Workshop part 1.0

Intro to Linear Modeling in R

Getting Set Up:

1. Make sure R and R Studio are installed (see instructions). When opened it should look like this:

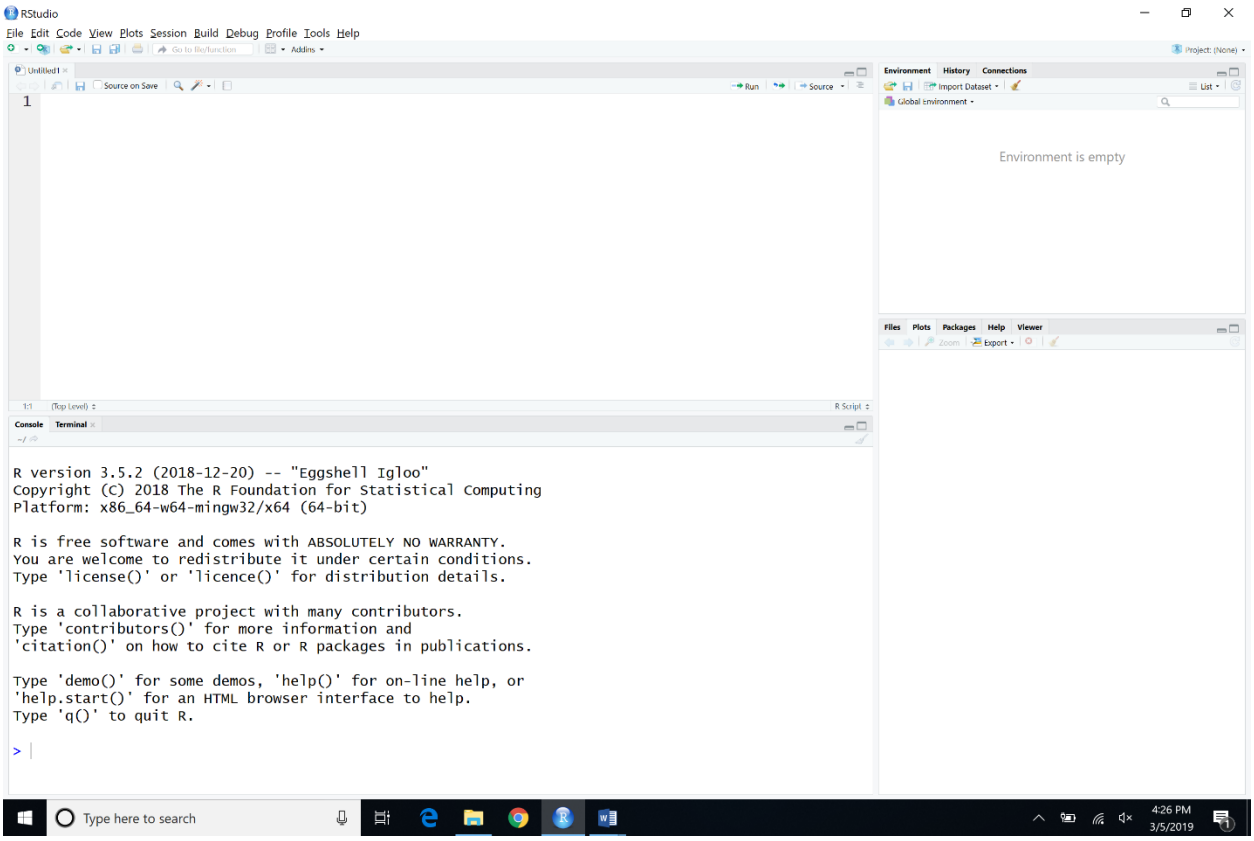

Figure 1. Opening set up for RStudio. Note that menus at the top and along all the windows. Explore them during free time, to see what your options are.

2. Set the working directory to the directory where the scripts and data files are (see Figure 2 below):

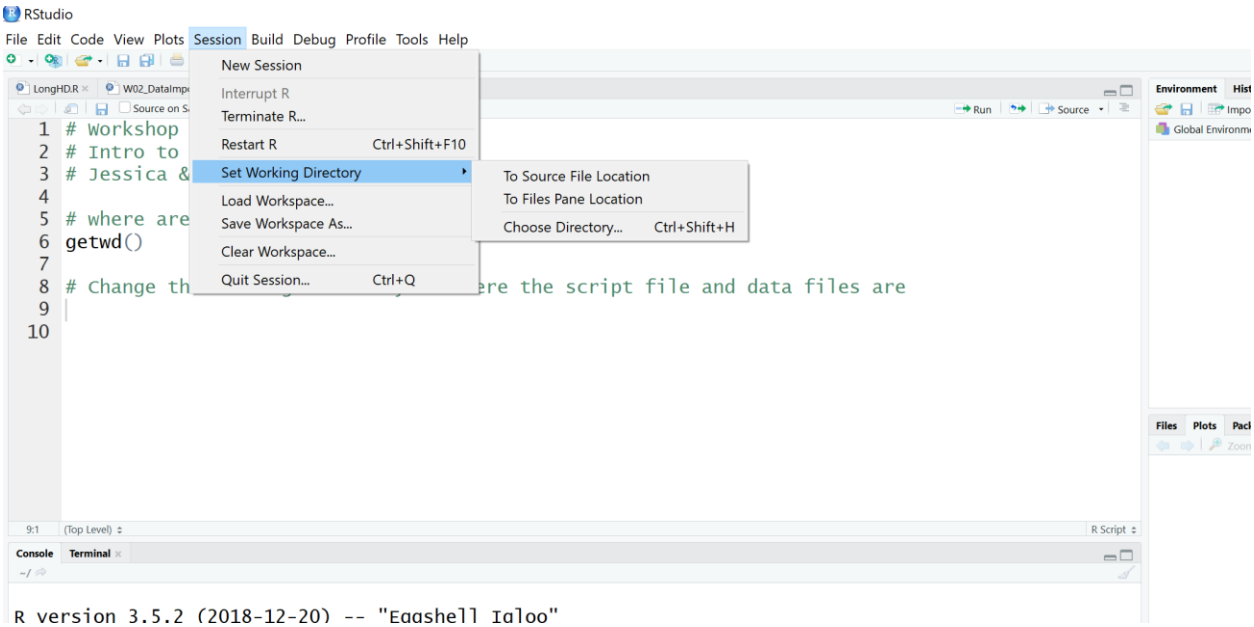

Figure 2. The GUI approach to setting your working directory—you can choose the directory directly ("Choose Directory…"), or pick it to match the location of the script file or the film pane.

3. Install the needed packages and libraries.

First example: A quick regression

We are working largely from Fox and Weisberg, An R Companion to Applied Regression, 2nd Edition (2011, Sage Publications). More recent editions are available, but these ideas are plenty for now. The "car" library comes from the title of the book.

Table 1.1 of Chapter 1 lists many of the functions available from the car package. (Have to get a legitimate copy!) We will focus on lm() today.

Table 1.1 Some R functions for basic statistical methods. All functions are in the standard R packages; chapter references are to the *R Companion*.

Method	R *Function(s)*	*Reference*
Basic Graphs		
histogram	`hist()`	Chapter 3
stem-and-leaf display	`stem()`	Chapter 3
boxplot	`boxplot()`	Chapter 3
scatterplot	`plot()`	Chapter 3
time-series plot	`ts.plot()`	
Numerical Summaries		
mean	`mean()`	
median	`median()`	
quantiles	`quantile()`	
extremes	`range()`	
variance	`var()`	
standard deviation	`sd()`	
covariance matrix	`var()`, `cov()`	
correlations	`cor()`	
Probability		
normal density, distribution, quantiles, and random numbers	`dnorm()`, `pnorm()`, `qnorm()`, `rnorm()`	Chapter 3
t density, distribution, quantiles, and random numbers	`dt()`, `pt()`, `qt()`, `rt()`	Chapter 3
chi-square density, distribution, quantiles, and random numbers	`dchisq()`, `pchisq()`, `qchisq()`, `rchisq()`	Chapter 3
F density, distribution, quantiles, and random numbers	`df()`, `pf()`, `qf()`, `rf()`	Chapter 3
binomial probabilities, distribution, quantiles, and random numbers	`dbinom()`, `pbinom()`, `qbinom()`, `rbinom()`	Chapter 3
generating random samples	`sample()`, `rnorm()`, etc.	
Basic Linear Models		
simple regression	`lm()`	Chapter 4
multiple regression	`lm()`	Chapter 4
analysis of variance	`aov()`, `lm()`, `anova()`	Chapter 4
Contingency Tables		
contingency tables	`xtabs()`, `table()`	Chapter 6
printing tables	`ftable()`	Chapter 6
percentage tables	`prop.table()`	Chapter 6
Simple Hypothesis Tests		
t-tests for means	`t.test()`	
tests for proportions	`prop.test()`, `binom.test()`	
chi-square test for independence	`chisq.test()`	Chapter 6
various nonparametric tests	`friedman.test()`, `kruskal.test()`, `wilcox.test()`, etc.	

Do not copy, post, or distribute

Table 1.1 from the R Companion to Applied Regression (used without permission), showing many of the functions available for analysis.

Quick intro to the lm() function

See the powerpoint regarding the basics of linear modeling in R.

If your dataframe D has variables x as the dependent variable and y1 and y2 as the regressors, and the model is x ~ y1 + y2, then to set up the command to fit the model, you need to decide an output object (let's call it D.mod, for modeling the D data). Then the command is

```
D.mod = lm(D$x ~ D$y1 + D$y2)
```

The D.mod object will now be in the R environment too, and you can work with it to extract results, make plots, etc.

Step 1. Read in the data. We will be using the file Duncan.txt, which includes some basic information from a 1961 study of the prestige of various occupations. This file is organized in columns, including the occupation name, type, median income, a ranking of education level required for the occupation, and the perceive prestige of the occupation.

We use the command read.table().

```
duncan=read.table("Data/Duncan.txt", header=TRUE)
```

Check the global environment window: What changed?

The variable "duncan" is now a data frame in R's memory that contains the contents of the Duncan.txt file. You can look at the dataframe, change it, analyze it, etc., and even write it back out to the disk with a new file name, if you want to. Reading the data into R is somewhat equivalent to opening a file in Excel or SPSS to work with it—except by reading in the data and working with the dataframe, you don't change the original file. Duncan.txt is not affected by anything you do in R to the dataframe, unless you deliberately write the data back out to a new file called Duncan.txt, and overwrite the original file.

Step 2. Check the data. One of the most powerful functions in R is the summary() command. It can be applied to all sorts of objects or variables and will do its best to return something meaningful about the object. In this case,

```
summary(duncan)
```

will apply the summary() command to the duncan dataframe, and return a 5 number summary of each of the variables or columns in the dataframe. This is a very useful way to make sure that your data are reading in as numbers or text as you expected, the range of values is as you expect, and that you're aware of any missing data, etc.

You can also look at just one of the variables in the dataframe, using the $ character:

```
summary(duncan$education)
```

Provides just a summary of the education data.

How would you look at just the prestige data?

What about making a histogram of one of the variables:

```
hist(duncan$income)
```

Or make a scatterplot of income (x) against prestige (y):

```
plot(duncan$income, duncan$prestige)
```

Step 3. Estimate the model. We can start by looking at prestige regressed against income:

```
duncan.mod = lm(duncan$prestige ~ duncan$income)
```

You'll notice that nothing much happens—though over in the Environment window you'll see duncan.mod now exists. What command do we use to look at objects generally?

```
summary(duncan.mod)
```

What are the results?

Step 4. Do the usual regression checks. You can get a histogram of the studentized residuals from the model using the hist() command and the rstudent() command on the duncan.mod object:

```
hist(rstudent(duncan.mod)
```

And a qqplot:

```
qqPlot(duncan.mod)
```

There are many many more options at that point! The ones in the car package can be found in the official pdf for the package: https://cran.r-project.org/web/packages/car/car.pdf

Quick sidebar on scatterplot: Scatterplot() from the car package is well worth playing around with! It will do quick regressions and make pretty pictures, but doesn't save the regressions.

Step 4. Apply what you've learned to ask a new question: How would you create an object called duncan.mod2 which models prestige as the result of **both** income and education levels?