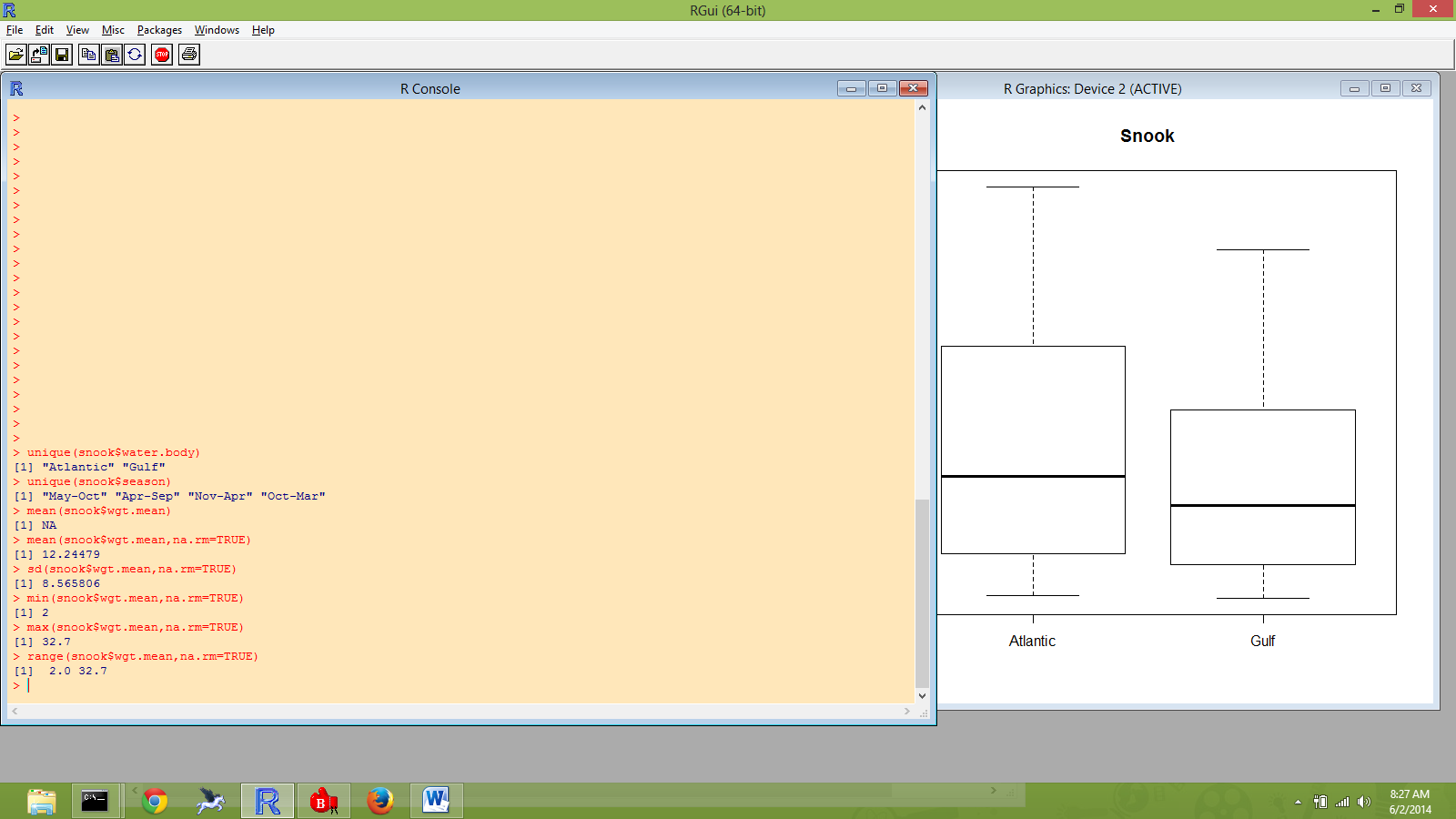
mean(snook$wgt.mean,na.rm=TRUE)

sd(snook$wgt.mean,na.rm=TRUE)

min(snook$wgt.mean,na.rm=TRUE)

max(snook$wgt.mean,na.rm=TRUE)

range(snook$wgt.mean,na.rm=TRUE)

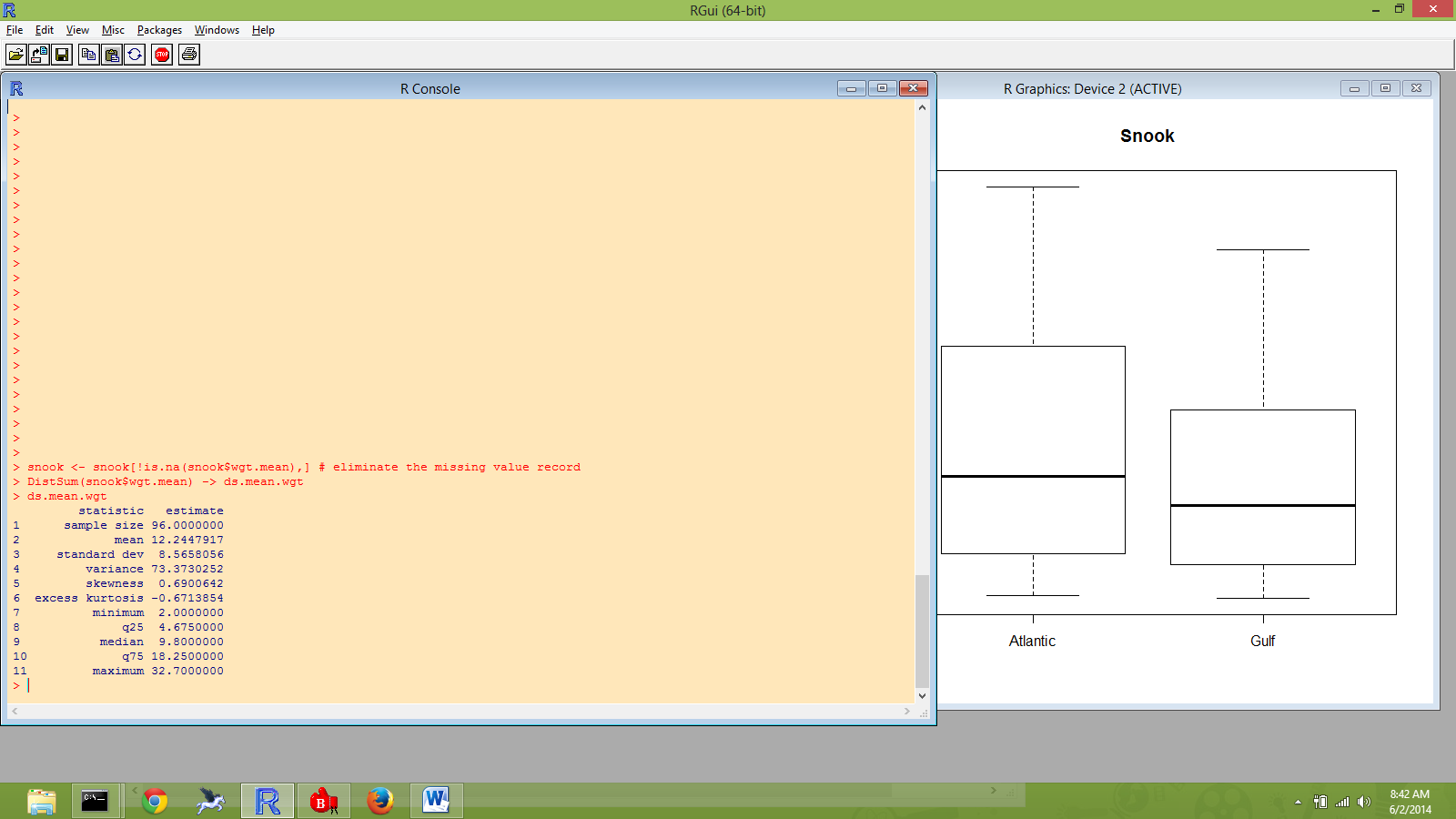


My first call to mean result in NA (not available), because one value of the wgt.mean column is NA. Recall, that I set this to NA because is appeared to be a typo. In my next call to mean, I added the option na.rm = TRUE which asks R to remove the NA observations when computing the mean. I used this option for the other function calls as well. Executing all of these individual functions may seem a bit tedious as compared to call “Proc Univariate” in SAS. I have written a user defined function, DistSum, that produces a collection of summary statistics for one vector. Before calling my function, I make a new version of the snook data frame than eliminates the row with the missing value for wgt.mean.

snook <- snook[!is.na(snook$wgt.mean),] # eliminate the missing value record

DistSum(snook$wgt.mean) -> ds.mean.wgt

ds.mean.wgt



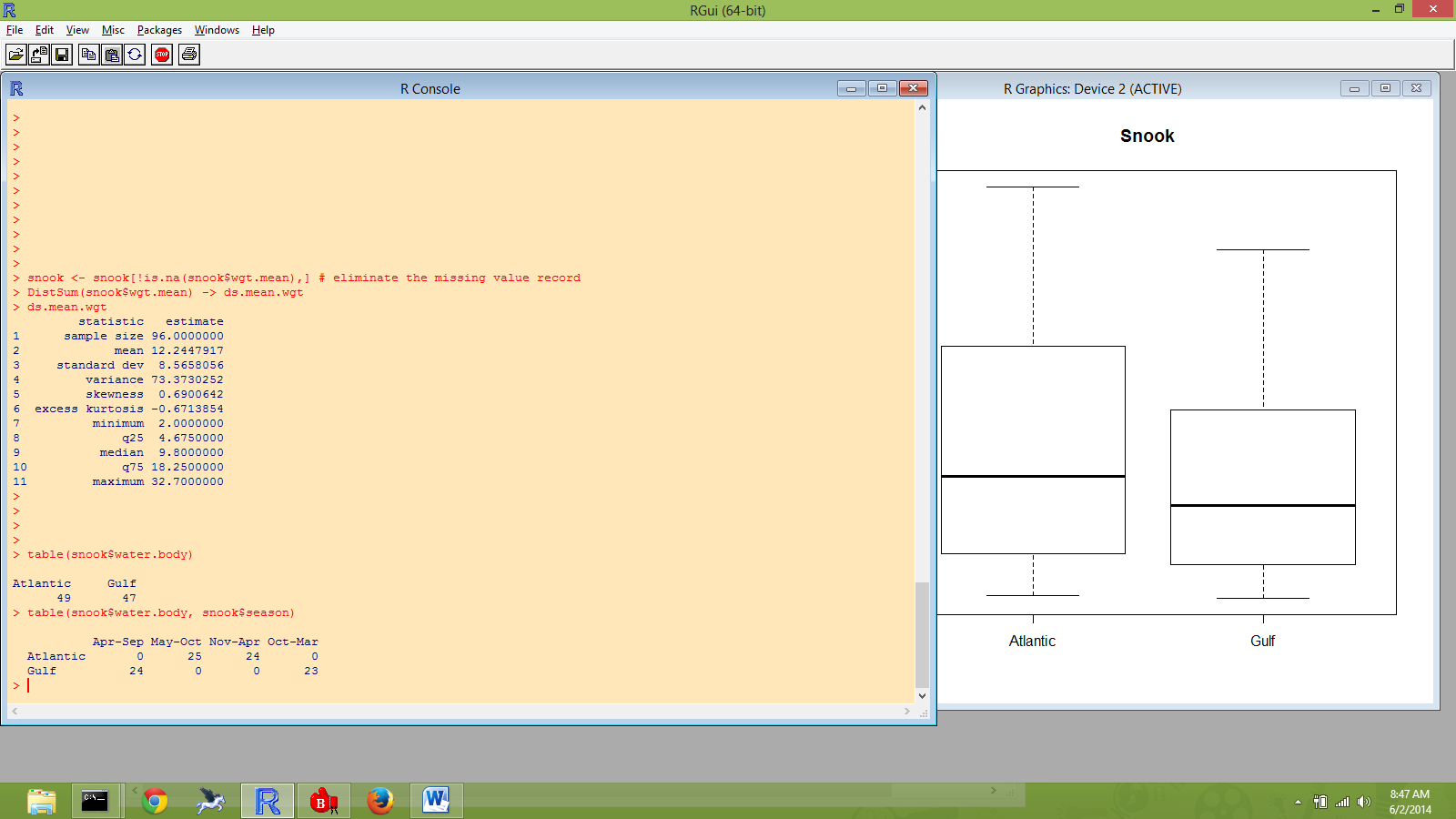
User defined functions is an upcoming topic.

**Tables**

Frequency Tables are another common summary for data and these are created by the table function.

table(snook$water.body)

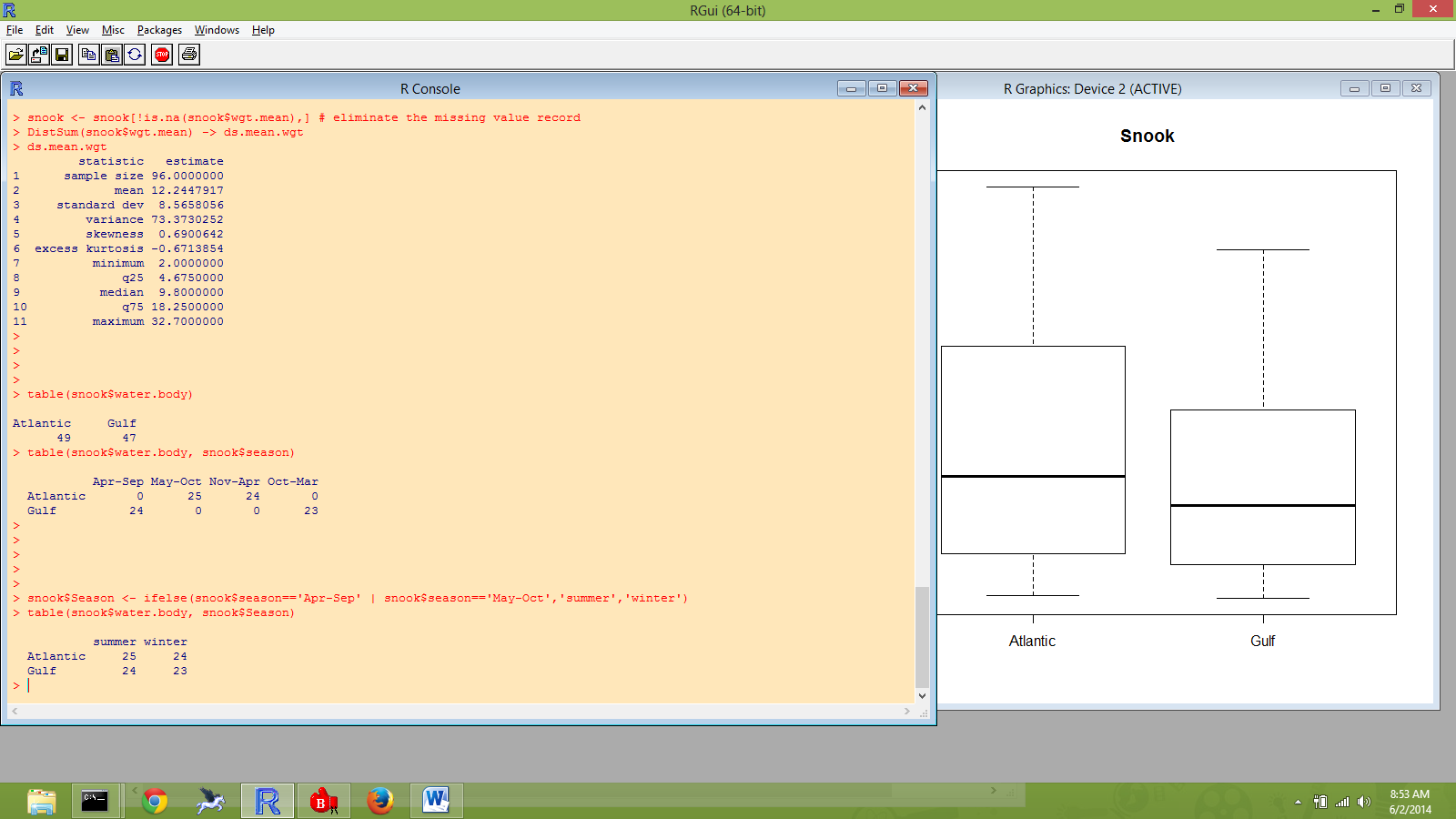
table(snook$water.body, snook$season)



Here I discovered that the seasons for the two waterbodies are did not use the same period of months. I defined a new Season column (see below) using an upper-case ‘S’ to differentiate it from the original season. The table is recomputed with the new Season column.

snook$Season <- ifelse(snook$season=='Apr-Sep' | snook$season=='May-Oct','summer','winter')

table(snook$water.body, snook$Season)



I have failed to mention before now, that you can always save the results of a function call using the ‘<-‘ or ‘->’ assignment operators. This makes it easy to use the results for future calculations. Here the results of table are a 2x2 array and you can refer to the individual elements using [row,column] notation.

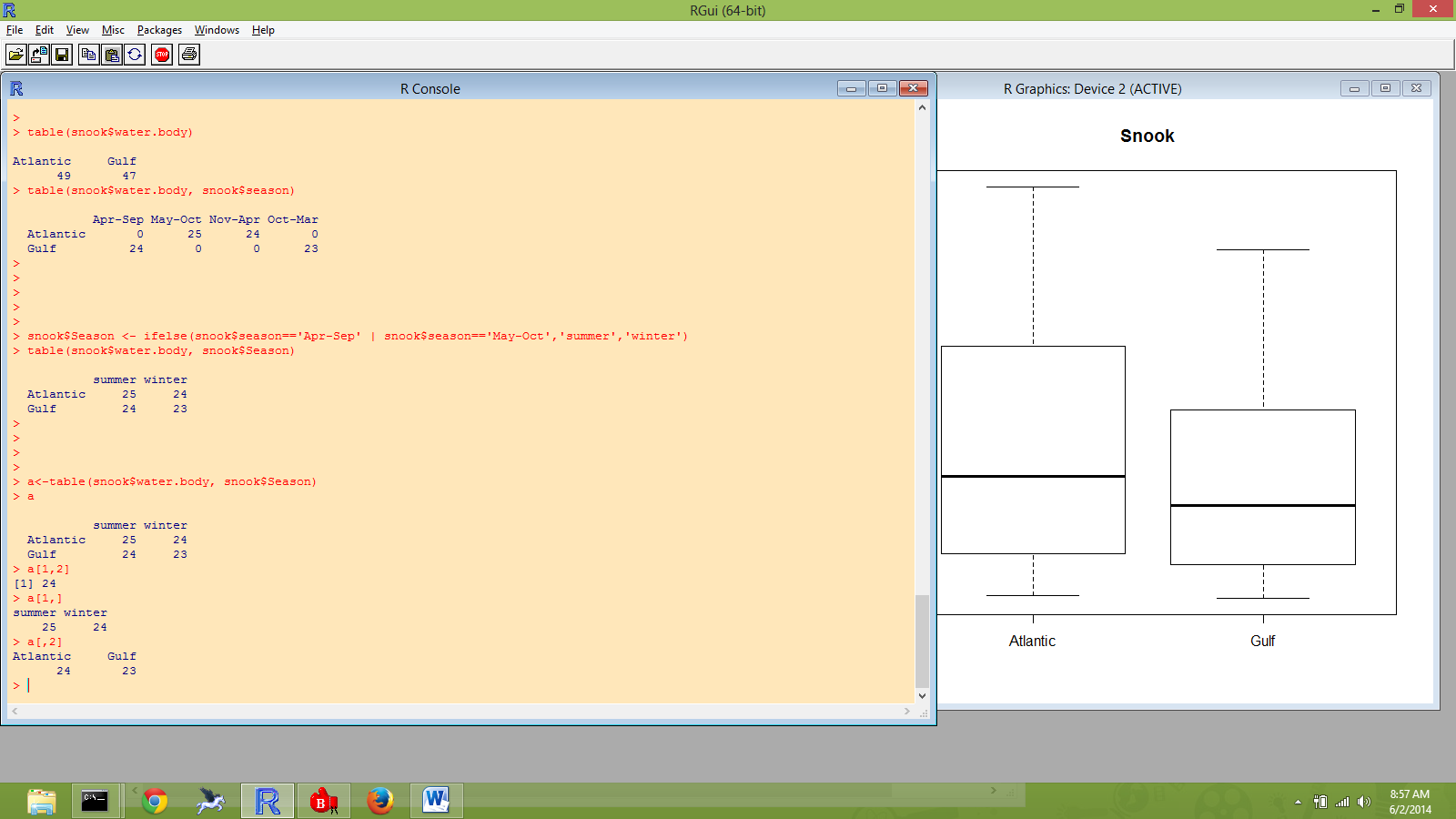
a<-table(snook$water.body, snook$Season)

a

a[1,2]

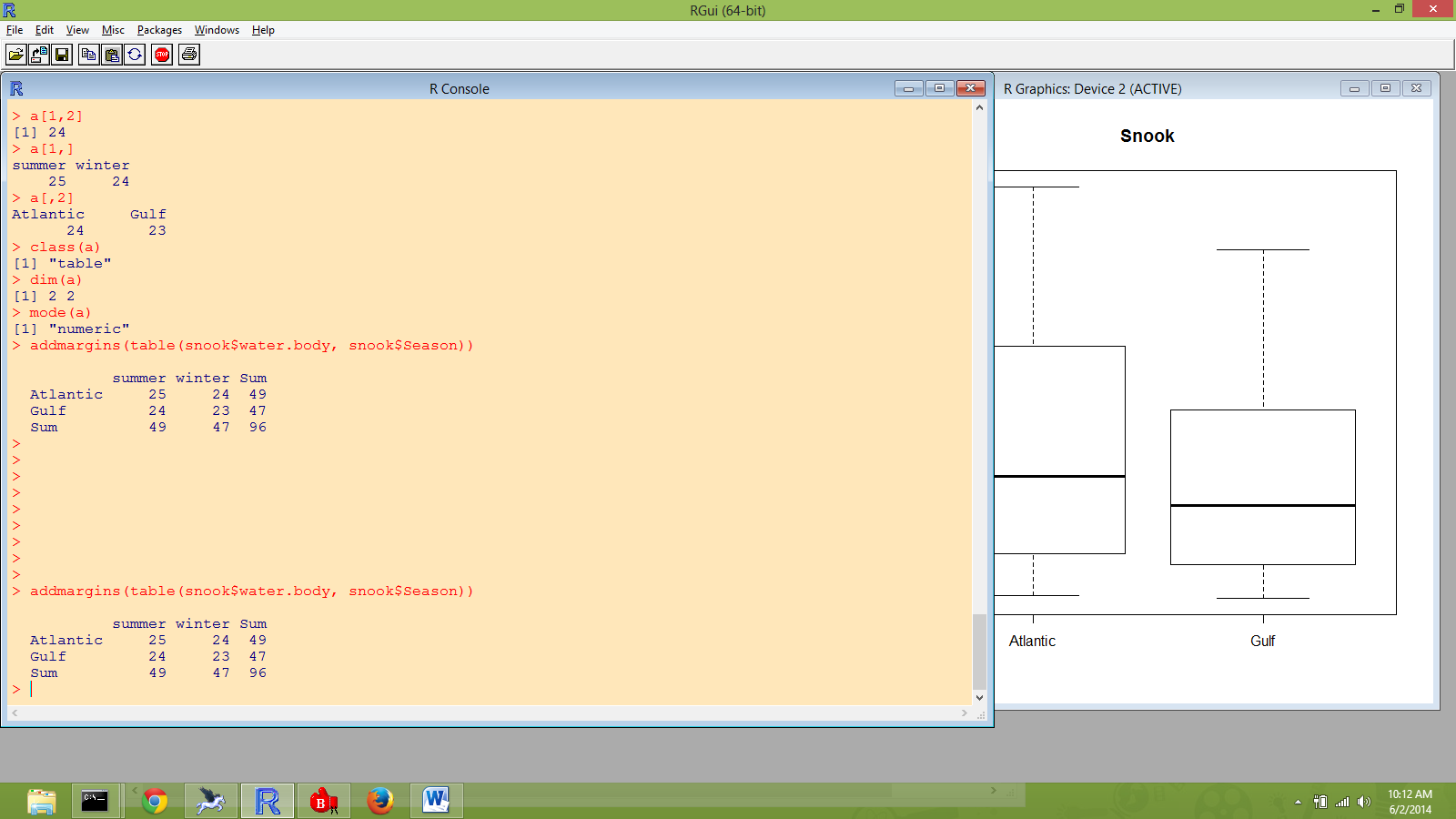
a[1,]

a[,2]



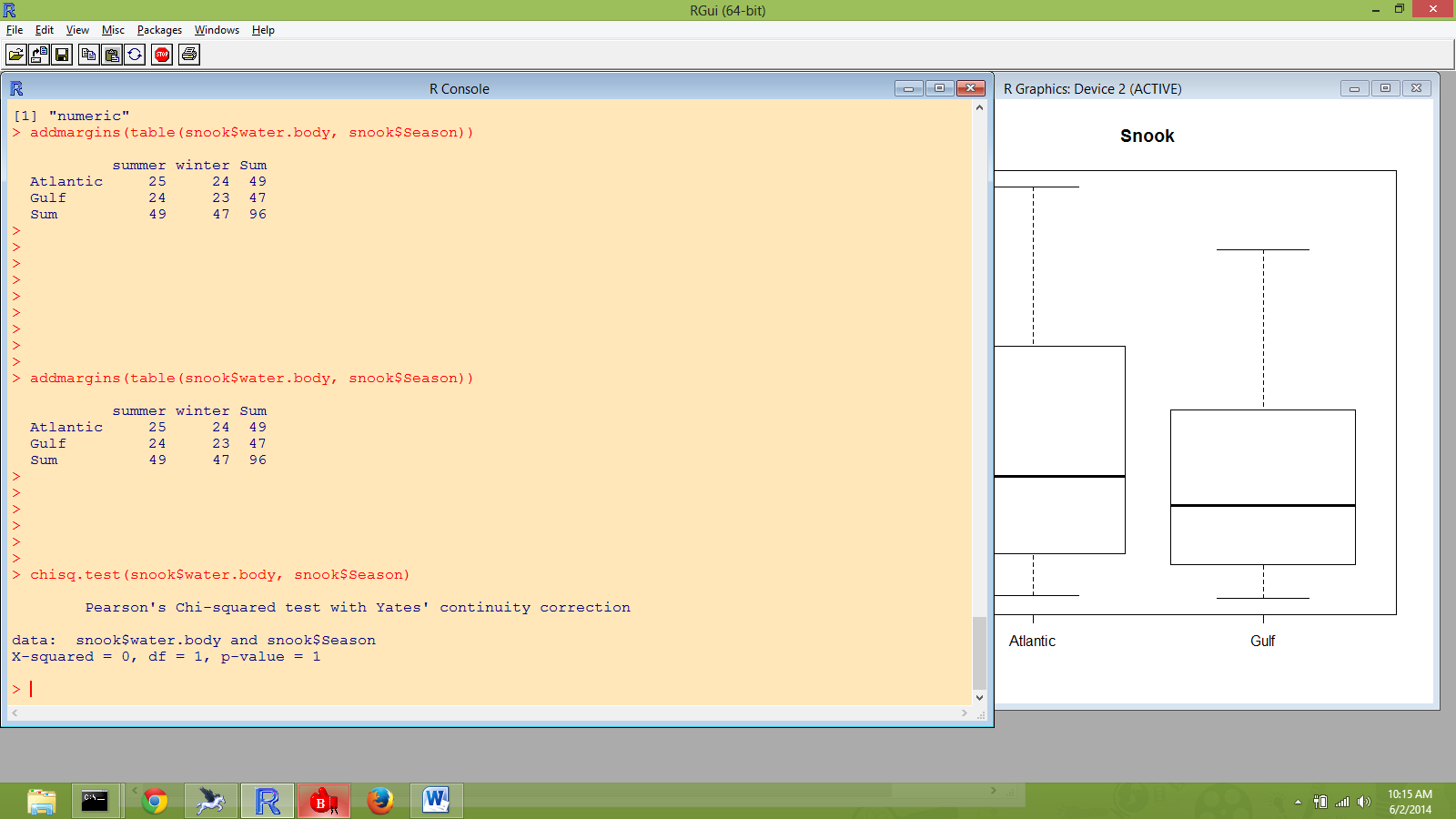
The addmargins function can be wrapped around the table function to create a table with marginal totals.

addmargins(table(snook$water.body, snook$Season))



To get a chi-square test on a two-way table use the chisq.test function.

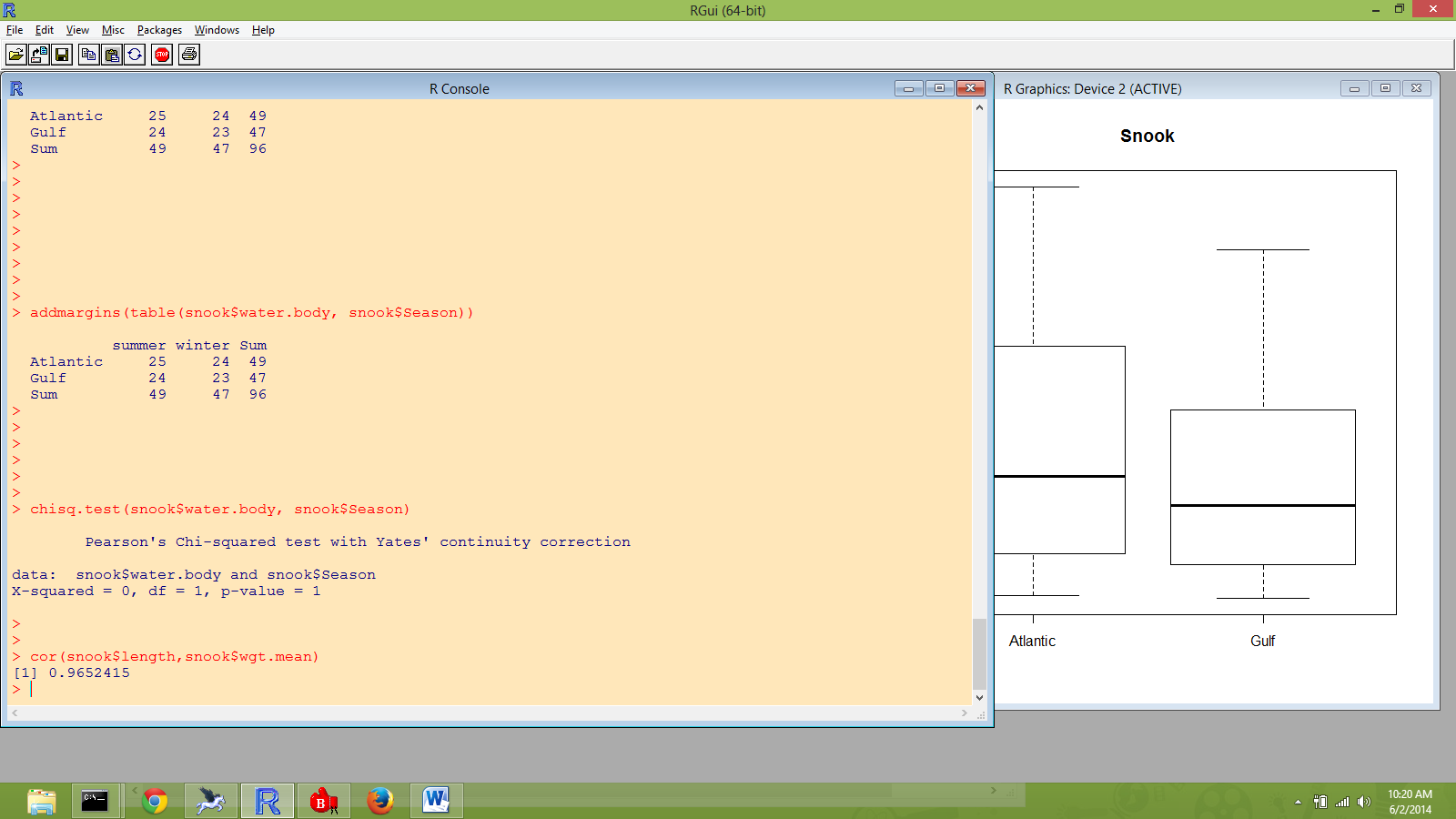
chisq.test(snook$water.body, snook$Season)



**Correlation**

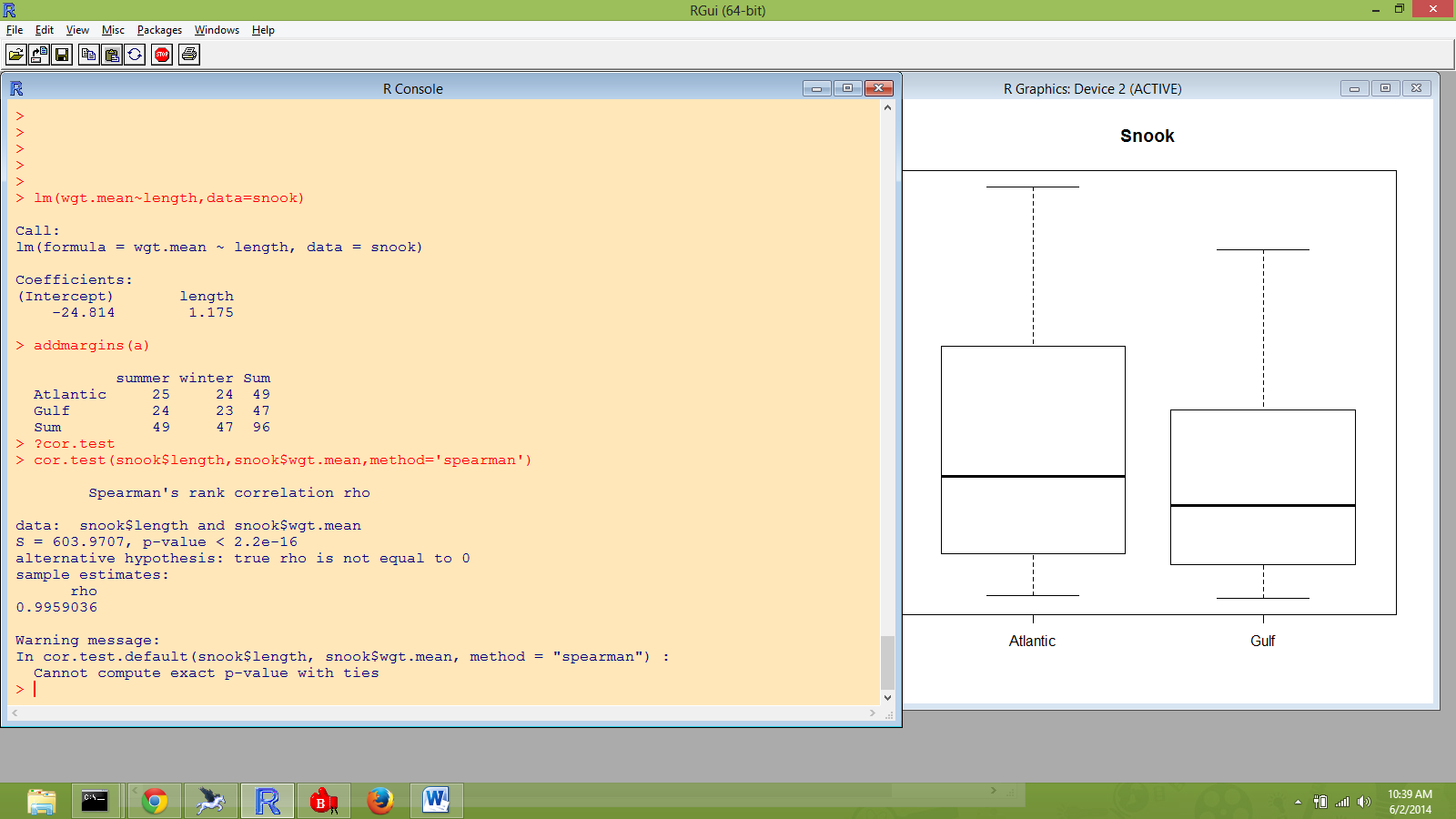
The correlation function is fairly self explaining.

cor(snook$length,snook$wgt.mean)



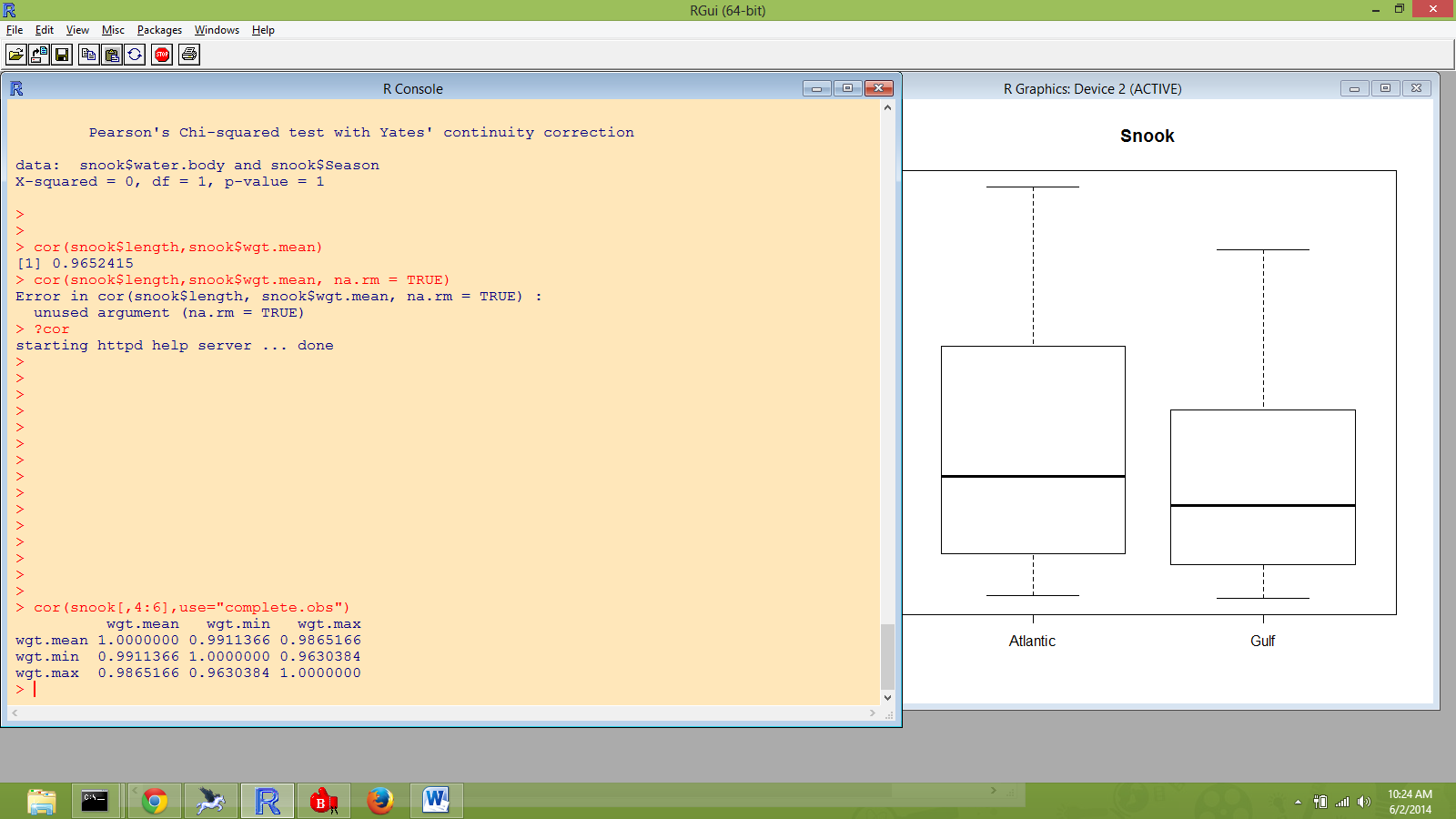
To get a p-value for the correlation, use the cor.test function. It offers choices of Pearson product moment, Spearman rank, or Kendall tau methods.

cor.test(snook$length,snook$wgt.mean,method='spearman')



The cor function will produce a correlation matrix if given a matrix of data.

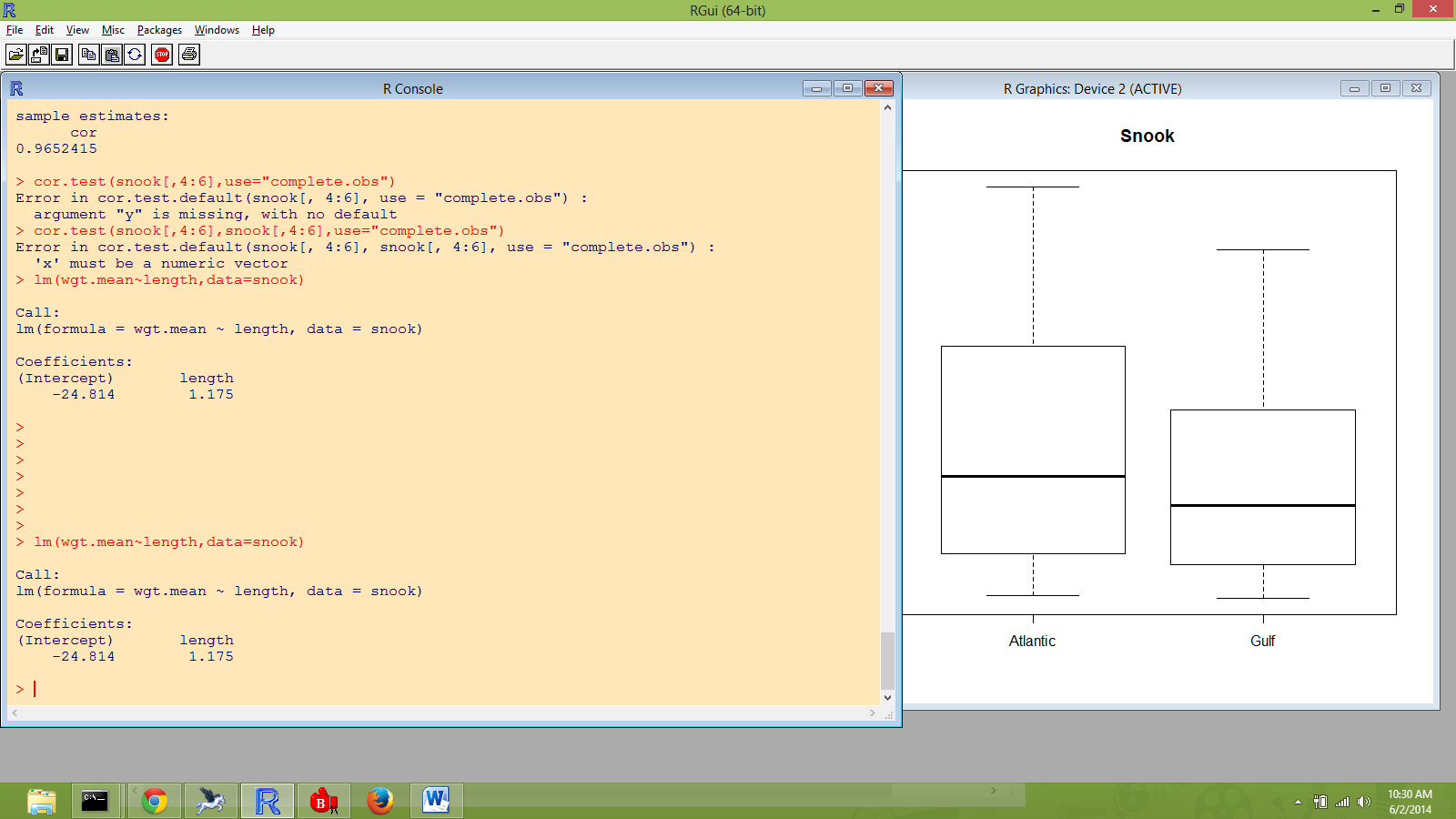
cor(snook[,4:6],use="complete.obs")



**Linear Regression**

Linear regression as well as other linear models (anova, ancova) are computed by the lm function. A simple call to lm will return the model coefficients.

lm(wgt.mean~length,data=snook)



More often it is useful to save the output of the lm function as an object for future use.

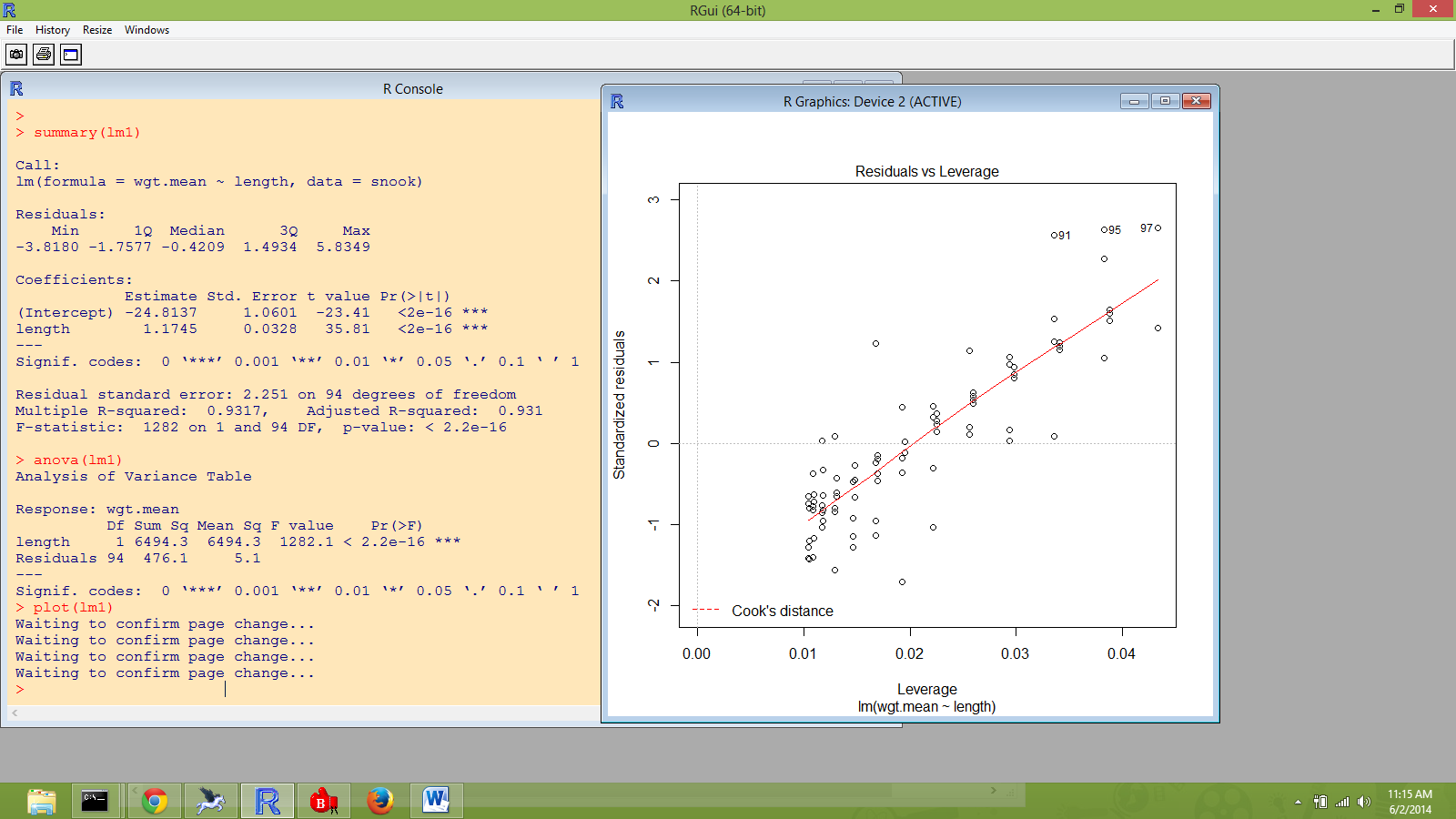
lm1 <- lm(wgt.mean~length,data=snook)

Functions in the R-language are often context sensitive. For example, when the plot() function is called with two vector arguments, it produces a scatter plot. When the plot() function is called with a linear models object, it produces a set of plots related to the linear model fit. Here are some context sensitive functions used with the object lm1.

summary(lm1)

anova(lm1)

plot(lm1)



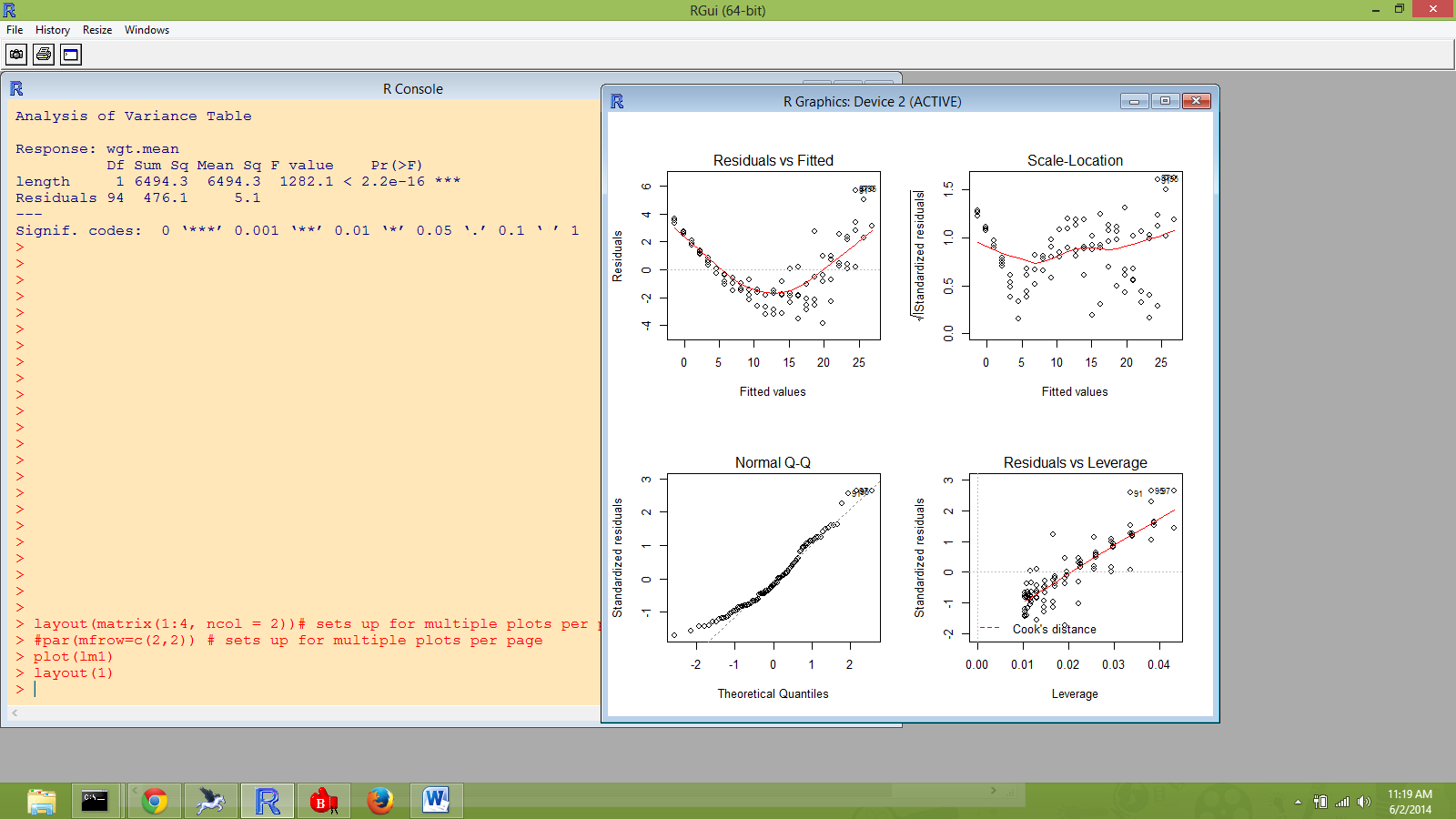
All plots can be put on one page using the layout() function or the par() function.

layout(matrix(1:4, ncol = 2))# sets up for multiple plots per page

#par(mfrow=c(2,2)) # sets up for multiple plots per page

plot(lm1)

layout(1)

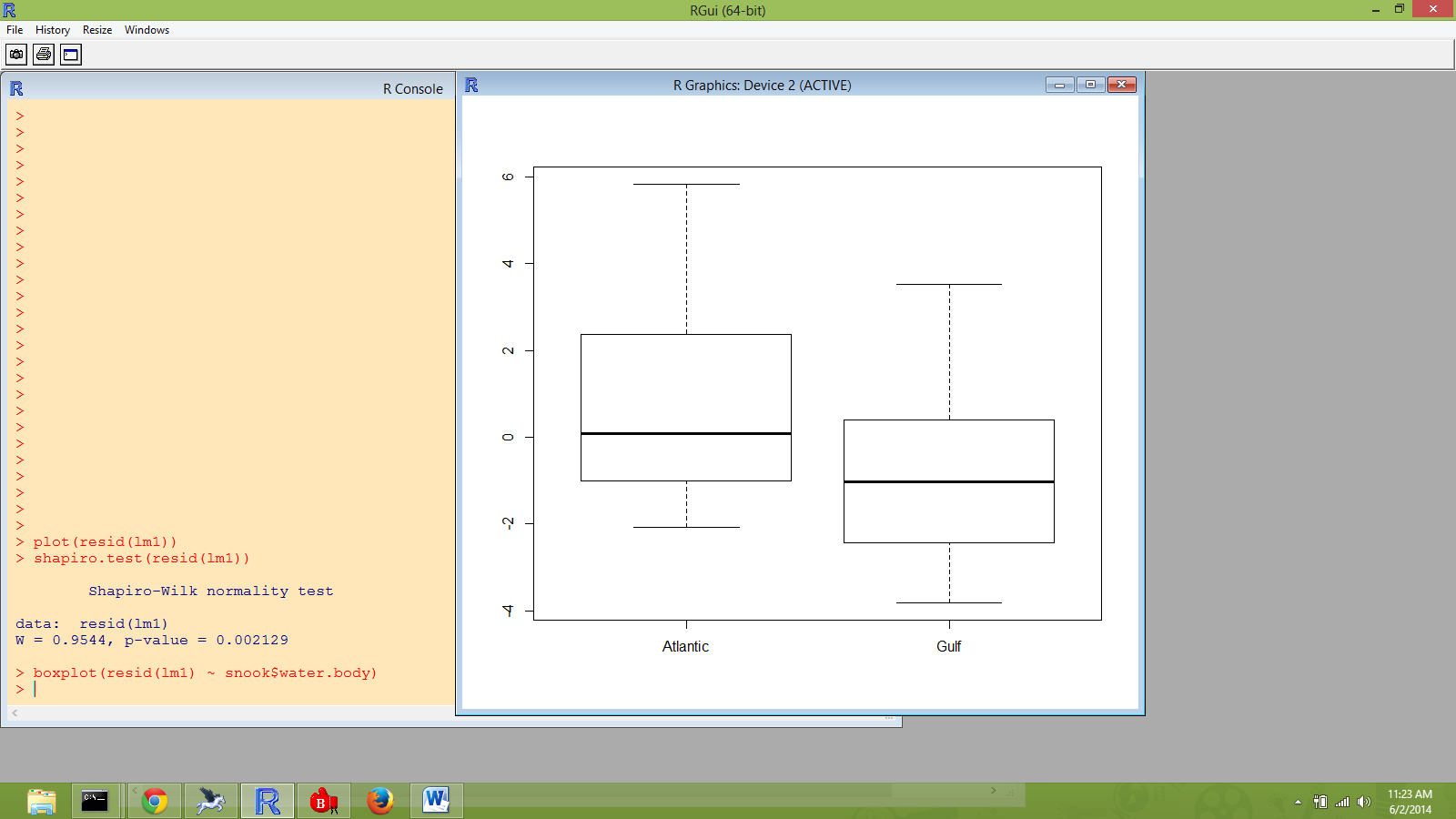


Other regression diagnostics using the lm1 object.

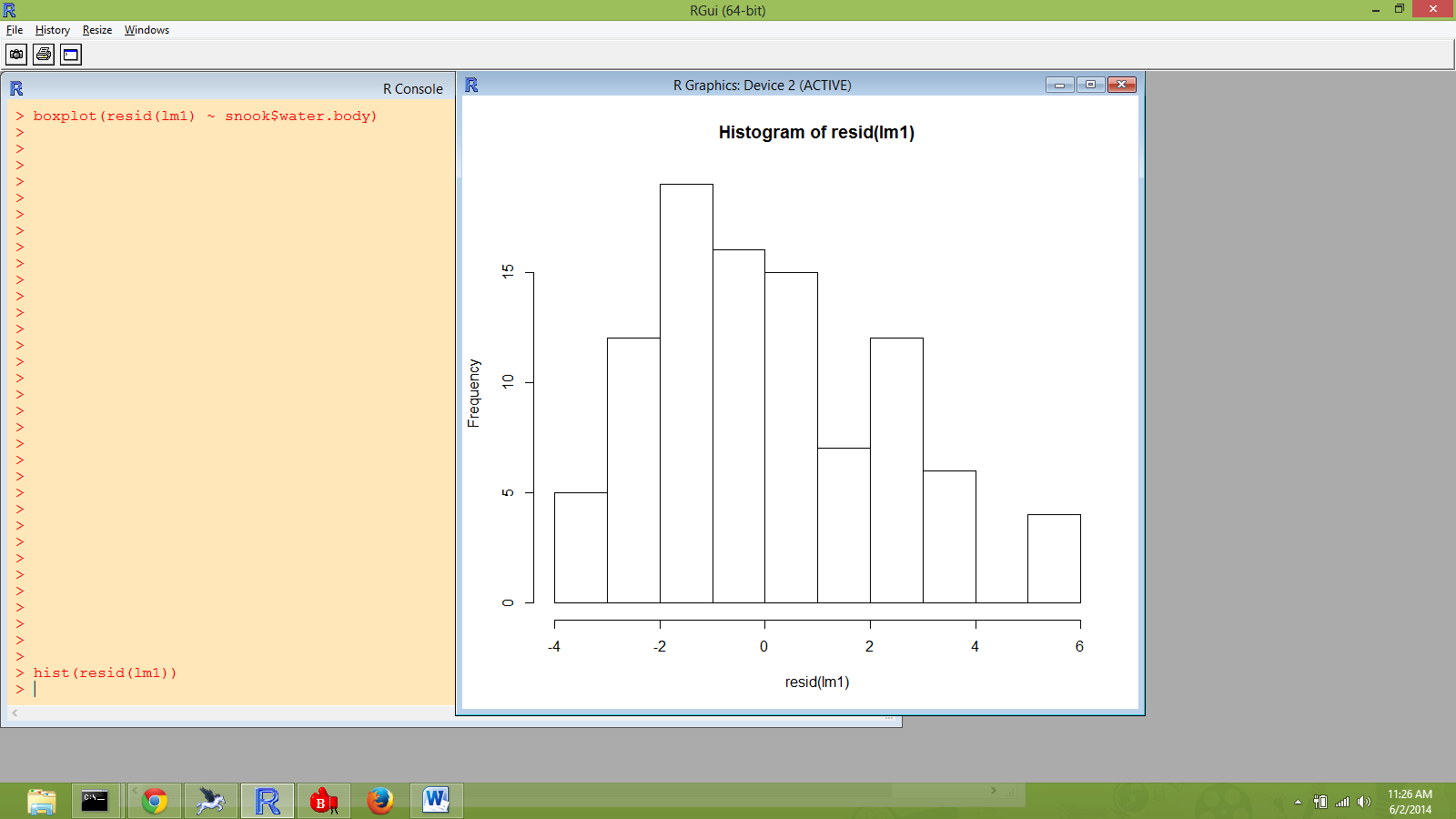
plot(resid(lm1))

shapiro.test(resid(lm1))

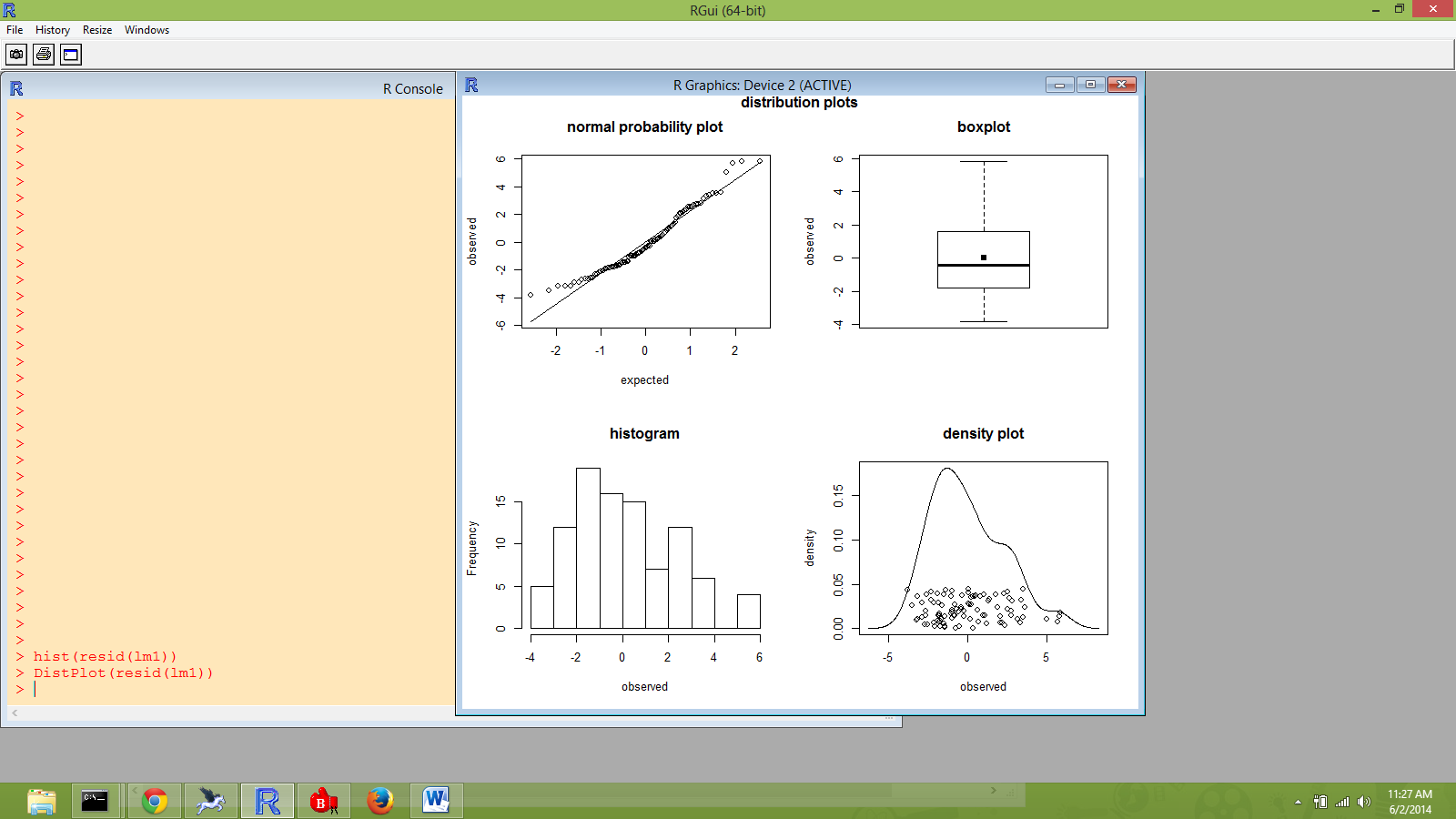
boxplot(resid(lm1) ~ snook$water.body)



hist(resid(lm1))



DistPlot(resid(lm1))



A classic regression plot is to show the data plus the predicted curve.

# plot of data and predicted values

plot(snook$length,snook$wgt.mean,xlab='Length',ylab='Mean Weight',main='Snook',col=snook$wbcol)

lines(snook$length,predict(lm1),lwd=2,col='red')



In the figure above, we see the poor fit of linear regression for these data. Alternatively we might try a log linear regression. Notice that the log function gets embedded in the lm() arguments.

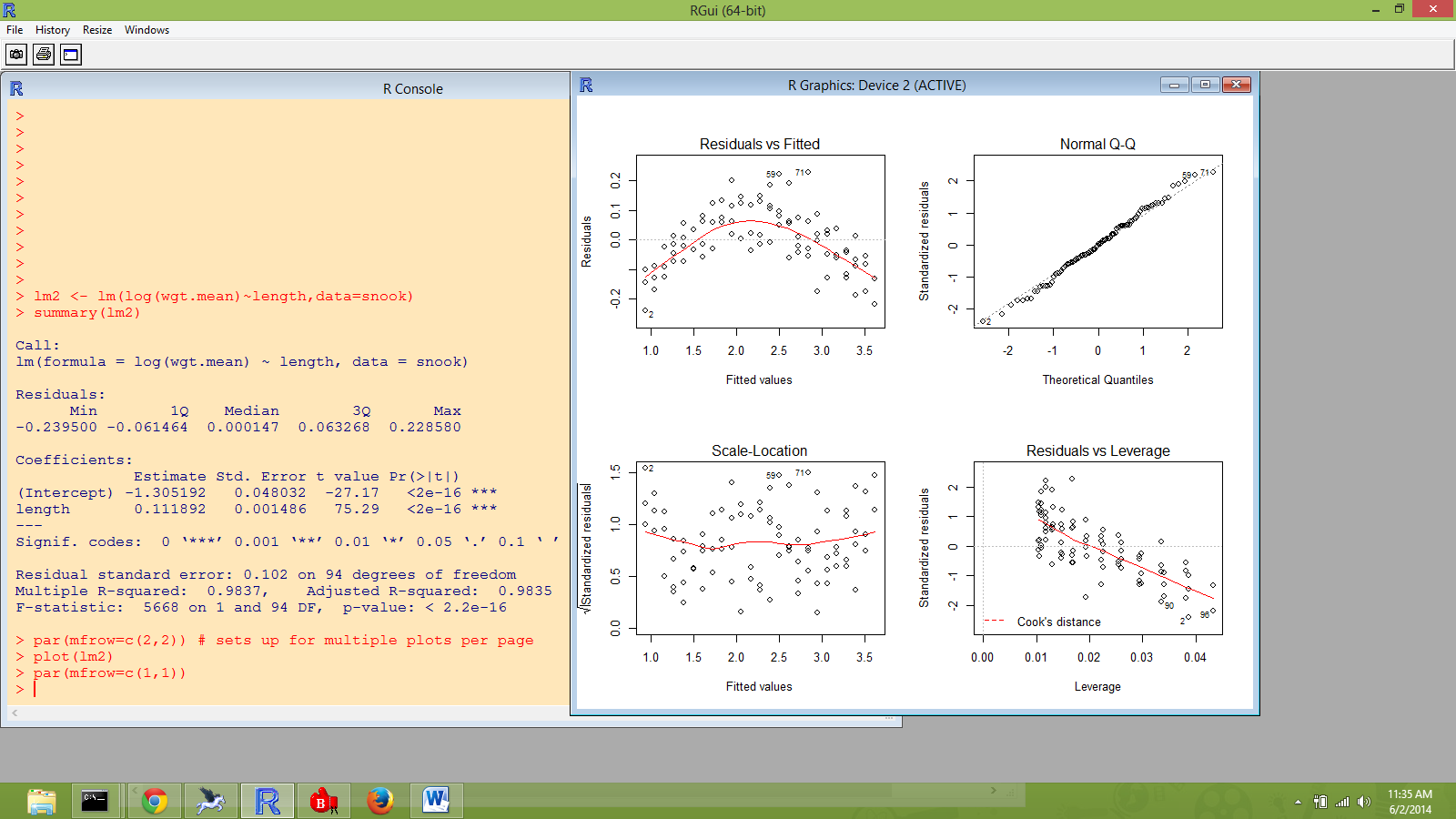
lm2 <- lm(log(wgt.mean)~length,data=snook)

summary(lm2)

par(mfrow=c(2,2)) # sets up for multiple plots per page

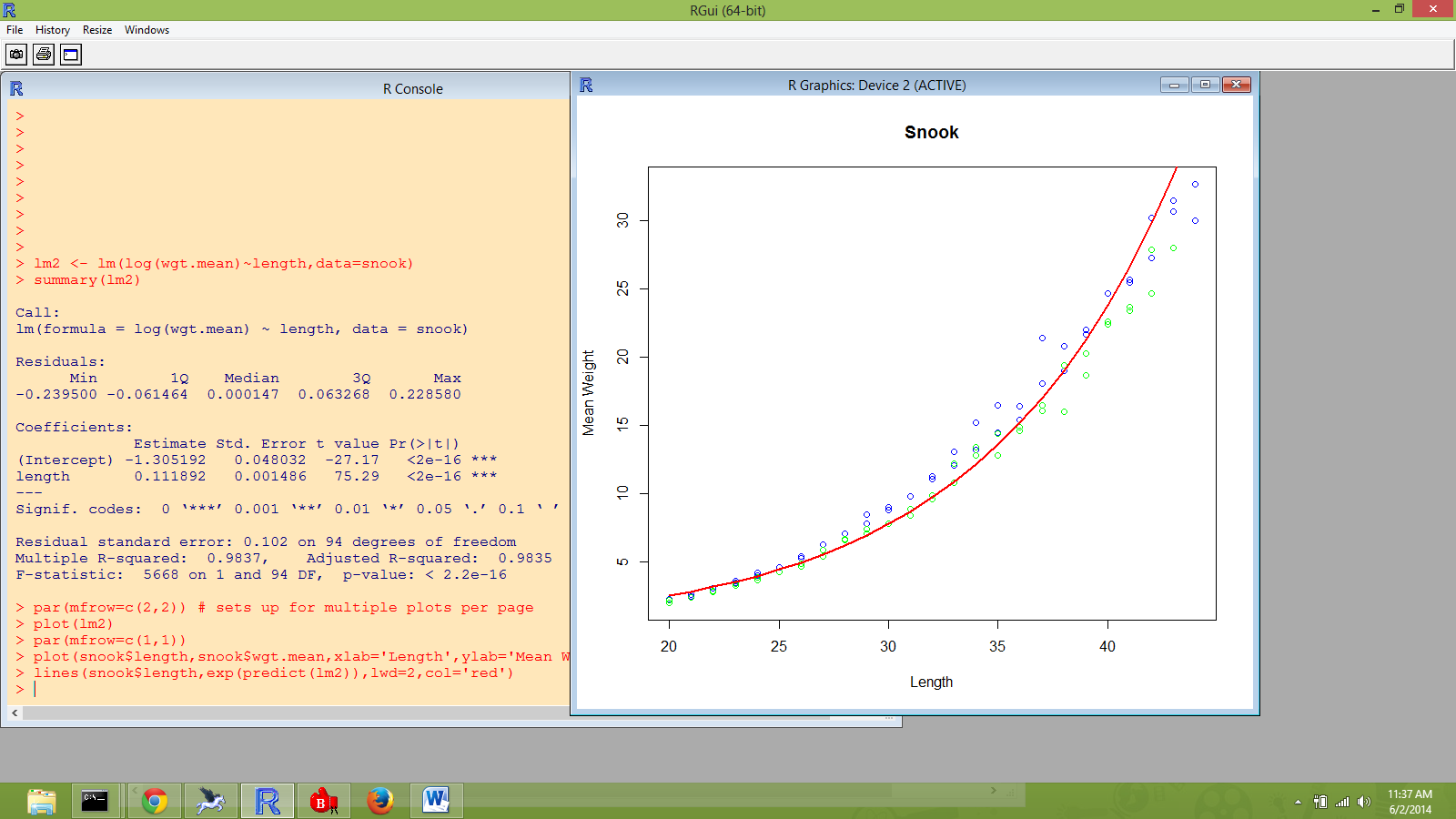
plot(lm2)

par(mfrow=c(1,1))



plot(snook$length,snook$wgt.mean,xlab='Length',ylab='Mean Weight',main='Snook',col=snook$wbcol)

lines(snook$length,exp(predict(lm2)),lwd=2,col='red')



The fit of the log-linear regression is clearly much improved. Alternatively you might want to try a loess regression

Loess Regression

# loess regression

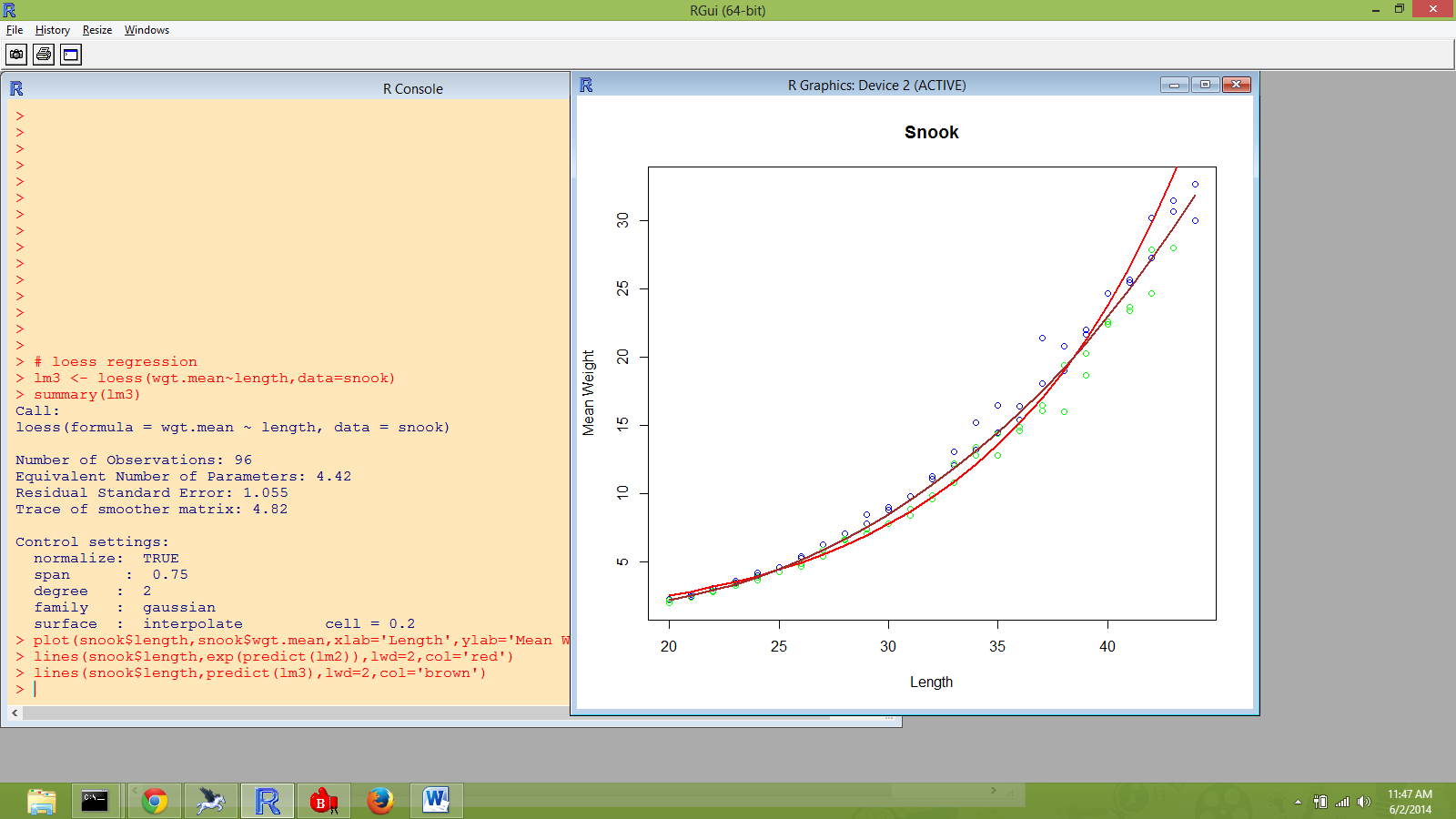
lm3 <- loess(wgt.mean~length,data=snook)

summary(lm3)

plot(snook$length,snook$wgt.mean,xlab='Length',ylab='Mean Weight',main='Snook',col=snook$wbcol)

lines(snook$length,exp(predict(lm2)),lwd=2,col='red')

lines(snook$length,predict(lm3),lwd=2,col='brown')



In this figure, the red line shows the fit of the log-linear model (lm2) and the brown line shows the fit of the loess regression model (lm3).

The remainder of this document will be completed later.

Multiple Linear Regression

cc

Analysis of Variance

cc

Logistic Regression

cc

Mixed Models and Time Series

These analyses will be demonstrated on the Snook data. In the interest of keeping it simple, some of the examples will illustrate the function but not have a meaningful result.