Data Analysis Through Modeling:Thinking and Writing in Context Technology Guide: Using Excel 2016 and R

Kris Green and Anne Keller Geraci

Fall 2017 Edition[1]

(this page left blank intentionally)

Contents

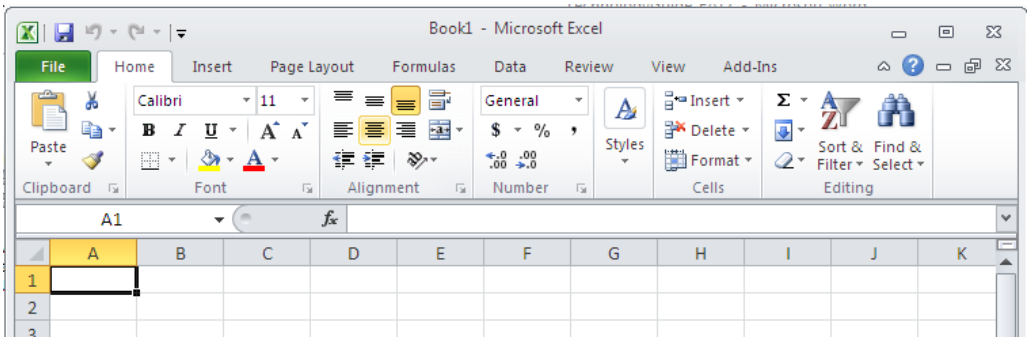
Chapter 1. Format of computer information in this guide

This technology guide is provided as a supplement to *Data Analysis Through Modeling: Thinking and Writing in Context*, by Kris Green & Allen Emerson.

This textbook contains three kinds of computer information to help you. Each will be formatted a little differently, so here is a brief overview of each to help you.

1. Basic information on using computers. One of the most important things you will need to learn is how to use computers efficiently. Where will you store your data files? How will you retreive your half-completed assignment if you are not on campus? How will you organize your files so that you can find them again?

2. Using Microsoft Excel 2016. Chapters 3 through 10 contain information on using Excel efficiently to do various modeling activities. Since Excel is a visual environment, will present much of this information through screenshots like this:



Your screen might look slightly different from this if you are using a different version of Excel, however the procedures outlined here can be performed in ANY version of Excel. Remember that you can usually find out how to something with a quick Google or Youtube search such as "How to create a pivot table".

3. Using R and Rstudio. Chapter 11 is entirely on how to use the statistical software R and the Interactive Development Environment (IDE) Rstudio. Since R is a command line language, you will see R statements like this, which will need to entered into either an R script (a stored program) or the R console. :

```
require(mosaic)
rm(list=ls())
```

You are encouraged to download and install R and Rstudio to your personal computer so that you are able to work on assignments outside of the computer lab environment.

Chapter 2. Basic Computer Information

2.1 Advice on computers and doing work electronically

There is nothing so tragic as bad things happening to good students.
Unknown Instructor

If you want to avoid being one of those good students to whom bad things happen, take heed of the following advice. It should become a mantra, repeated to yourself over and over until it is a part of your psyche:

SAVE EARLY, SAVE OFTEN.

Anytime you make a substantial change to your work like pasting a graphic in, typing a whole sentence or paragraph, adding a table, or reformatting, you should save your file. Save as soon as possible after starting a file. There is also a keyboard shortcut for saving files: CTRL+S. Use this frequently to avoid losing a substantial part of your work.

2.2 A Note About Naming Files and File extensions

When you save your files (”Early and often”, remember) be sure to save them with a meaningful name. If the file includes your solution to homework 2, then include ”homework 2” in the title. You may also want to save all the files for each course you are taking into a separate folder, named for the course. Finally, if the file is going to be sent electronically to your instructor (through email or some course management system) it’s a good idea to make sure that your name appears on the file in some way. After all, unless you are the only student in the class, the file name ”homework 2” could belong to anyone. Your instructor may even establish guidelines for naming files in order to make file management for the entire course easier on him/herself and the teaching assistants (if any). Be sure to check whether your instructor has a preferred file-naming system

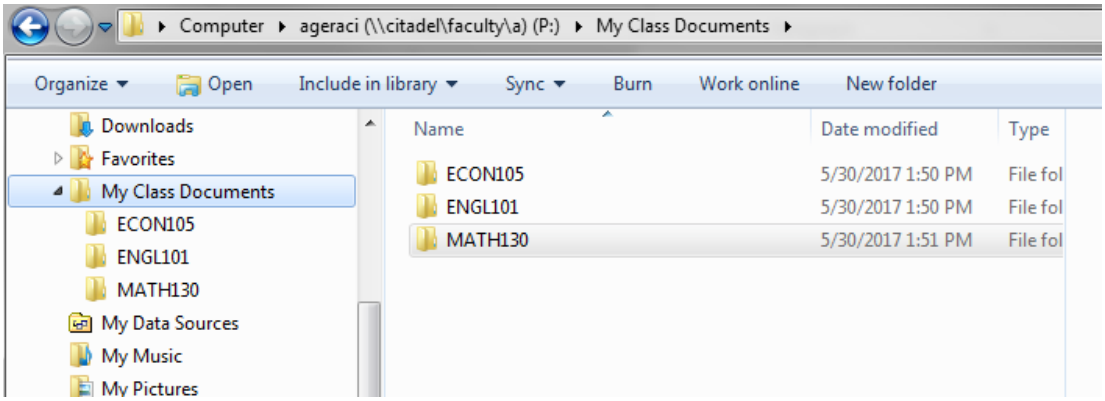
It is also helpful when saving files to name them meaningfully. If you name the first Excel workbook for every course you take ”File1” you will have a lot of files with the same name. Come up with a naming convention that clearly helps you locate the files you want.

Files with this extension	…. are typically used with this softare
.DOC or .DOCX	Microsoft Word
.XLS or .XLSX	Microsoft Excel
.R or .Rmd	R Studio. (.Rmd is for R Markdown)
.CSV	Comma-Separated File (data)

2.3 Folders and Organization

When saving your files, it also helps to have some sort of plan for organizing the files. In Windows, the way to do this is to use **folders**. These can be named anything you want, and you can have as many folders inside a folder as you want. You can also put folders inside other folders. Just be careful: it's easy to create such a complex nest of storage folders that you cannot remember where your files are.

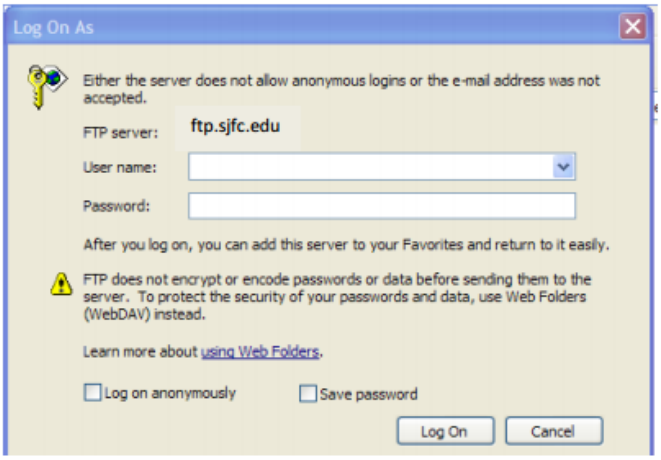
Option 1 – P:\\ Network Drive (limited to 250MB)



One way of storing your files is on your P: drive. If you need to access your P: drive files from somewhere off-campus (or from your own computer), follow these instructions:

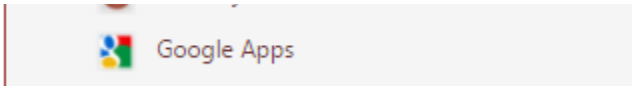
- Open **My Computer** >in the address bar enter: ftp://ftp.sjfc.edu and press **enter**.

- The **Log On As** dialog box will pop up asking for your **username** and **password.**
- Your username is: academia/ your **SJFC username**
- Your **password** is the same as logging into the computers on campus, and click **Log On**.
- Your files should appear and look exactly like your P:drive on campus.

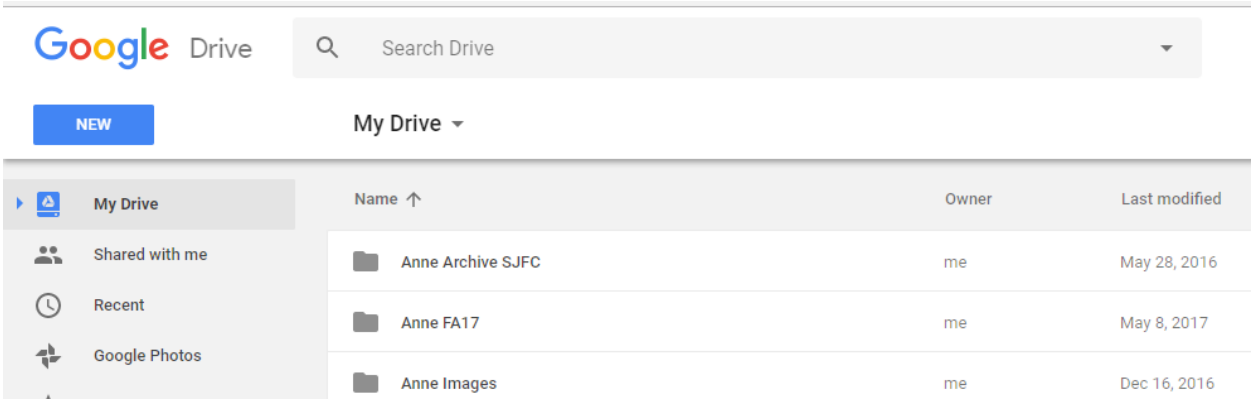


Option 2 – Google Drive (unlimited storage space)

The other way of storing your files is on Google Drive. These files can be accessed using this icon on the SJFC launchpad:



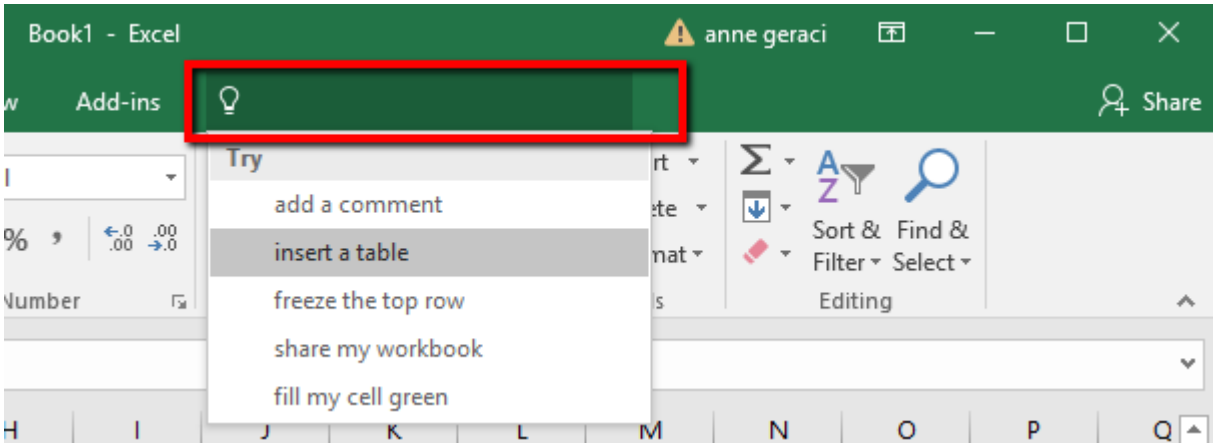
For example, here's how you might organize your Google Drive:



The benefit of this method is that you can download and install a program called Drive (from https://tools.google.com/dlpage/drive), which will automatically synchronize your Google Drive files to your personal computer.

2.4 Using the help system in Microsoft Office 2016

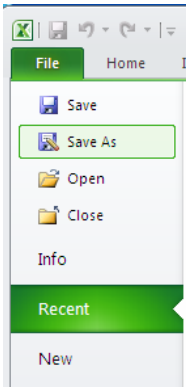
The help system for Microsoft Office 2016 is fairly extensive. In Word or Excel (or Power Point), the help function is found near the top of the screen with the words "*Tell me what you want to do"*.



You may need to get used to using the help features. Very often, your first try will not turn up anything, but always check the "see also" line that appears with most help information. This will link you to other information that is related to the topic you originally searched for.

Within the information portion of the help window, most of the phrases and sentences are hyperlinked to allow you to navigate through the information to locate what you need.

For example, using the "Save As..." feature allows you to change the format of the file. To do this, use the pull-down menu below the file name to select a different file type.

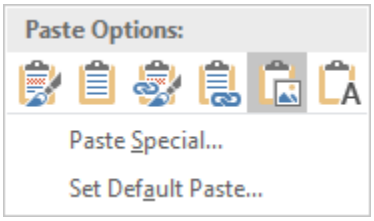


To open a file, you can either double-click the file icon in the browser or the Windows explorer, or you can open the file from within Excel. Simply open the file menu and choose "Open"; then browse through the folders on the computer to locate the file you want.

2.5 Copying and pasting between programs

Microsoft Office 2016 is designed so that you can select information in one program, copy it (using either the keyboard shortcut CTRL + C or the menu command "Edit/ Copy") and then paste it into another program. When you copy selections, they are placed in an area called the "clip board". To take these selections from the clipboard and place them into a document (either another location in the same document, or in another document altogether) simply place the cursor where you want the information to go and either use the keyboard shortcut "CTRL + V" or the menu "Edit/ Paste" to paste the object in the location you have selected.

When copying information from Excel to Word, consider using the "Paste Special – Bitmap" option, shown here, to paste a bitmap of your work into a document. This will ensure that your information (in Word) looks exactly like it does in Excel – this may prevent excessive "clean-up" re-formatting of your information.

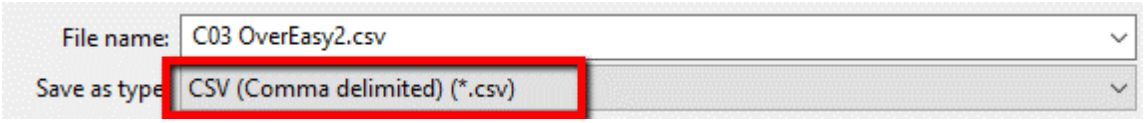


2.6 Saving your Excel data as .CSV (Comma-separated values)

In order to import your data into other software such as R or SAS, it is recommended that you SAVE your data in a format known as Comma-separated Values, or CSV. For example, the following file has five variables (indicated by their names in the first line) and each data item is separated from the next by a comma

```
ID,Day,Time,Location,Count
1,M,500,Nook,6
2,M,530,Nook,15
3,M,600,Nook,44
4,M,630,Nook,26
```

To save your data in the CSV format, use the "File>Save As" menu option and indicate CSV for the filetype:



NOTE: If you have mutiple Worksheets in your Excel file, uou will need to save each sorksheet to an individual CSV file.

Chapter 3. Using Excel

In this chapter, we will present some information about using Excel with a brief set up of how a worksheet should be organized in order to perform proper data anlaysis and conversion to other software packages such as R or SAS.

One of the benefits of using Excel to do computations is the use of formulas. Any formulas in the format below show the syntax of formulas to be typed into Excel. Formulas might refer to other cells within the worksheet, as in this example where we are adding cells B3 and C3, then dividing by the contents of cell D3:

```
=(B3+C3)/D3
```

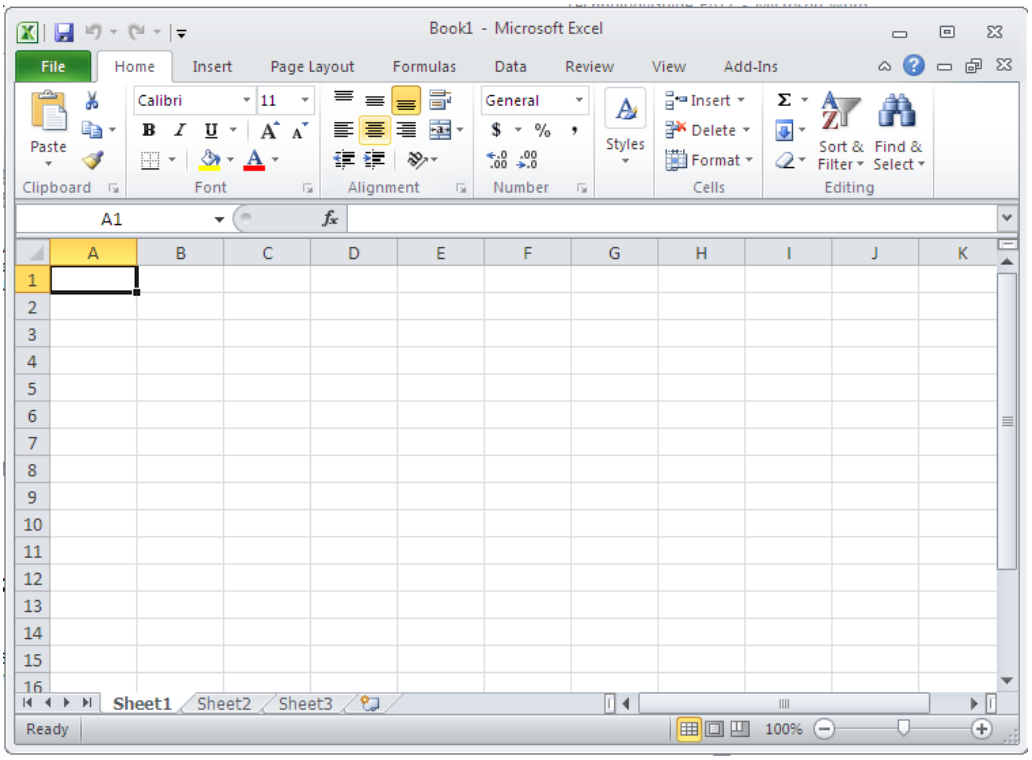
… or they might make use of built-in Excel functions, as in this example which will return the sum of the values found in cells B3, C3, D3, E3, and F3, or B3:F3 for short.

```
    =SUM(B3:F3)
```

Any formulas shown in this format, with specific cells or cell ranges in the formula, should be typed exactly as shown, assuming that your spreadsheet is set up as described in the information or as shown in the accompanying screenshots and images.

3.1 Excel desktop

This lesson will introduce you to the Excel desktop. To begin this lesson, start Microsoft Excel. The Microsoft Excel window appears and your screen will look similar to the one shown here.



Starting at the top of the Excel desktop, you will see the following items:

- The program control bar: This displays the name of the current workbook and provides buttons for minimizing, maximizing, and closing the program on the right-hand side. The left-hand side has a quick button for saving the workbook (the disk icon), undo-ing and redo-ing actions (the looping arrows), and a pull-down menu to customize the interface. The extreme left of the program control bar has a Microsoft Office icon; clicking this accesses features that let you save, open, or print workbooks.

- Menu tabs: These tabs, labeled ”Home”, ”Insert”, ”Page Layout”, ”Formulas”, ”Data”, ”Review”, ”View”, and ”Add-ins”, control the ribbon below. Unless you have installed an Add-in (like StatPro) you may not see the last of these options.

- Ribbon: The ribbon (formerly the toolbar) has icons for most of the actions you might need to perform in Excel. Selecting different menu tabs changes the icons on the ribbon to the ones associated with that tab. Usually, these are grouped together. Hovering the cursor over any of the icons provides additional information about the tool.

- Formula Bar and Name Box: The formula bar has two regions. The left area (the Name Box) allows you to select, view, or name cell ranges in the current worksheet. The right area (the Formula Bar) displays the formula entered into the current active cell, or allows you to type in a formula.

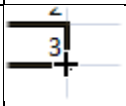
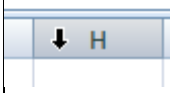
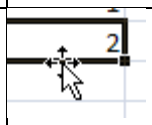
- Workspace: The main area of the screen is a grid of cells into which you enter information, data, and formulas. Each of these cells has a name, identified by first the column (A, B, C, etc.) and then the row (1, 2, 3, etc.) So cell D6 is in the fourth column (labeled D) and the sixth row.

- Worksheet Control: This area, just below the workspace, has tabs to select different worksheets in the workbook.

- Status Bar: The status bar provides quick statistics for the region of data that is currently selected in the worksheet along the right side. Along the left side is where you will see error messages and notifications.

3.2 Excel cursor shapes

The pointer on the screen can take any of nine different shapes. The shape of the pointer is a clue to what actions will take place when you click the left mouse button.

Shape of pointer		Action when you left click
Normal arrow		Selects the current item
Fat plus sign		Selects the cell (either for entering data/formulas or for other purposes, like copying)
Skinny plus sign		Click and drag to copy the formula(s) or the pattern in the selected cell(s) to other cells on the worksheet
I-beam		Enter text
Short down arrow		Selects entire columns of data
Short right arrow		Selects entire rows of data
Double arrow (one line)		Click and drag to change cell widths (left-right arrows) or heights (up-down arrows)
Four-headed arrow		Click and drag to move cells or toolbars around

3.3 Excel Errors

Under certain circumstances, even the best formulas can appear to have freaked out once you get them in your worksheet. You can tell right away that a formula's gone haywire because instead of the nice calculated value you expected to see in the cell, you get a strange, incomprehensible message in all uppercase letters beginning with the number sign (#) and ending with an exclamation point (!) or, in one case, a question mark (?). This weirdness is known, in the parlance of spreadsheets, as an error value. Its purpose is to let you know that some element - either in the formula itself or in a cell referred to by the formula - is preventing Excel from returning the anticipated calculated value.

Here is a list of some error values and their meanings:

#DIV/0! Appears when the formula calls for division by a cell that either contains the value 0 or, as is more often the case, is empty. Division by zero is a no-no according to our mathematical rules (you can divide a pizza into 2 slices, but you cannot divide a pizza into zero slices).

#NAME? Appears when the formula refers to a range name that doesn't exist in the worksheet. This error value appears when you type the wrong range name or fail to enclose in quotation marks some text used in the formula, causing Excel to think that the text refers to a range name.

#NULL! Appears most often when you insert a space (where you should have used a comma) to separate cell references used as arguments for functions.

#NUM! Appears when Excel encounters a problem with a number in the formula, such as the wrong type of argument in an Excel function or a calculation that produces a number too large or too small to be represented in the worksheet.

#REF! Appears when Excel encounters an invalid cell reference, such as when you delete a cell referred to in a formula or paste cells over the cells referred to in a formula.

#VALUE! Appears when you use the wrong type of argument or operator in a function, or when you call for a mathematical operation that refers to cells that contain text entries.

3.4 Good Data Entry Practice

Organize your spreadsheets so that the data is stored with the variables in columns and observations are stored in rows. Make sure that each variable has a heading at the top of the column of data to identify it. It's a good idea to add comments to each variable name in order to explain the coding and the units of the data. Make sure each observation has a unique identifier.

It is also very important that each cell in the data contain information from only one variable. For example, if you are coding information about homes and you want to record data on the garage, you have two things to deal with: whether the garage is attached to the house or not, and the number of cars that the garage can hold. You would not want to have the cells coded as "Detached 2" and "Attached 1" and so forth. That is mixing two variables, type of garage and size of garage, into a single variable. It would be better to either

- create two variables, one for "Type," coded as "attached" or "detached" and a separate variable for number of cars, as shown here:

H	I
Garage	Attached
Bays	Detached
0	None
2	Detached
0	None
1	Attached
1	Attached
0	None
1	Detached

- code a single variable (nominal categorical) to include the information, perhaps using the codes below

```
1 = attached, 1 car garage
2 = attached, 2 car garage
3 = detached, 1 car garage
4 = detached, 2 car garage
5 = other type of garage
6 = no garage
```

3.5 Comments in Excel

Excel allows you to add notes, called ”comments” to any cell. These comments are not part of the data or formulas in the cell, and they do not normally appear in the worksheet. Instead, any cell with an attached comment will have a small red triangle in the upper right corner. If you place the mouse pointer over a commented cell, the comment will appear. Comments are used to include such information as the way in which a variable is coded, the units of numerical data, and references to the source of the data.

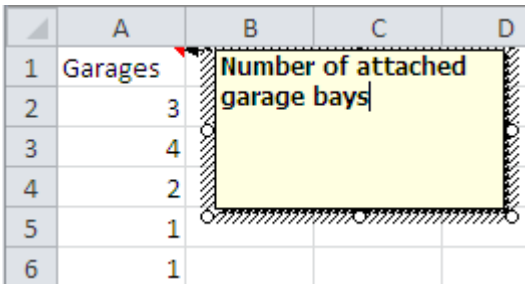
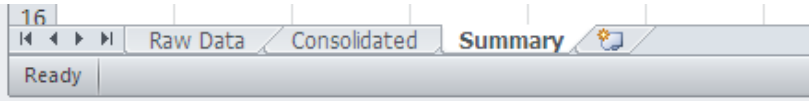


Figure 1: Example of a Cell comment

To add a comment to a cell, right click on the cell. In about the middle of the context menu, the option ”Add comment...” should appear. Select this option, and an editable comment box will appear. Type your comment in the box. When you are done, select another cell with the mouse. Your comment will be entered into the spreadsheet. To make changes to an existing comment, right click on the commented cell and select ”Edit comment...” To delete a comment from a cell, right click on the cell and select ”Delete comment...”

3.6 Managing Worksheets in Excel

An Excel workbook (e.g. MYDATA.XLSX) can contain many different worksheets. By default, these are named "Sheet1", "Sheet2", etc. Worksheets are accessed by the tabs at the bottom of the screen, as shown here:

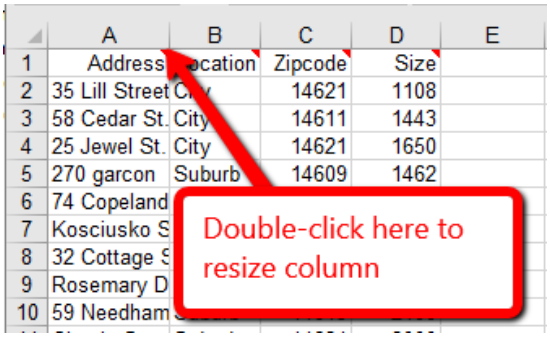


To re-name a sheet: double-click on the name of the sheet and it the sheet name will be highlighted. Type in the new name for the sheet and hit ENTER. (You can also right click on the sheet name and select "Rename" from the menu.)

To add another worksheet to your workbook you may either use the "Insert" menu, or right-click on the worksheet tabs and select "Add worksheet". You can also click on the worksheet tab to the right of the last worksheet in the workbook. To change the order of the worksheets, click and drag one of the tabs to a new place in the list; you will see a small sheet icon and a down arrow showing you where the sheet will be placed. It is also helpful to rename the worksheets with more meaningful names than "Sheet1" and "Sheet2". To do this, either

3.7 Sizing columns to make data fit

You may also run into the problem that information you enter into cells in a spreadsheet might not fit. You have two options to get information to fit: You can either resize the columns or you can enter the text on multiple lines. To resize the columns, you can go to the column header and either clip-and-drag the width of the column to the desired size or you can double-click on it so that it automatically resizes to be wide enough for the widest entry in that column.

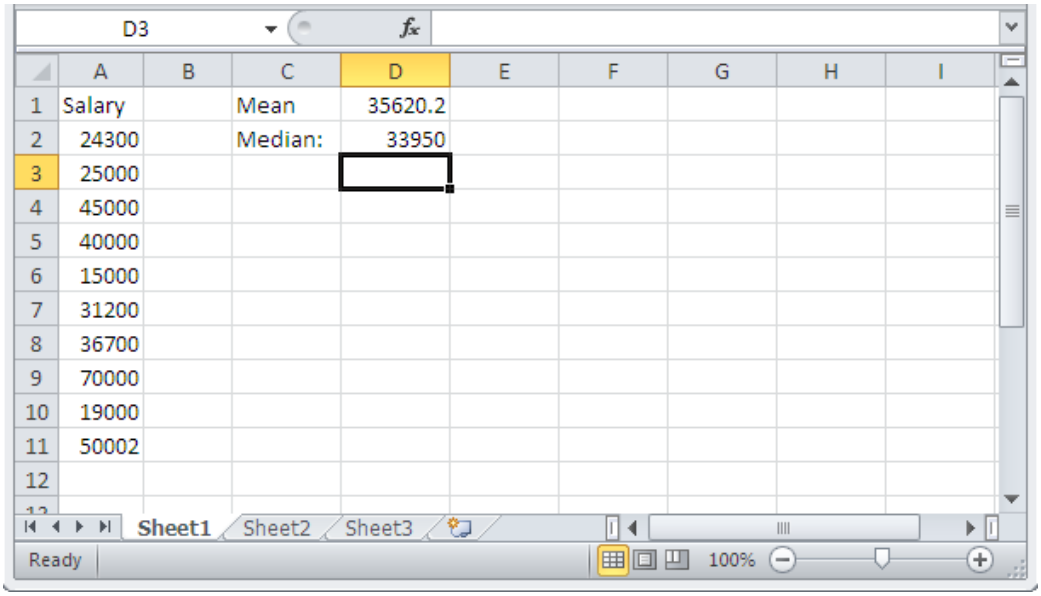


	A	B	C	D	E
1	Address	Location	Zipcode	Size	
2	35 Lill Street	City	14621	1108	
3	58 Cedar St.	City	14611	1443	
4	25 Jewel St.	City	14621	1650	
5	270 garcon	Suburb	14609	1462	
6	74 Copeland Street	Suburb	14609	1198	
7	Kosciusko Street	City	14621	1577	
8	32 Cottage ST	City	14608	1110	
9	Rosemary Drive	City	14621	1737	
10	59 Needham	Suburb	14615	2100	

To enter information in multiple lines within a single cell in Excel, first type the information on the first line and then hit ALT+ENTER to move to the second line. You can use as many lines as you want.

3.8 Good spreadsheet organization

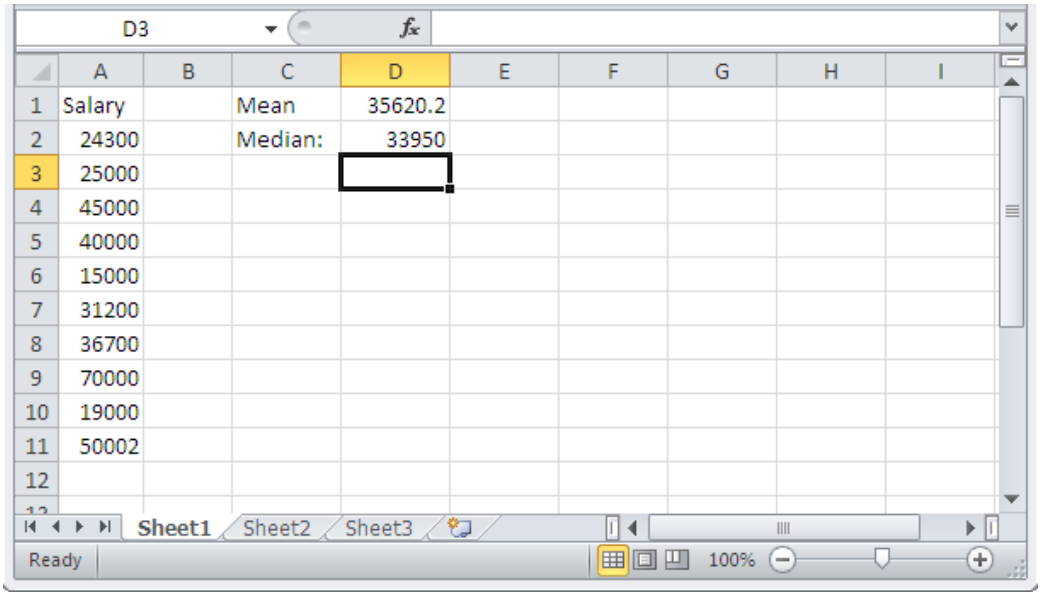
When doing summary calculations in Excel, it is recommended that you leave a space between the raw data and the calculations. For example, in this worksheet we are computing the mean of the Salary data (in column A). Note that we left column A blank and make our calculations in Column C and D.



It is also acceptable to use a new worksheet for such calculations.

3.9 Cell References in Excel

Excel organizes information into sheets. Each worksheet is then organized by columns (labeled by letters) and rows (labeled by numbers). Thus, every cell (rectangle on the worksheet that contains information) has a name, called a cell reference. This cell reference is usually given the way you called out locations on the game Battleship: as a column and a row. For example, in the worksheet shown at the right, the word ”SALARY” is in cell A1. The mean of the salary data is in cell D1. Such a reference is called a relative cell reference.



There are other types of cell references (called absolute cell references) that you will learn about below. The other important thing to know about cell references is that you can easily refer to a block of cells, as long as the block of cells forms some sort of rectangle. For example, to refer to the salaries themselves on the worksheet above, you would refer to all the cells in the rectangle starting in cell A2 and continuing through cell A11. In Excel notation, this entire range of cells is identified by A2:A11.

Absolute Cell References in Excel

Above, you learned how to refer to any cell or range of cells using the grid system in Excel. If there is data in the cell in column D in row 2, this cell is referred to as D2. However, this type of cell reference (a relative reference) will change if the formula is copied to another cell. Many times (as in the example below of computing deviations) a particular cell reference will need to be absolute. This means that it will not change if the formula is copied. To make a cell reference absolute, place a dollar sign ($) in front of both the column and row. Thus, an absolute reference to cell D2 would look like $D$2.

As you may have guessed, you can have mixed references also, where either the column or row is absolute. In general, if you don't want part of the reference (either the row or column) to change as you copy the formula, be sure to place a dollar sign in front of it.

When you are typing a cell reference into a formula, you do not have to type the dollar signs to convert them to absolute references. After you type a cell reference in a formula (say you type A2), hit the F4 button along the top row of the keyboard. This converts the current cell reference into an absolute reference (so now you would have $A$2). If you hit the F4 button again, it is converted to a mixed reference with the row fixed (A$2), hitting it again will convert it to a mixed reference with the column fixed ($A2). Finally, hitting F4 a fourth time will cycle back to a relative reference (A2).

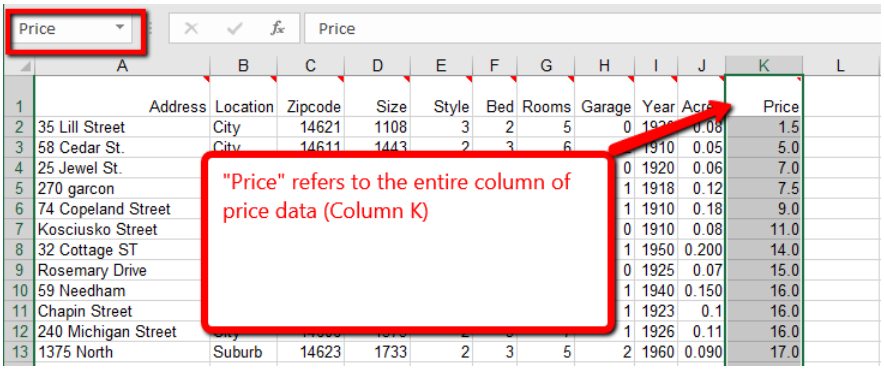
Three dimensional cell references in Excel

In addition to referring to cells by the column and row, Excel allows you to build formulas that include references to cells on other worksheets in the current workbook. Suppose you are entering a formula in 'Sheet 1' of a workbook and there is a number in cell D4 of 'Sheet 2' that you want the formula to look up. Simply typing D4 in the current formula will not work; Excel will simply look up the value in cell D4 of the workbook containing the formula. To get around this, you must use a 3D cell reference. All this involves is including the name of the worksheet in single quotes, followed by the "bang" or exclamation mark symbol (!) and then the normal cell reference. So, in our example, to get a formula in 'Sheet 1' to use the value in cell D4 from 'Sheet 2', you need to type the cell reference exactly in the form

```
'Sheet 2'!D4
```

Naming Cell Ranges

There is another way to refer to cell ranges (or individual cells) besides a cell reference. You can give the cells or cell ranges their own names and then use these names in formulas for computations. The figure below shows the C05 Homes.xls data with the data in the Price variable (column M) selected. To give this range of cells a name, we simply click on the ”Name Box” to the left of the formula bar and type a name; in this case, we’ll call the range of cells ”Price”. Note: there are no spaces allowed in the name box.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Address	Location	Zipcode	Size	Style	Bed	Rooms	Garage	Year	Acr	Price	
2	35 Lill Street	City	14621	1108	3	2	5	0	193	0.08	1.5	
3	58 Cedar St.	City	14611	1443	2	3	6		1910	0.05	5.0	
4	25 Jewel St.							0	1920	0.06	7.0	
5	270 garcon							1	1918	0.12	7.5	
6	74 Copeland Street							1	1910	0.18	9.0	
7	Kosciusko Street							0	1910	0.08	11.0	
8	32 Cottage ST							1	1950	0.200	14.0	
9	Rosemary Drive							0	1925	0.07	15.0	
10	59 Needham							1	1940	0.150	16.0	
11	Chapin Street							1	1923	0.1	16.0	
12	240 Michigan Street							1	1926	0.11	16.0	
13	1375 North	Suburb	14623	1733	2	3	5	2	1960	0.090	17.0	

Now, in any formula in the worksheet we can use ”Price” instead of the range M4:M278. Instead of typing the formula =AVERAGE(M4:M278), you could just type the formula =AVERAGE(Price). Notice that such references are always absolute.

This has the benefit of making all the formulas more readable. You can see a list of all the named ranges in the workbook by clicking on the downward pointing triangle next to the name box.

3.10 Copying Formulas in Excel

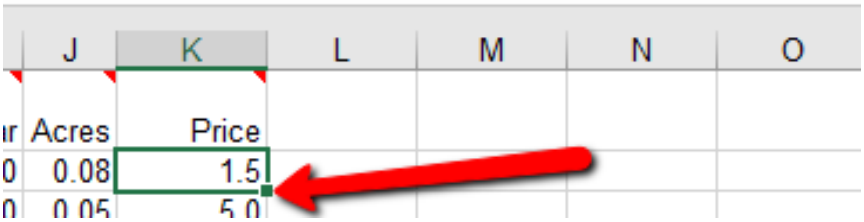
There are three different ways to copy formulas in Excel from one cell to another cell or to a group of cells (like a whole column): standard copy and paste commands, dragging the fill handle, or double-clicking the fill handle.

Using Copy and Paste Commands

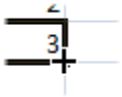
This method is the most obvious. First select the cell with the formula you want to copy. Copy this using either CTRL+C, the copy button on the toolbar, or the ”Edit/Copy” menu command. Now highlight the cell or cells where you want the formula to be placed and paste it in using either CTRL+V, the paste button on the toolbar, or the ”Edit/Paste” menu command.

Dragging the Fill Handle

If you want to copy the formula to the column of cells beneath it, or to the row of cells beside it, you can use the fill handle.



The fill handle is a tiny square that appears in the lower right corner of a cell you have selected. If you click on this fill handle and drag down (or right), then, when you release the mouse button, the formula from the first cell (or group of cells!) is copied to all the cells in the area you highlighted by dragging. Be sure that you are clicking on the fill handle, though. You’ll know for certain that you are on the fill handle because the cursor will change from a fat plus sign to a skinny plus sign.



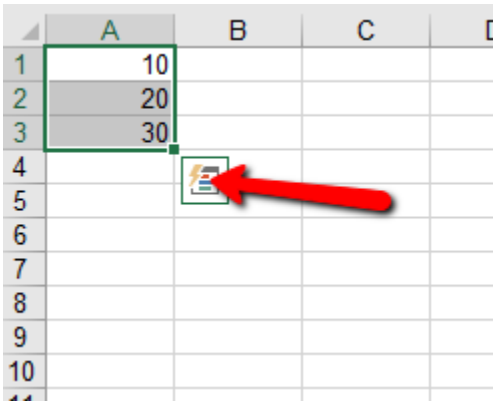
Double-clicking the fill handle

In certain circumstances, you can double-click the fill handle and Excel will automatically copy and paste the formula all the way down the column until it reaches the end of the column to the left of the one in which you are pasting the formula.

How the Fill Handle Works to Complete a sequence of numbers

In Excel, you may have used the fill handle to copy a formula down a column or across a row. Remember, the fill handle is the little dot in the lower right corner of the active cell or active cell region. The fill handle can also be used to fill in patterns in a sequence of numbers that you enter.

For example, suppose you want to generate a column of numbers 10, 20, 30, 40, on up to 300. It would be tedious to type these by hand. Excel can help! Start by typing 10 in cell A1, 20 in cell A2 and 30 in cell A3. Now highlight the cells (A1:A3).



Click and drag the fill handle all the way down the column until the little floating box that follows the cursor says ”300”. Release the mouse button and your list of numbers is filled in!

Chapter 4 Computing Summary Statistics in Excel

4.1 Computing mean & standard deviation in Excel

Excel uses the function AVERAGE for the mean. To compute the mean of the data in cells A2:A11, we enter the formula

```
=average(A2:A11)
```

into any cell on the spreadsheet. If you later move or copy the cell containing this formula, the cell references will be changed since we used relative cell references. This means that the formula will probably not point to the right cells anymore. Also remember that if you change any of the data in cells A2:A11 the mean will be re-calculated instantly. If, however, you add data outside this range, you will need to change the formula.

There are two different standard deviation functions to use in Excel, depending on whether the data is from a sample or a population.

To compute the standard deviation of a sample (this is the most commonly used version), use the formula

```
=stdev(range of cells)
```

For the standard deviation of a population, use the formula

```
=stdevp(range of cells)
```

4.3 Adding up a list of values

If you have a list of values, you can quickly add them together using the SUM command in Excel. For example, if your values to be added are in cells A2:A26, entering the command

```
=SUM(A2:A26)
```

into cell B2 (or any other cell) will add the values together.

4.4 Computing deviations in Excel

In order to compute the deviations in Excel, we first need the mean of all the data. Let's calculate this with Excel by typing =average(A2:A20) into cell F1.

Now, we will create a new column for the deviations. In cell B1, type "Deviation" so that the column has a label. Now, in cell B2, we want to enter a formula to compute difference between the first data point (in cell A2) and the average (an absolute reference to cell F1). Thus, we enter the formula

```
=A2 - $F$1
```

Now we simply copy this formula (see below) down to the other cells, using the "Fill Handle" procedure described above.

4.6 Computing Medians & Modes in Excel

To compute the median of the data in cells A2:A11, we enter the formula

= MEDIAN(A2:A11)

into any cell on the spreadsheet. Remember, though, that if you later move or copy the cell, the cell references will be changed since we used relative cell references. Also remember that if you change any of the data in cells A2:A11 the median will be re-calculated instantly. If, however, you add data outside this range, you will need to change the formula.

The mode is computed with the formula

=MODE(A2:A11)

You may get the result #N/A if there is no mode. If there is more than one mode, Excel just guesses and gives one of them. The fact that there may be more than one mode, or no mode at all, is why this statistic is rarely used except for categorical data.

4.8 Computing z-scores

To compute z-scores for the variable Price (cells M3:M278), we first need to compute the mean and standard deviation. In cell P1 enter =AVERAGE(M3:M278) to compute the average and in cell P2 enter =STDEV(M3:M278) to get the standard deviation. Now, in column N, enter "Z Score" in N3 and enter the formula below into N4

```
=(M4 - $P$1)/$P$2
```

All that is left is to copy the formula to the rest of column N (N5:N278).

4.9 Computing Percentiles in Excel

To calculate percentiles in Excel, use the formula

```
=PERCENTILE(array of cells, percentile)
```

Note that percentile should be entered as a decimal number. Thus, for the 80% percentile, you should enter 0.80. For the 35th percentile, enter 0.35.

Chapter 5. Making and Using Pivot Tables

5.1 What is a pivot table?

A pivot table is a data summarization tool used by Excel to present *numerical* information broken down by *catagorical* variables. For example, we might compute the average overall price for our Homes data as a value of 119.8 (representing $119,800). Wouldn't it be interesting to "slice and dice" that data using the Style of the house (shown here in columns) and Location (rows).

Average of Price	Style					
Location	1	2	3	4	5	Grand Total
City	44.67	63.67	28.17	100.00	440.00	71.71
Rural	107.15	178.17	134.86	123.33	149.76	151.76
Suburb	87.10	129.94	79.91	98.32	85.25	107.34
Grand Total	93.02	136.97	86.11	108.18	163.19	119.77

Note that we still have the Grand Total average of 119.77 in the lower right corner, but now we can compare the average price for a City house with the average price for a Rural house.

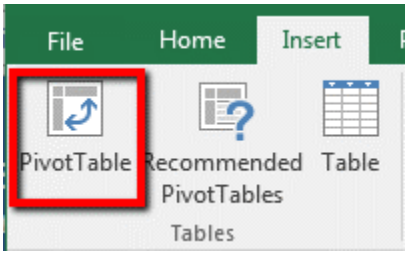
5.2 Creating a pivot table

Let's create a pivot table using the Enpact Data:

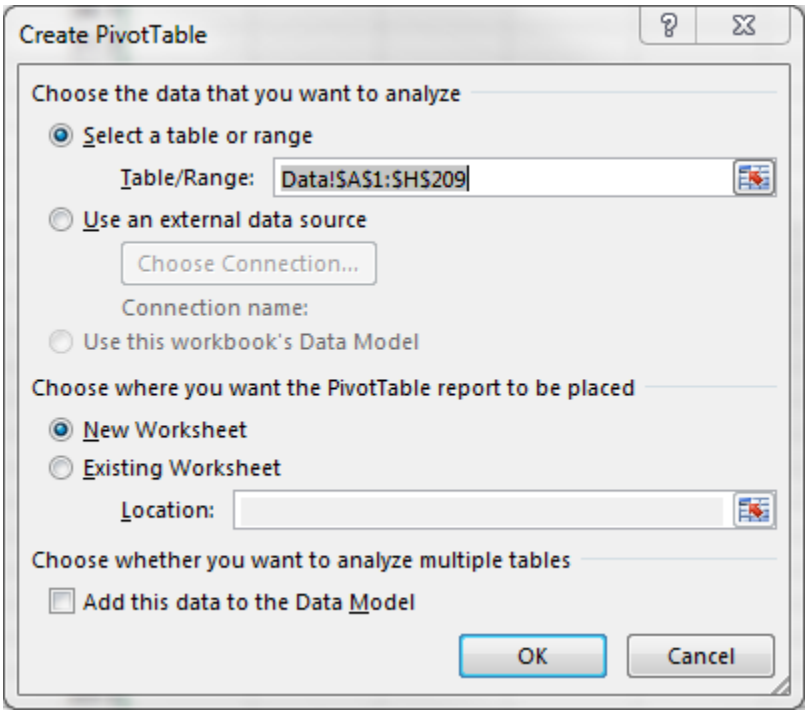
	A	B	C	D	E	F	G	H
1	Employee	EducLev	JobGrade	Age	YrsExp	Gender	YrsPrior	Salary
2	1	3	1	26	3	Male	1	35.4
3	2	1	1	38	14	Female	1	41.6
4	3	1	1	35	12	Female	0	35.8
5	4	2	1	40	8	Female	7	34.1
6	5	3	1	28	3	Male	0	31.9
7	6	3	1	24	3	Female	0	33.1
8	7	3	1	27	4	Female	0	32.8
9	8	3	1	33	8	Male	2	29.7

1. Start by selecting a cell inside the range of the data. For example, you might select cell A1 (or any other cell in the range of data).

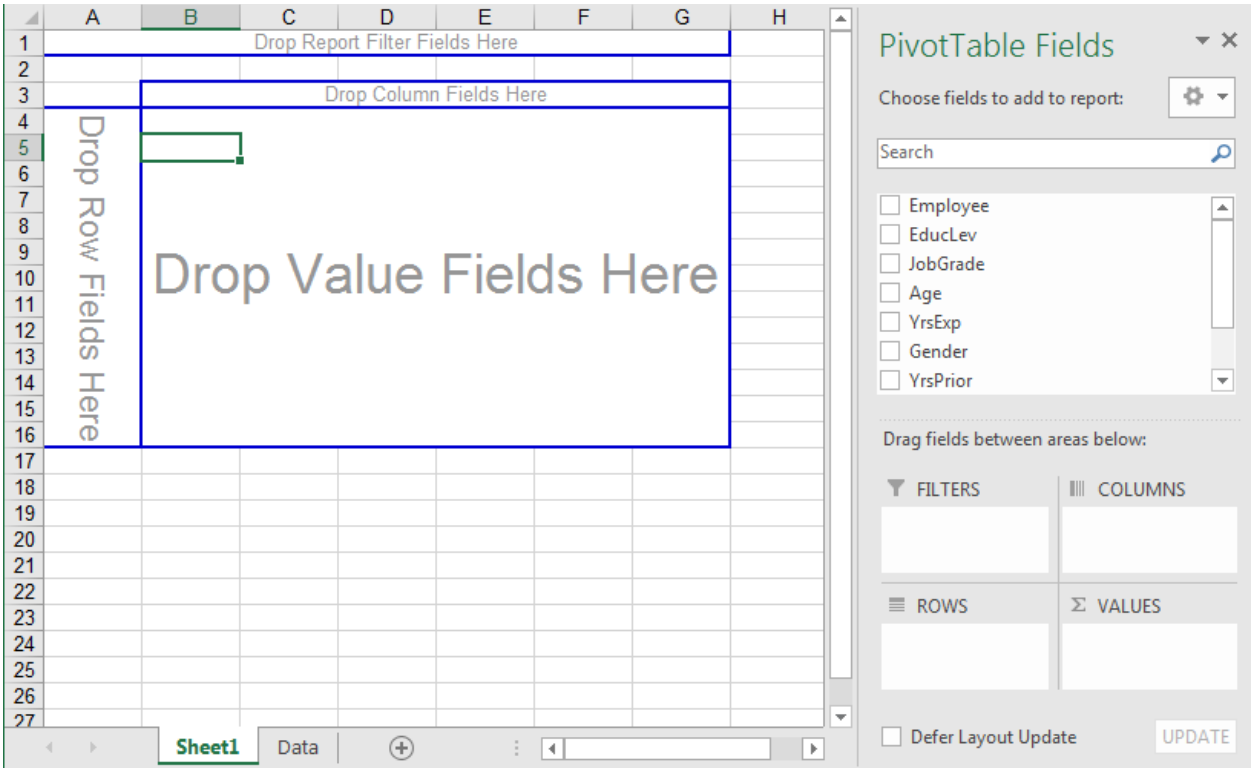
2. Go to the *Insert* menu and select *Pivot Table*.



3. Normally, you will not need to change anything here. Verify that the data range is correct, make sure it has "New worksheet" selected for where to create the table. Click OK.



4. What you see on this are (a) the pivot table area (on the left) and (b) the pivot table field list (along the right-hand side). The PivotTable names should match the variable names/headings in the spreadsheet.



While you can place any variable in any field, and in particular you can use either row or column variables interchangeably, if your variable has many values (for example, ”Years Experience” in the C03 EnPact.xls file) you are best off making it a row variable, rather than a column variable in order to make it easier to read the resulting table, and avoid having to scroll horizontally to get information.

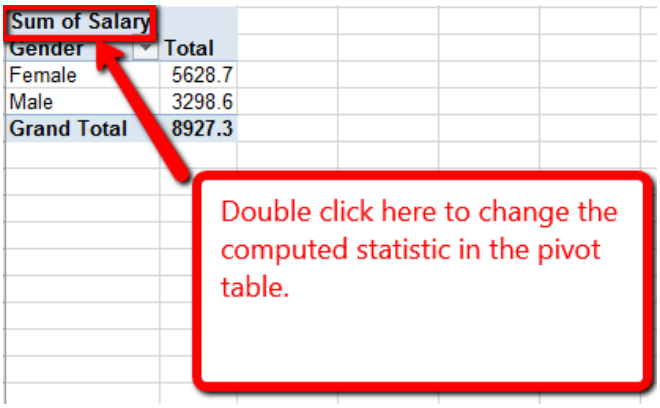
Starting from the screen shown above, you now simply drag fields from the field list into the various regions of the pivot table. No data will be displayed until there is one field in the area marked ”Drop Data Items Here”.

For example, to look at the average salaries of the employees, broken down by gender, complete the following.

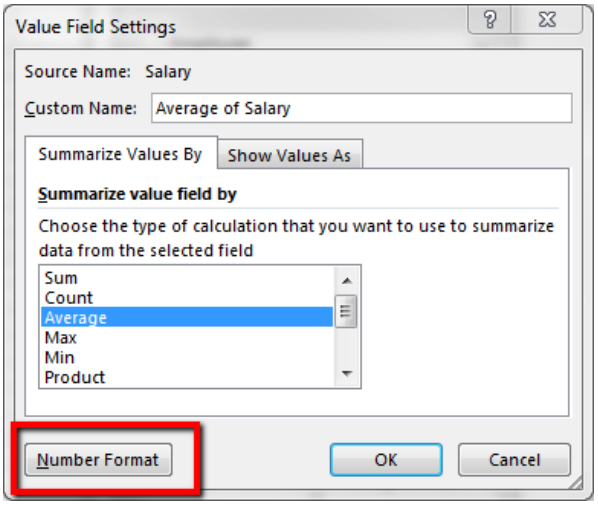
1. Drag *Gender* to the area marked *Drop Row Fields Here* or drag it into the area in the lower right marked *row labels*.

2. Drag *Salary* to the area marked *Drop Data Items Here* or drag it into the area in the lower right marked *values*.

3. By default, the pivot table will either show the sum of the data variable (in this case, the total of all salaries for males and females) or the count of the data variable (the number of males and females). We would rather see the averages. To display the averages, double-click where it says *Sum of Salary*.



4. To summarize the data by averages, select *Average* from the list on the left.



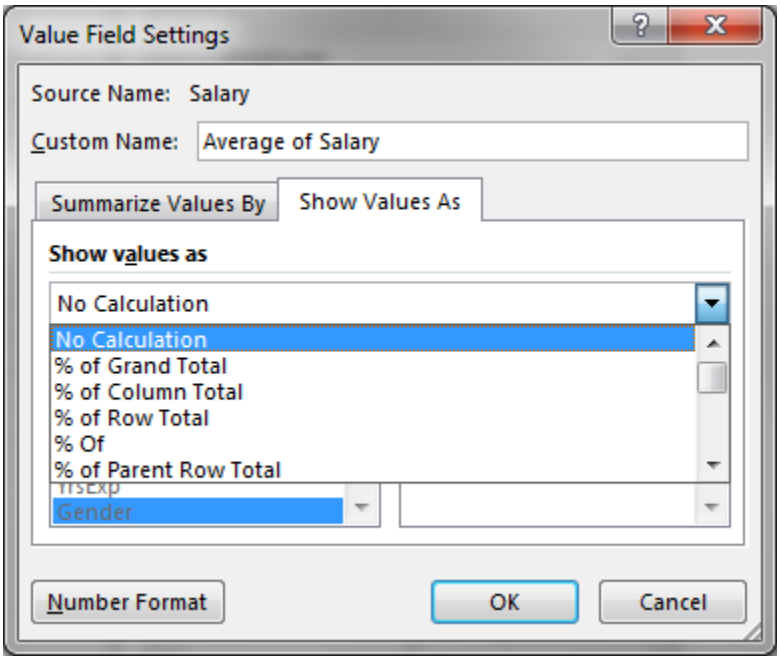
Use the "Number Format" button at the bottom to specify an Excel format such as $ or %.

5. The following pivot table will be displayed.

Average of Salary		
Gender	**Total**	
Female	$	40.21
Male	$	48.51
Grand Total	**$**	**42.92**

5.3 Pivot Table – Show values as

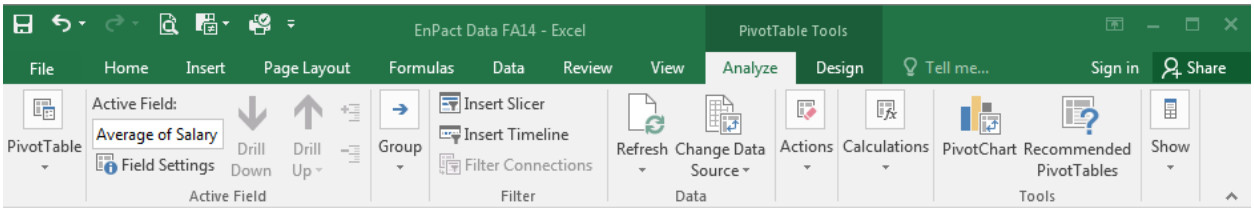
For more sophisticated pivot tables, you might display the data as percentages, etc. To access this feature, simply click the tab marked *Show values as* in the pivot table field display dialog box.



To select a different way of presenting the data, select one of the options from the pull down menu under *Show values as*. This gives you the various options for displaying the data. The most useful items from the list are probably *Normal*, *% of row* and *% of column*.

5.4 The Pivot Table Ribbon

The pivot table ribbon, shown below, provides a lot of flexibility for working with the pivot table.



Two of the most important items on the ribbon are the *Refresh* button and the *Change Data Source* button.

Refresh forces Excel to re-check the original data and re-build the current pivot table. This is useful if you change or add data to the original database. This makes it easy to update information, without having to create the pivot table again. If you have more data - that is, data outside the original range of the pivot table - you can use the *Change Data Source* button to modify the data range.

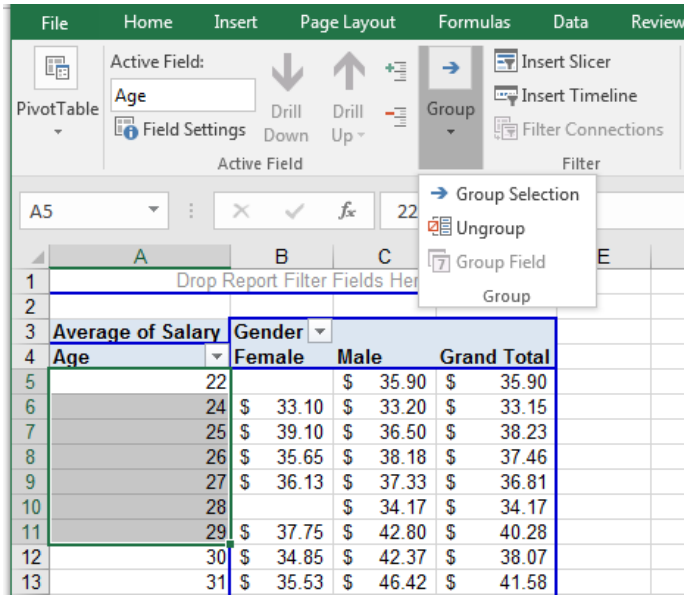
5.5 Grouping items in the table

This feature allows you to take a variable that has many values (like a numerical variable) and group it together in the pivot table. For example, one could easily use *Age* as a row variable in the pivot table above, but the wide variety of values makes it hard to see any details or compare results. However, if you group some ages together (like 20-29, 30-39, etc.) you can see more interesting results as illustrated below.

Average of Salary	Gender		
Age	Female	Male	Grand Total
20-29	$ 36.83	$ 37.87	$ 37.47
30-39	$ 39.79	$ 44.27	$ 40.96
40-49	$ 41.67	$ 52.78	$ 43.66
50-59	$ 40.35	$ 53.31	$ 46.24
60-69	$ 39.30	$ 91.60	$ 58.32
Grand Total	**$ 40.21**	**$ 48.51**	**$ 42.92**

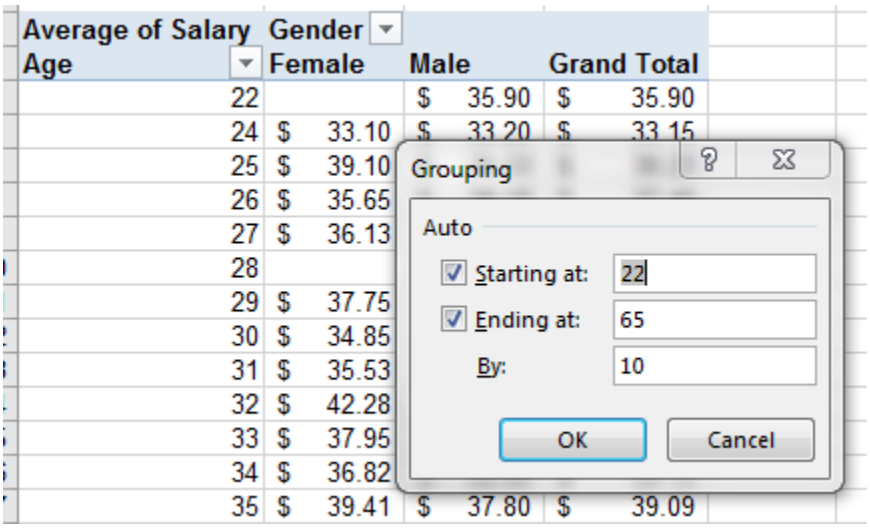
From this table we can observe the average salaries for Females and Males in the various age brackets show.

To group the salary categories in order to see the data more easily, select a range of years experience (like all the rows with va;ies 20 through 29) and click the *Group Selection* button on the ribbon.



Repeat this with the other ages. Then you can collapse or expand the individual groups of experience to look at the data more easily.

You can also right click on the *Age* field in the pivot table. Select *Group...* from the context-sensitive menu that appears. You should see a dialog box like the one below.

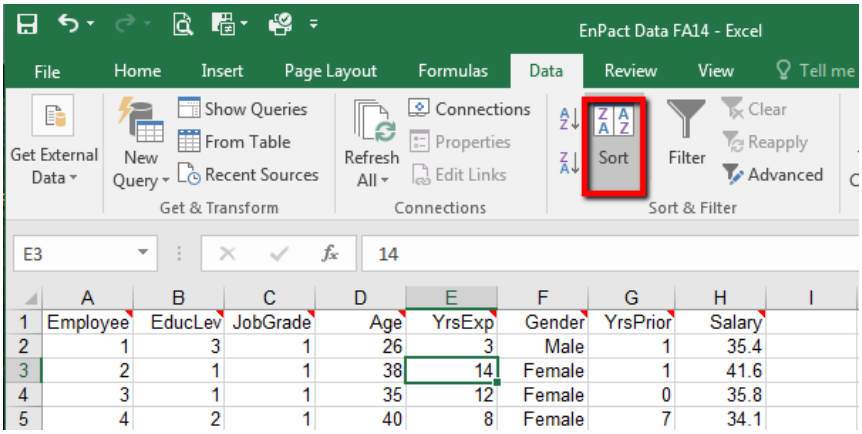


Here you can select the starting value, ending value and space between groupings. The settings below, for example, produce the age groupings shown in the table above.

Chapter 6 Sorting data

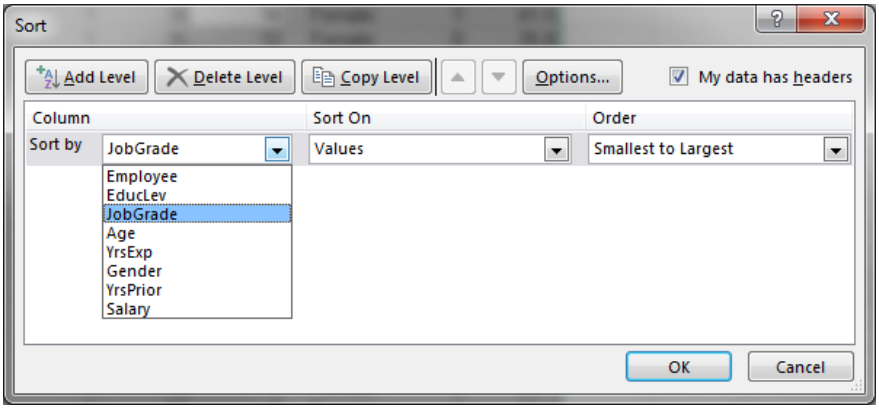
Excel makes it relatively easy to sort your data on many variables simultaneously. In order to use this effectively, though, you need to have your data organized as we have discussed in chapter two: your variables (fields) should be the columns and the observations (records) should be the rows. It is also a lot easier if you make sure the first row of the data contains headers (variables names).

First select (click on) any cell in the data range. Then go to the data ribbon and select *Sort:*



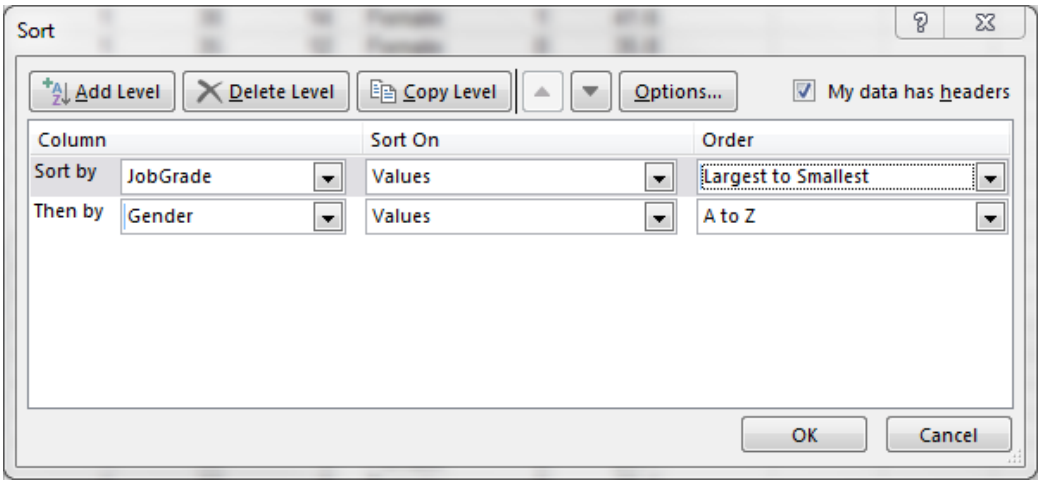
	A	B	C	D	E	F	G	H	I
1	Employee	EducLev	JobGrade	Age	YrsExp	Gender	YrsPrior	Salary	
2	1	3	1	26	3	Male	1	35.4	
3	2	1	1	38	14	Female	1	41.6	
4	3	1	1	35	12	Female	0	35.8	
5	4	2	1	40	8	Female	7	34.1	

Now you may use the dialog box shown below to select the desired sort column and sorting order.



You may sort on several variables by adding more sort conditions using the ”Add Level” button. You can delete conditions or add as many as you like. In the upper right-hand side of the dialog box make sure the ”My data has headers” is checked, so that Excel knows the variable names you are using

If you wanted to sort by job grade and then by gender, you might select ”Job Grade” in the first sort condition (and ”largest to smallest” for the sort order) and then add another level to the sort and select ”Gender” for that level. This will sort the list on two variables. Excel will collect all the employees with JobGrade = 6 at the top of the list, and within that group, the Female employees will be at the top of the list and the Male employees at the bottom. Sorting on three variables is similar.

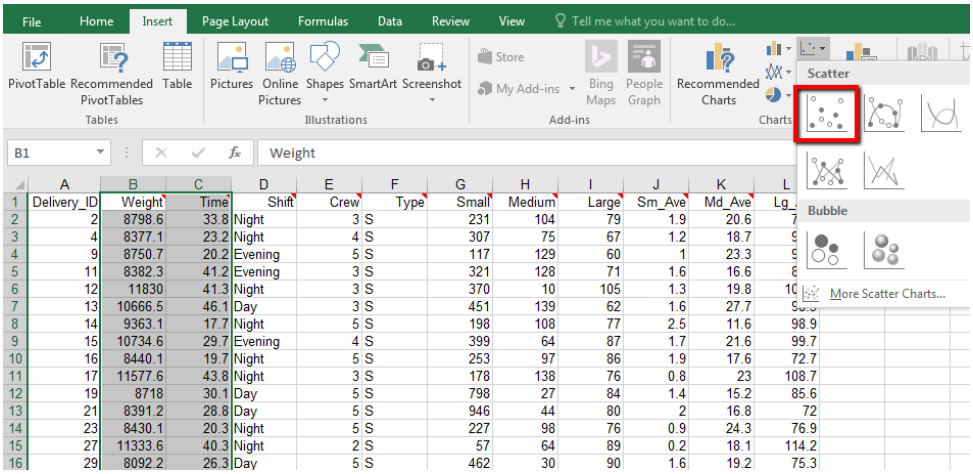


Sorting data can be very useful for identifying outliers in the data or other anomalies. For example, if you have data on the diameter of parts being produced by one of your factory machines, and you determine that the mean size of these is 0.45" with a standard deviation of 0.03", sorting the data on the diameter variable would let you quickly find any parts produced that are too far above or below the standards of your company.

Chapter 7 Making Charts and Graphs with Excel

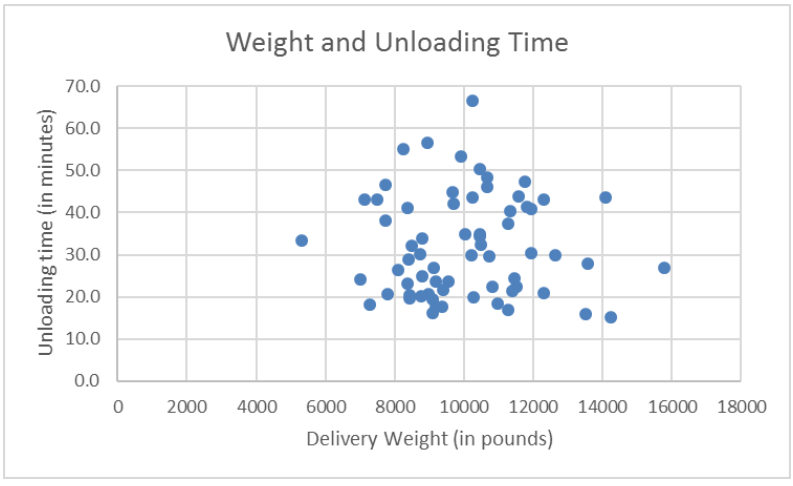
7.1 Scatter Plots

First, select the data you want. For Excel, this means that you must highlight all the data (and the variable names at the tops of the columns) that you want to graph. If the two variables are not right next to each other, highlight the first column of data, then hold down the control key (CTRL) and highlight the second column of data. Click the ”Insert” ribbon and select *scatter* from the list of plot types. Then select the subtype of graph that you want to create. See figure 30.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Delivery_ID	Weight	Time	Shift	Crew	Type	Small	Medium	Large	Sm_Ave	Md_Ave	Lg_
2	2	8798.6	33.8	Night	3	S	231	104	79	1.9	20.6	7
3	4	8377.1	23.2	Night	4	S	307	75	67	1.2	18.7	9
4	9	8750.7	20.2	Evening	5	S	117	129	60	1	23.3	9
5	11	8382.3	41.2	Evening	3	S	321	128	71	1.6	16.6	8
6	12	11830	41.3	Night	3	S	370	10	105	1.3	19.8	10
7	13	10666.5	46.1	Day	3	S	451	139	62	1.6	27.7	9
8	14	9363.1	17.7	Night	5	S	198	108	77	2.5	11.6	98.9
9	15	10734.6	29.7	Evening	4	S	399	64	87	1.7	21.6	99.7
10	16	8440.1	19.7	Night	5	S	253	97	86	1.9	17.6	72.7
11	17	11577.6	43.8	Night	3	S	178	138	76	0.8	23	108.7
12	19	8718	30.1	Day	5	S	798	27	84	1.4	15.2	85.6
13	21	8391.2	28.8	Day	5	S	946	44	80	2	16.8	72
14	23	8430.1	20.3	Night	5	S	227	98	76	0.9	24.3	76.9
15	27	11333.6	40.3	Night	2	S	57	64	89	0.2	18.1	114.2
16	29	8092.2	26.3	Day	5	S	462	30	90	1.6	19.2	75.3

Note that when making scatterplots in Excel, the software will assume that the left column variable is the independent variable and the right-column variable is the dependent variable.



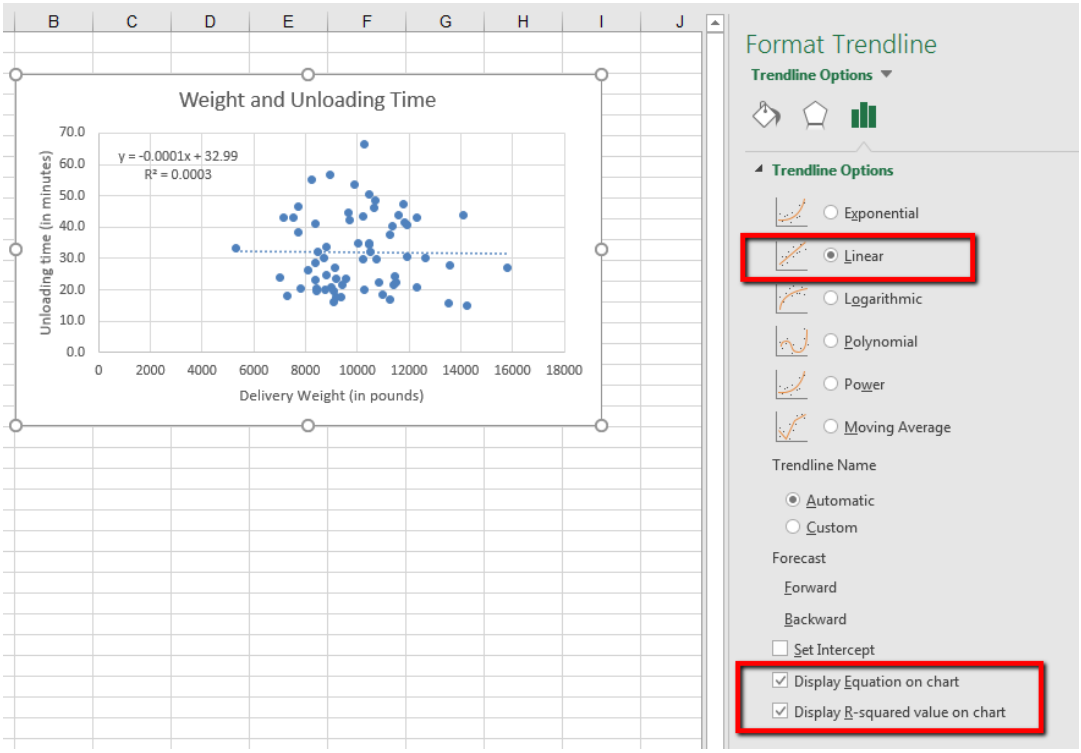
Don’t forget to use the *Chart Elements* function to add axis labels and an appropriate title.

7.2 Adding Trend Lines to a Scatter Plot

Now we will use EXCEL's capabilities to explore the relationship between the two variables by creating a "Trend line".

1. Position your pointer over one of the points on the scatter plot and right-click your mouse. Select "Add Trendline..." from the menu that appears.

2. You will now have a panel on the right (shown below) that shows several different types of functions that EXCEL can graph on top of your data. Let's select "Linear", which is the default choice.



3. Make sure you select "Display Equation on Chart" and "Display R-squared value on chart". This will help us in the future.

4. When you have finished setting the options, click on "Close". You should now see your scatter plot with two new things added. One of these will be a dotted line, the other will be a little text box that displays the equation of the line and the R-squared value. For right now, think of R-squared as a measure of how closely the line resembles the data. The closer this number is to "1", the better the line describes the data. We can also get R^2 values and equations for the other types of trend lines that EXCEL will graph.

5. Try out some other trend lines with this graph. EXCEL can put as many onto the same graph as you want. Simply repeat all the steps above for making a trend line, except choose a different function in step 2 each time.

A note about the other shapes for trendlines: In later chapters, we'll explore the other types of trendlines and what they are good for. For right now, just be aware that straight lines aren't the only option. Also, as we'll find out, some trendlines simply can't be used with certain data. If this is the case, Excel will automatically "grey out" those choices from the list.

A note about the Polynomial choice for trend lines: Polynomials come in different degrees. You can control the degree of the polynomial that Excel uses by adjusting the number in the box next to the polynomial trendline. Excel allows degree 2 through 6 polynomials.

7.3 Adding Trendlines for Non-proportional Models

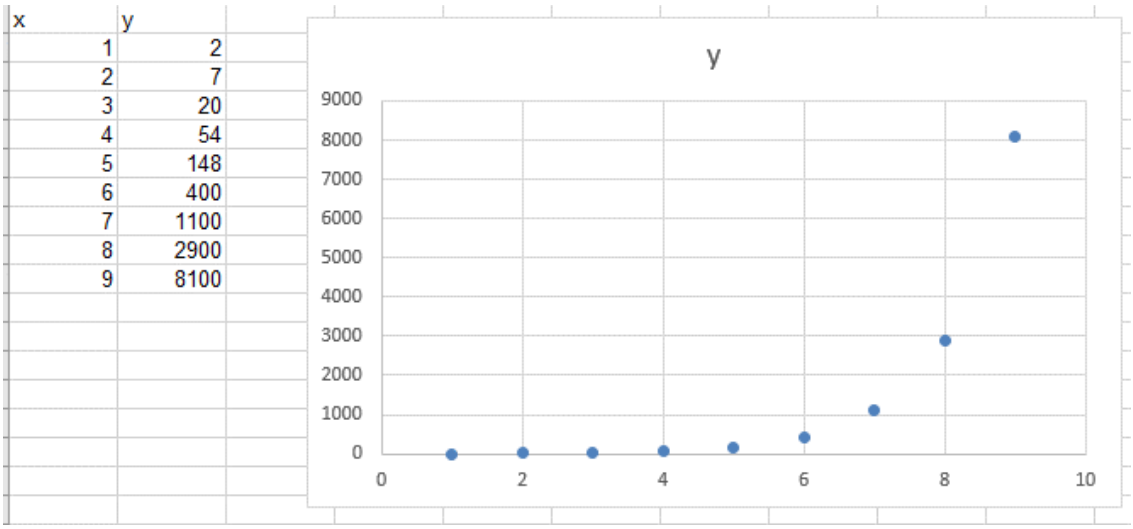
Excel can add trendlines for some non-proportional models to graphs. The process is virtually identical to the process used above to add linear trendlines in Excel. The only difference is that in step 2 you should select the following options:

- To get an exponential fit, choose "exponential"
- To get a logarithmic fit, choose a "logarithmic"
- To get the square fit, use "Polynomial" and select "Order 2"
- It is not possible to force Excel to generate trendlines for reciprocals or square roots directly. As it turns out, these are specific cases of the more general "Power" models. However, if you add a "power" trendline to a graph, the power is one of the parameters in the model (like slope or y-intercept) so you probably will not get a power of 0.5 ($=\frac{1}{2}$ which is a square root model) or a power of -1 (for a reciprocal model).

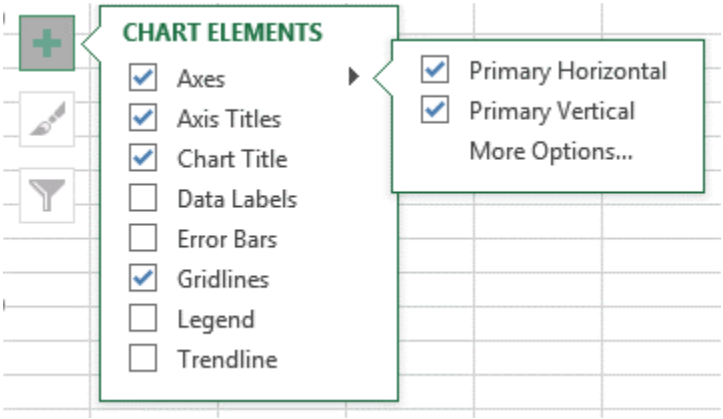
7.4 Logarithmic and Log-Log plots

When you have data that spans many order of magnitude (like 1, 10, 100, 1000, 10000...) taking the logarithm of the data reduces it to a much more manageable set of numbers. For example, if we take the base-10 logarithm of each number in the preceding list, we get the numbers (0, 1, 2, 3, 4, 5...) which are must easier to use. This is the essence of many commonly used scales of measurement (the Richter scale for measuring earthquake energy and the unit of measuring sound, the decibel, are both logarithmic). This is also useful in dealing with models in which the variability in the residuals increases.

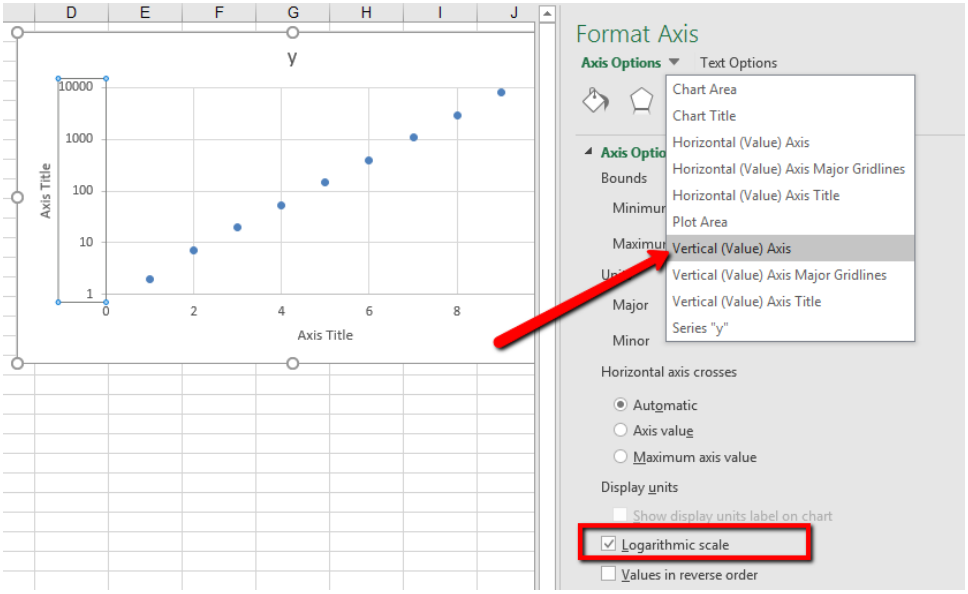
An alternate approach to actually computing the logarithm of each data point is to simply graph the data on a logarithmic scale. This is easy to do in Excel. For example, if you enter the pairs of (x, y) data points shown below and generate a standard XY (scatter) plot of the data, the graph is obviously curved, indicating a nonlinear relationship between the variables.



Now, modify the Primary Vertical Axis by clicking on *More Options*:



First, select *Vertical (Value) axis* from the Axis Options pull-down. Next click on *Logarithmic Scale* as shown below



The graph will still display exactly the same data, but will appear to represent an almost linear relationship. This is shown on in figure 36. Notice that the vertical axis now looks very different. In the original graph, the evenly spaced gridlines represented an increase in the y variable of 1,000, regardless of whether you were at the top of the axis or the bottom. The spacing on the logarithmic scale, though, increases by a factor of 10 for each gridline (from 1 to 10, 10 to 100, 100 to 1000, etc.)

You can change the scale on the Primary Horizontal axis as well, letting you create log-linear, linear-log and log-log type graphs.

Chapter 8. Constructing Regression Models in Excel

8.1 Linear Regression in Excel

We can use the LINEST function in Excel to compute the regression coefficients for a linear regression. Let's use the backpack data from the examples and perform a linear regression on it to compute "Price" as a function of "Number of Books." This file has the x-data (number of books) in cells C2:C31 and the y-data in cells A2:A31.

	A	B	C	D
1	Price	Volume	Number of Books	
2	48	2200	59	
3	45	1670	49	
4	50	2200	48	
5	42	1700	52	
6	29	1875	52	

The LINEST formula has the following syntax:

```
=LINEST(known y values, known x values, const, stats)
```

- *Const* refers to whether you want to calculate the y-intercept (the constant) from the regression (make it TRUE) or whether to force it to be zero (FALSE). We'll usually use TRUE.
- *Stats* is another true/false variable. It indicates whether to calculate and output the summary measures. We'll almost always want it to be TRUE.

However, before you type in the formula, you should know that the output of it will have ten (10) pieces of information. Obviously, we can't put ten different numbers in a single cell, so we have to enter the formula as an array calculation.

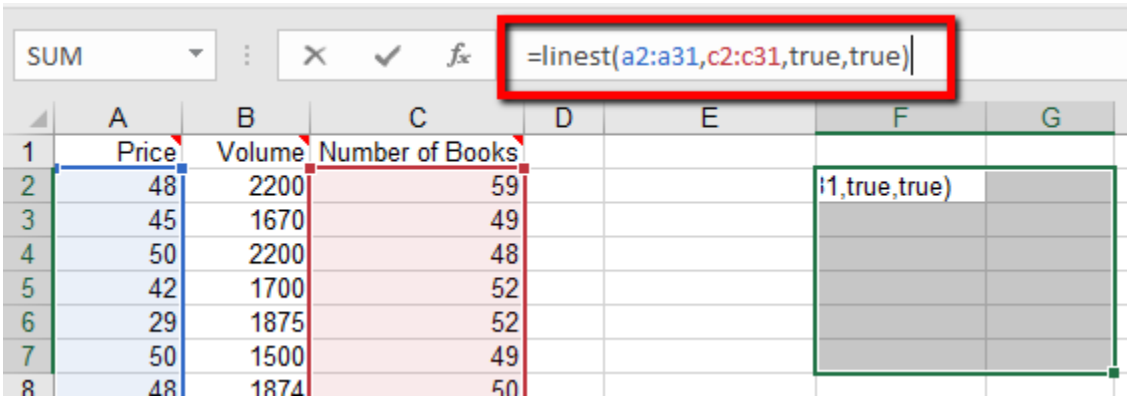
First, highlight a block of cells that is two columns wide by five columns high.

Now, type the formula in the formula area at the top of the screen

```
=LINEST(A2:A31, C2:C31, TRUE, TRUE)
```

and hit **CTRL+SHIFT+ENTER**. (If you hit enter, you will only get the first of the ten numbers; then you have to start over!)

The output will then appear in a 5 row by 2 column grid with the information shown below. The most important information is shown in bold.



The information may then be formatted as follows

	Slope	Y-intercept
Regression Coefficients:	1.46	-30.68
Standard Error (for coefficients)	0.28	13.60
R^2 (for model) and Se	0.49	9.85
F statistic and DF	27.24	28.00
SS (regression) and SS (Resid	2640.49	2714.18

We can then write the equation for the best fit line, in context, as follows:

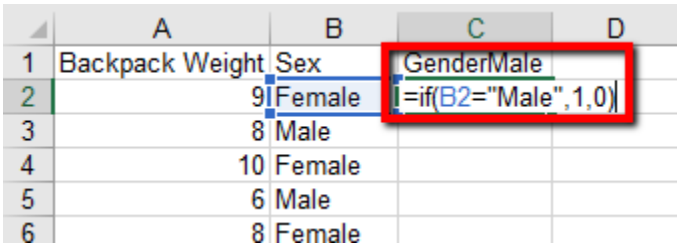
```
Price = -30.68 + 1.46 (Number of Books)
```

For more information about the LINEST function, type "regression" into the help system. If you check the "See also" portion of the help information, you will find out about the TREND function which helps you calculate other values, based on a set of known x and y values. There is also a separate SLOPE function which computes just the slope of the regression line. It has the syntax SLOPE(known y values, known x values). Used with INTERCEPT(known y values, known x values), you can get both coefficients.

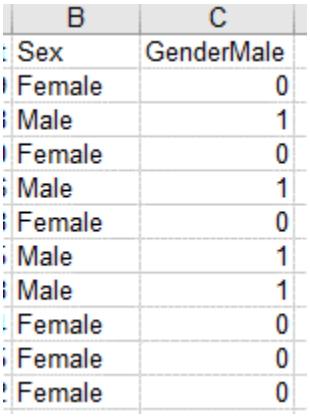
8.2 Dummy variables with IF functions in Excel

In order to perform a linear regression on a categorical variable, you will need to create a dummy variable, or a binary indicator of the value of the variable. Suppose you have a variable ”Gender” coded as ”Male” or ”Female” in column B2:B40, with the variable name ”Gender” in cell B1. Let’s say you want to create a ”GenderMale” variable in column C. First type the variable name ”GenderMale” in cell C1. Then, in cell C2, enter the formula

=IF(B2=”Male”, 1, 0)



After the formula is entered, select the cell and double-click the fill handle to copy the formula down the column.



The IF function has the following syntax.

IF(condition, value if true, value if false)

The condition can be any sort of logical condition and can include checking for whether a cell is equal to a particular value, greater than a particular value, or whatever. See the help files on ”IF” for more information.

Chapter 9. Advanced Excel Functions

9.1 Using an Excel VLOOKUP table

In doing some tasks, we find that we need some way to use different information depending on the result of some number. For example, in calculating employee pay, different job types might have different, standardized pay rates at our company. Wouldn't it be nice if Excel could figure it out from the information given and calculate the pay rate correctly? Using a lookup table, in this case a VLOOKUP table, Excel can.

If you open the file "C10 HowTo.xls" you'll see an example. Shown below is an image of the screen illustrating a sample employee database. This database contains information on each employee: hours worked that week, job type, and years of experience.

	A	B	C	D	E	F	G	H	I
1	Employee	Hours	JobType	YrsExp	GrossPay				
2	1	36	5	9	$ 1,184.40		JobType	BasePay	Raise
3	2	44	2	12	$ 741.40		1	$ 6.00	$ 0.55
4	3	20	4	13	$ 562.00		2	$ 7.25	$ 0.80
5	4	16	4	5	$ 296.00		3	$ 8.25	$ 0.95
6	5	40	5	11	$ 1,484.00		4	$ 12.50	$ 1.20
7	6	36	3	13	$ 741.60		5	$ 14.00	$ 2.10
8	7	38	1	12	$ 478.80				
9	8	42	3	2	$ 426.30				
10	9	35	3	11	$ 654.50				
11	10	27	2	11	$ 433.35				
12	11	5	2	1	$ 40.25				
13	12	18	1	9	$ 197.10				
14	13	40	4	3	$ 644.00				
15	14	37	1	13	$ 486.55				
16	15	41	3	11	$ 766.70				

Off to the right of database, in cells G2:I7 is the lookup table. (Normally, one would put this on a different sheet of the workbook and name the entire range of cells to make it easier to reference, but for this example, we wanted to keep it easy to visualize.) Now we want Excel to take the employees hours and multiply it by the correct hourly rate, based on the job type and the years of experience. This hourly pay rate will be something like

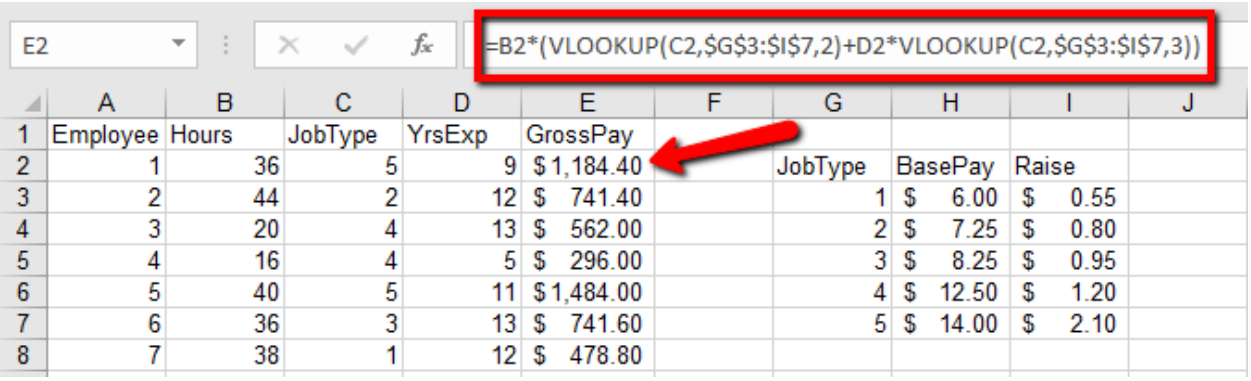
(Base Pay Rate) + (Years Experience)*(Annual Raise)

But Excel must use the Job Type to determine both the base pay rate and annual raise. To do this, we use VLOOKUP:

=VLOOKUP(Lookup Value, Lookup range, Column, [range lookup])

So, we can find the base hourly rate for employee 1 by looking up his/her job type (cell C2) in the lookup table ($G$3:$I$7 - the absolute reference is a MUST here!) and using the information in column 2 of the table. To find the annual raise, we perform the same lookup, but instead of returning the information in column 2, we want the information in column 3. Thus, we can compute employee 1's pay by the following formula (shown in text and Excel notation to make it easier to read).

Pay = (Hours Worked) * ((Base Pay Rate) + (Years Experience)*(Annual Raise))



	A	B	C	D	E	F	G	H	I	J
1	Employee	Hours	JobType	YrsExp	GrossPay					
2	1	36	5	9	$1,184.40		JobType	BasePay	Raise	
3	2	44	2	12	$ 741.40		1	$ 6.00	$ 0.55	
4	3	20	4	13	$ 562.00		2	$ 7.25	$ 0.80	
5	4	16	4	5	$ 296.00		3	$ 8.25	$ 0.95	
6	5	40	5	11	$1,484.00		4	$ 12.50	$ 1.20	
7	6	36	3	13	$ 741.60		5	$ 14.00	$ 2.10	
8	7	38	1	12	$ 478.80					

E2 = B2*(VLOOKUP(C2,$G$3:$I$7,2) + D2*VLOOKUP(C2,$G$3:$I$7,3))

Copying this formula to the cells in E3:E16 will compute each employee's pay, using the correct job type to calculate the pay rate. One could also use this to calculate the taxes based on the number of dependents declared on W4 forms, or practically anything.

IMPORTANT TIP: Lookup tables must be organized a certain way. Excel always uses the leftmost column of the table to match with the LookupValue in the formula, so be sure this is the way it is organized. It is also vital that the table be sorted in ascending order by the first column. If it is not sorted, Excel cannot find the proper match, and you will see an error in the calculation.

NICE FEATURE: Lookup tables don't have to return numbers; they can return any type of data. And, they don't require an exact match. If you have a range of possible values that should return a certain result, then just put the lower end of each range in the left column.

9.2 Computing Values of Exponentials and Logarithms

Excel uses a standard notation to compute the exponential or logarithm of a number. The notation looks a lot like the notation we have been using above:

- To compute the value of e^3, type "=EXP(3)" in a cell and hit enter.
- To get the value of e raised to whatever is in cell B2, type "=EXP(B2)"
- To compute the natural logarithm of 3, type "=LN(3)"
- To compute the natural log of the number in cell B2, type "=LN(B2)"

Note that Excel (and most calculating tools) have another logarithm function. This is the LOG(x) function. There is a slight difference between LOG(x) and LN(x). For our purposes, we will always use LN(x) when we talk about the logarithm of x.

Technical details: LOG(x) stands for the base-10 logarithm of x. LN(x) stands for the base-e logarithm of x. Essentially, when we compute a base-b logarithm of the number x we are finding the value of a so that the following equation is true: $b^a = x$. For example, since $10^2 = 100$, we know that the base-10 logarithm of 100 is 2 (i.e., $\log_{10}(100) = 2$.) Since $2^5 = 32$ we know that $\log_2(32) = 5$. Excel really only has options for base-e logarithms (LN) and base-10 logarithms (LOG). There are many other useful logarithm bases, but these are the most common, and there is a mathematical technique that relates logarithms of any two bases: $\log_b x = \frac{\ln(x)}{\ln(b)}$.

9.3 Setting up functions in Excel for shifting and Scaling

Previously, we introduced the idea of setting up an Excel spreadsheet to calculate a table of function values. We can use this same idea for calculating the values of a function with arbitrary shifts (horizontal and vertical) and scalings. For example, suppose we want to fit a shifted curve to a set of data that has x-values in cells A6:A25 and y-values in cells B6:B25. Let's add in some parameter values. Enter labels for each shift in A1:A3 and sample values for these shifts in B1:B3. You now have a worksheet that looks something like the one at the right.

Now we need to add in a column of values for the predicted y data, according to our formula, using the shifts and scales. Suppose that we want to use a logarithmic function to try and fit the data. So, we want to try to use the formula

y = (Vertical shift) + (Vertical Scale)*ln(X + Horizontal shift)

To do this, we enter the following formula into cell C6 and copy it down the column (Note: This is a formula for the logarithmic model we are currently working with; for other models, you will have to develop a different formula):

= $B$1 + $B$3*ln(A6 + $B$2)

Notice that we are using absolute cell references to look up the values of the parameters and compute the predicted y-values. This way, the constants will remain correct as we copy the formula down, but the x-values will change, based on which row we are in. This format will easily allow us to change the shifts and scales to try and match the actual data (in column B). A visual representation (a scatterplot) would also help, since the graph would help you see the shifts and try to move the graph of the predicted y-values closer to the actual data.

Chapter 10 Using SOLVER

10.1 Introduction to using SOLVER to minimize and maximize a function.

Excel has a very powerful equation solving tool built into it. This routine has limitations, and it certainly won't work for solving equations that don't have numerical values for the parameters, but it is a powerful tool for solving specific problems.

To use the solver, you need to have two things set up on your spreadsheet:

1. A cell that calculates something (the target cell)

2. Other cells (virtually any number of them) whose values are used in the calculation of the target cell (the parameter cells)

Using the solver[2] is easy, once it's set up. Select the target cell. Then activate the solver routine with "Tools/ Solver". In the dialog box, make sure the "Set Target Cell" refers to the correct target cell. Then, click on the option for what solution you want: either maximum, minimum, or exact value - like goal seek. Finally, click in the space next to "By changing cells" and then highlight the parameter cells on the worksheet (use the control key to select multiple, non-adjacent parameter cells). Finally, hit the SOLVE button and let Excel compute.

Since the process is numerical, there are some errors that may occur. First, Excel may not find a solution. This can happen for a variety of reasons, but most often it's related to having the stating values of the parameter cells too far from the solution, so try changing the values of the parameter cells and starting over. You might also get problems if your target cell involves calculations with logarithms, since the process may need to try a variety of values for each parameter and this may lead to computing the log of a negative number, which is impossible.

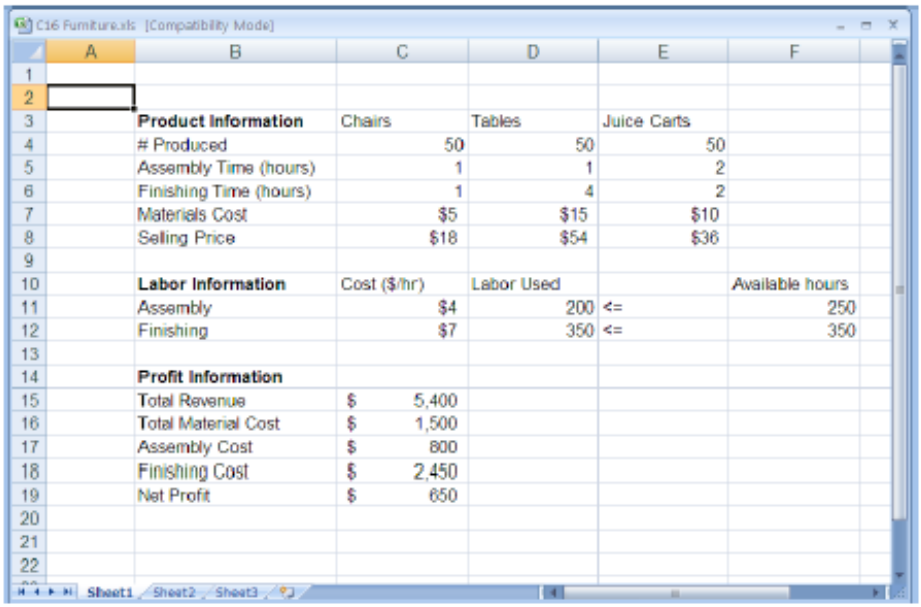
[2] The solver add-in is not always installed when you load Excel. To make it available, click on the "Tools" menu and select "Add-ins". Regardless of what other add-ins you have installed, you should see "Solver add-in" in the list of available add-ins. Check the box next to it, and then hit "OK" and from now on, solver should load in Excel and be available for solving optimization problems

10.2 Setting up constraints in Excel

In order to use Excel's routines for solving constrained optimization problems, you must first set up the spreadsheet so that (a) all the information is present and easily understood by a person reading it and (b) it contains all the formulas needed to connect the optimization variables and constraints to the objective function.

Let's return to an example, where we have translated an optimization problem into mathematical language as a preparation for solution – we were trying to determine the optimum mix of three products (chairs, tables and juice carts) to manufacture in order to maximize our profits.

The screenshot below shows this problem set up (see C16 Furniture.xls). Notice that the first block of data, in cells B3:E8 is the given information about producing each of the three products (chairs, tables and juice carts). The only part of this that is not given is the second row (B4:E4) where we start with a guess regarding the number of each product to make in order to maximize our profit. In this case, we have simply guessed at 50 of each.



	A	B	C	D	E	F
1						
2						
3		**Product Information**	Chairs	Tables	Juice Carts	
4		# Produced	50	50	50	
5		Assembly Time (hours)	1	1	2	
6		Finishing Time (hours)	1	4	2	
7		Materials Cost	$5	$15	$10	
8		Selling Price	$18	$54	$36	
9						
10		**Labor Information**	Cost ($/hr)	Labor Used		Available hours
11		Assembly	$4	200	<=	250
12		Finishing	$7	350	<=	350
13						
14		**Profit Information**				
15		Total Revenue	$ 5,400			
16		Total Material Cost	$ 1,500			
17		Assembly Cost	$ 800			
18		Finishing Cost	$ 2,450			
19		Net Profit	$ 650			
20						
21						
22						

One of the reasons for setting up the spreadsheet as shown is that calculating the various costs and total times for assembly and finishing can be done efficiently with the SUMPRODUCT formula in Excel.

The next block of information contains the labor statistics (cells B10:F12). A few words about this are needed. First, the symbols in E11:E12 are not used by Excel in solving this problem; these are given strictly to remind the user about the set up of this problem and some of the constraints present. Second, the ”total labor used” in cells D10:D12 are calculations, based on the number of each item produced and the number of hours each step takes for that item. In cell D11, the following formula has been entered:

=SUMPRODUCT($C$4:$E$4, C5:E5)

This calculation is an Excel shorthand formula for ”=C4*C5 + D4*D5 + E4*E5” which goes through the list of products and computes the assembly time needed for each product (number of items*hours per item) and then adds these times together. Using the sumproduct formula makes it easy to generalize this calculation to any number of products you might be considering, without having to type an unnecessarily long formula in. Using the absolute cell references for the product amounts (C4:E4) means that you can then directly copy the formula in D11 to cell D12 to compute the total hours of finishing labor used for this mix of products.

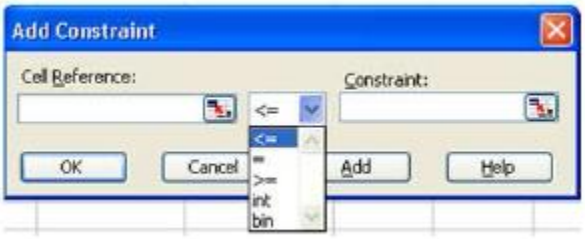
The third block of information (B14:C14) contains the calculations for the total revenue from selling the items, the total labors costs (broken down by assembly and finishing), the total materials costs and the net profit from sales. Each formula is a sumproduct, multiplying the row of information about quantities if each item produced ($C$4:$E$4) by the appropriate information (for revenue and materials costs) or it is a simple product (e.g., assembly cost in cell C17 is simply the product of total assembly hours used and cost per assembly hour = D11*C11). The net profit is then the revenue (C15) minus the total costs involved. Thus, profit is given by

=C15-(C16+C17+C18)

It is this quantity, profit, that we seek to maximize (objective function) by changing the amounts of each product we make (input variables) subject to various constraints.

10.3 Adding constraints in Solver

In the solver dialog box, clicking on the ”Add” button under ”constraints” brings up a dialog box as shown below:



Notice that there are three pieces of information needed for each constraint: the ”Cell reference” for the constraint (which must somehow be connected to the input variables so that it changes as different input variable values are tested), the ”constraint” (the value that you want to meet) and how the cell is constrained with respect to the value you provide. You have five possible options for this:

<=	Forces the value in the cell reference to be less than or equal to the constraint
=	Forces the value in the cell reference to be equal to the constraint
>=	Forces the value in the cell reference to be greater than or equal to the constraint
Int	Forces the cell value to be an integer
Bin	Forces the cell value to be a binary (either 0 or 1)

Clicking ”Add” will enter your constraint and allow you to enter another constraint. Clicking ”OK” enters the current constraint and returns you to the solver dialog box.

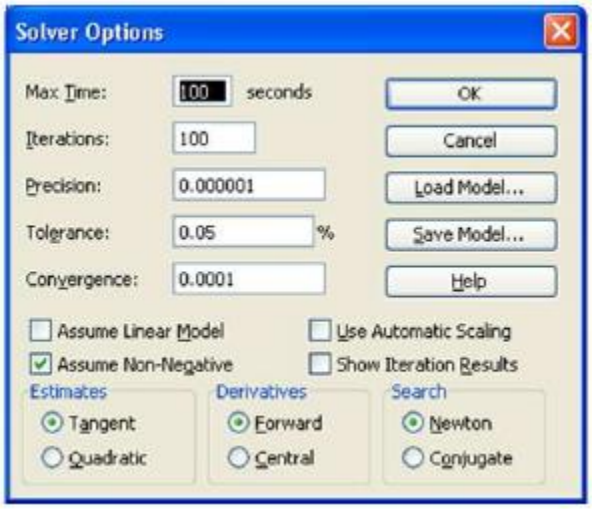
There is also a clever way to enter several constraints at once, if each is essentially the same type of constraint and the spreadsheet is properly organized. Consider the constraints shown in figure 76. (These are taken from a modified form of the solver scenario in C16 Furniture.xls.) Rather than entering one constraint for each of the product quantities to be integer, we have entered that "$C$4:$E$4 = integer" which forces each of the cells in the cell reference (C4, D4, and E4 in this case) to be an integer; this is much more efficient than entering each one separately, especially when you could have hundreds of products in the scenario.

Likewise, instead of entering the labor constraints separately, one for assembly hours and one for finishing hours, we have entered a single constraint "$D$11:$D$12 <= $F$11:$F$12". Solver goes through the list of cells on the left of the constraint and pairs them up with the corresponding cell on the right. Very efficient!

You can easily change an existing constraint in solver. Select the constraint in the solver dialog box, click "Change" and you can set all three of the options available when adding a new constraint.

Deleting a constraint is easy. Simply click on the constraint so that it is highlighted in the solver dialog box, and then click the "Delete" button.

10.4 Options in solver

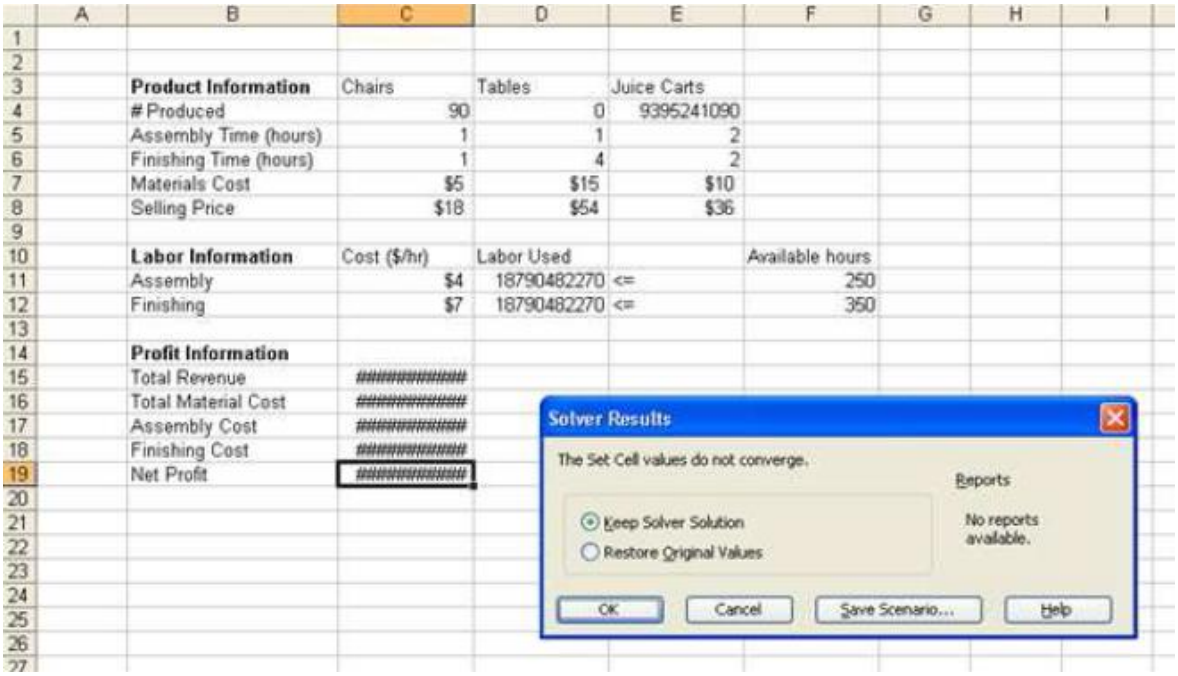


Most of the time, you will not want to change many of the options in the solver options dialog box. The first block of options all deals with the routine's limitations. The "Max time" is simply the maximum amount of time you want solver to search for a solution. The iterations relates to the number of times it loops through its procedures - there is almost no need to change these two options. Precision, tolerance and convergence all relate to the fact that, as numerical calculations, there is some rounding being done. Since all answers are approximate, these numbers let you specify how close solver must get to the specified values before it considers its work to be complete.

You will frequently encounter the need to set the "Assume linear model" and "Assume non-negative" options. Most of the rest of the options relate to the specific solution techniques used by solver. Details of these are fairly technical, and you rarely need to change them, but if you encounter a stubborn problem that defies solution, you may want to try solving it with a variety of options on these, to see if a solution can be found using alternative methods.

10.5 Errors in Solver

In example, suppose we had union requirements that led to a minimum number of labor hours at our company. If we change the constraints from ”Assembly hours <= 250” and ”Finishing hours <= 350” to be minimum hours, swapping all the ” <= ” for ” >= ” what happens?



Solving this model (with the labor hours as minimums, rather than maximums) produces the above screen. Excel tells you that ”the set cells do not converge” which means that you need to produce an infinite amount of each item in order to maximize profit. This is not a mistake or error in Excel; it is a problem with the scenario we set up. After all, if we have no maximum number of labor hours, then we can make as many of each item as we want; making more always results in more profit, so we should make as much as possible, which is infinite in this case. We can return this to a problem with a feasible solution by adding a constraint. For example, we may have a maximum amount of money available for materials.

Another possible error that could occur is ”Solver could not find a feasible solution”. This doesn’t mean the computer is broken; it simply means that the constraints you have specified over-determine the problem and cannot all be met simultaneously. You will have to carefully consider what constraints can be relaxed, and re-run the solver.

There are other possible errors that Solver may encounter. For more information, consult the help features in Excel under ”Troubleshoot Solver”.

One final note about solver. Since its routine is numerical in nature, essentially a sophisticated way of guessing an answer, checking whether it works, and then calculating how to adjust the guess, the results are highly dependent on your initial guess for the input variables. Changing them could drastically change your optimal solution, especially if your situation is highly sensitive, or if there are many possible solutions to the problem. For example, in file C16 Furniture.xls, if you change the initial values from 50, 50, and 50 for the production of chairs, tables, and juice carts, the solver routine will give very different results, summarized below.

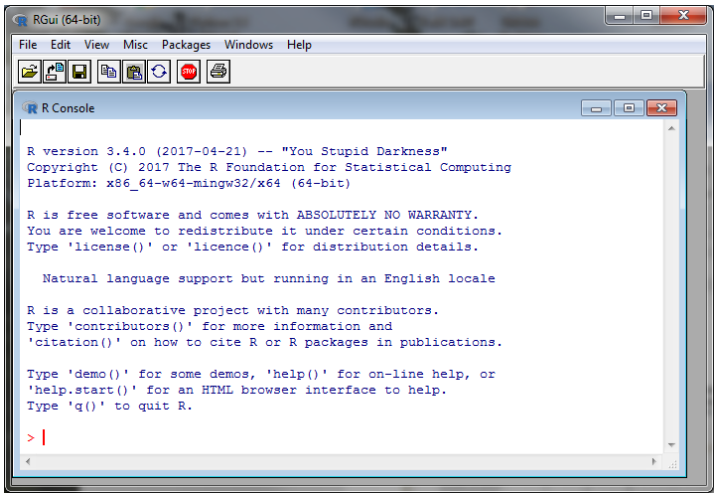
Initial Guess (C, T, J)	Optimal Solution (C, T, J)
(50, 50, 50)	(62, 34, 76)
(20, 20, 20)	(52, 34, 81)
(20, 50, 45)	(42, 34, 86)
(0, 0, 0)	(44, 34, 85)

Notice, though, that regardless of the initial values of the three variables, the number of tables in the optimal solution seems to be 34. Thus, we are simply trading off juice carts and chairs in the various solutions.

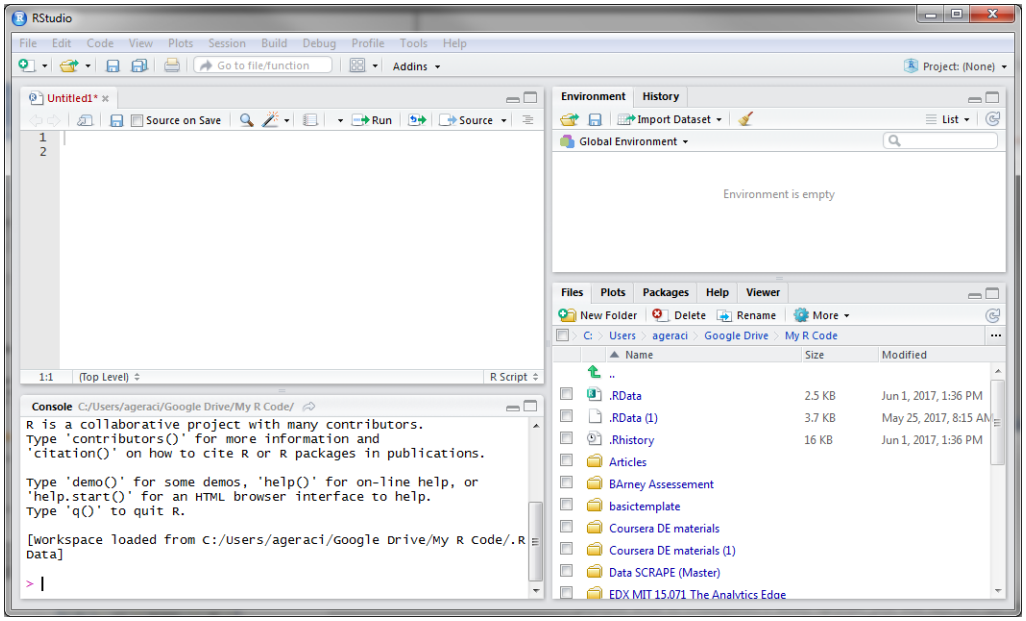
Chapter 11 Using R and R Studio

11.1 Overview of R and R Studio

R is available for Windows, Mac, and most other platforms. It is completely free and easy to update and extend to new capabilities. This software allows users to do sophisticated statistical analysis and mathematical modeling, similar to other commerical packages such as SAS, SPSS, or MiniTab.



Although it is possible to use R as a stand-alone package, in this class we will be using an interactive development environmnet (IDE) called *R Studio*, shown below, to develop our programs, called "scripts".



11.2 Downloading and Installing R and R Studio

Both R and R Studio is already installed on every SJFC lab computer. Both of the programs are available for Windows and iOS computers and may be installed on your computer for your personal use free of charge. Use the following URL's to download the install files.

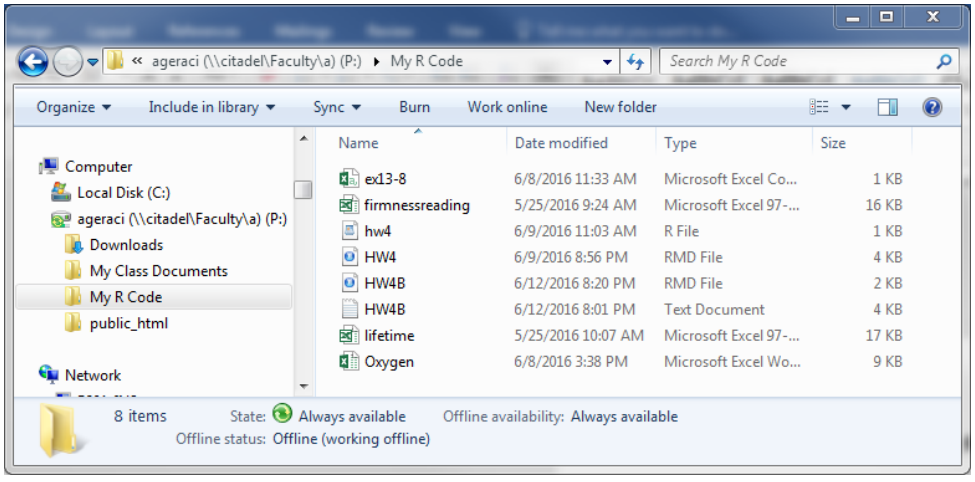
R	**https://www.r-project.org/**
R Studio Desktop	https://www.rstudio.com/products/rstudio/download2/ Choose "RStudio Desktop Open Source License" option

There are some useful ways to customize R Studio for convenience and efficiency. The following settings are strongly recommended:

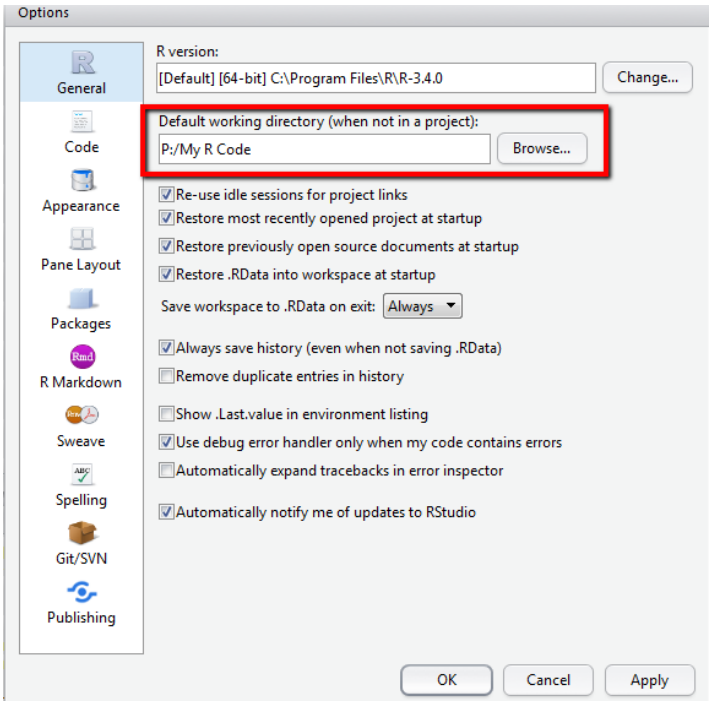
(1) Select a specific Working Directoy. Select a specific location on your computer where you will store all your work in R. This will include all programs, scripts, data files, Rmd files (Rmarkdown files), and output files.

- For example, I might create a directory called

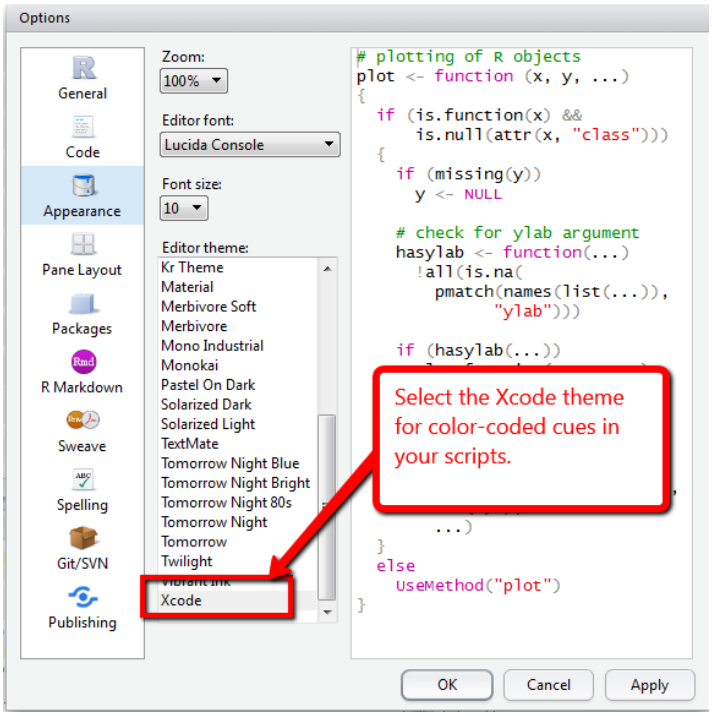
```
P:\My R Code
```

- 

(2) Establish your default working directory in R Studio by selecting Tools > Global Options, then specifying the working directory as follows:

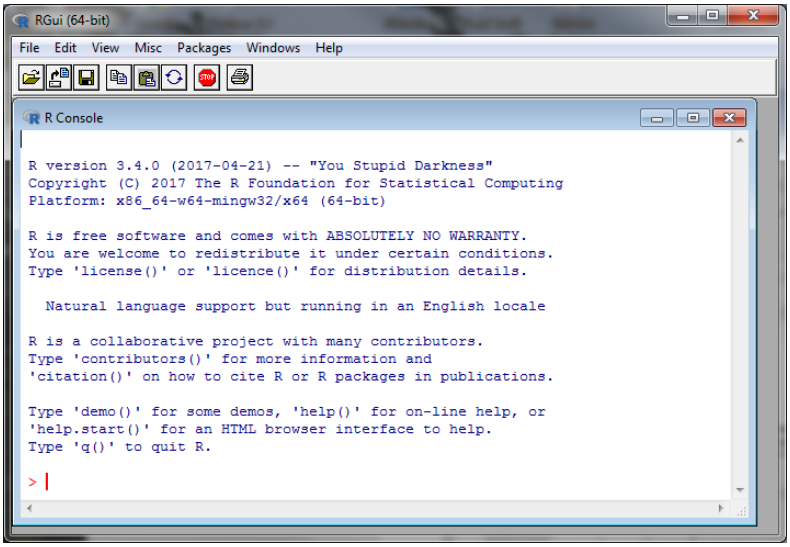
- 

(3) Select a preferred color-scheme for your R programs



11.3 The Basic R Interface

The basic R interface (figure 1) is not very exciting. It consists primarily of a menu bar, a box with a bunch of text in it, and blank space.



There are also some icons/buttons of the more commonly used commands just below the menus.

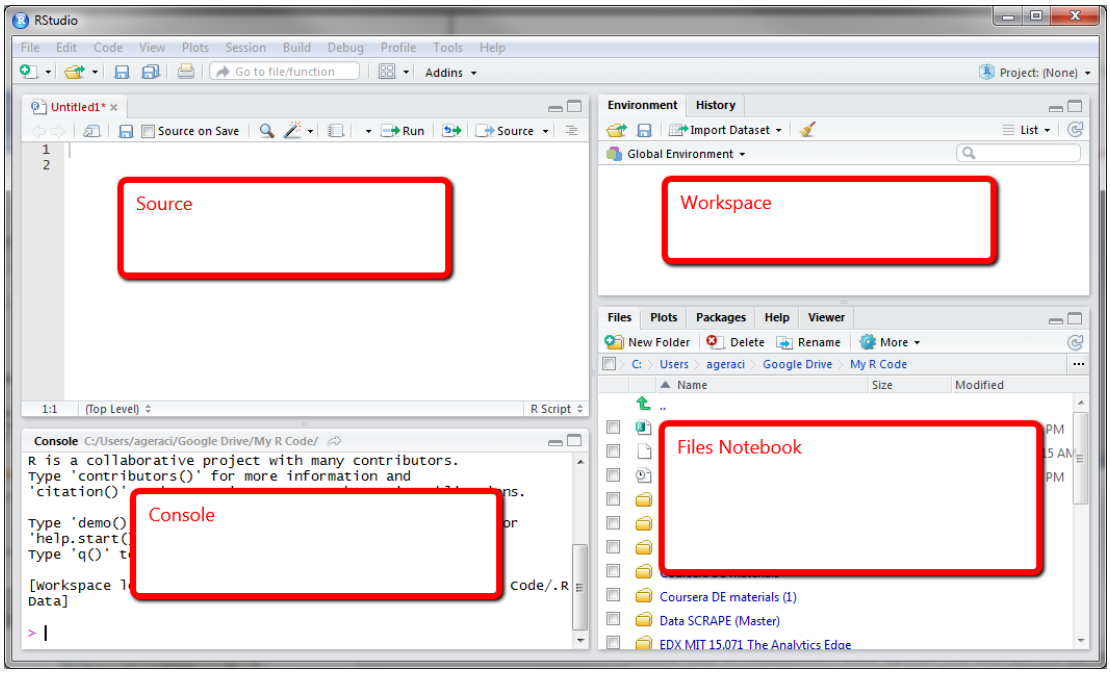
Menu options include the following:

- File: Lets you load files and save files and print stuff.
- Edit: Lets you select and copy and so forth.
- View: Lets you choose which features of the R interface are visible.
- Misc: Allows you to interrupt computations and adjust how R is working.
- Packages: Lets you load and install new packages to extend R's basic capabilities.
- Windows: Controls the various windows.
- Help: Accesses the help features.

The majority of working with R comes from typing commands into the "R Console". R is a command line program, so many of its powerful capabilities are not accessed through menus, but by typing the commands into the console.

11.4 Getting Started with RStudio Desktop

RStudio is started like most other applications, by clicking on the icon (on your desktop) or by clicking on an appropriately named file, such as `MyProgram.R`,which is stored somewhere on your computer.

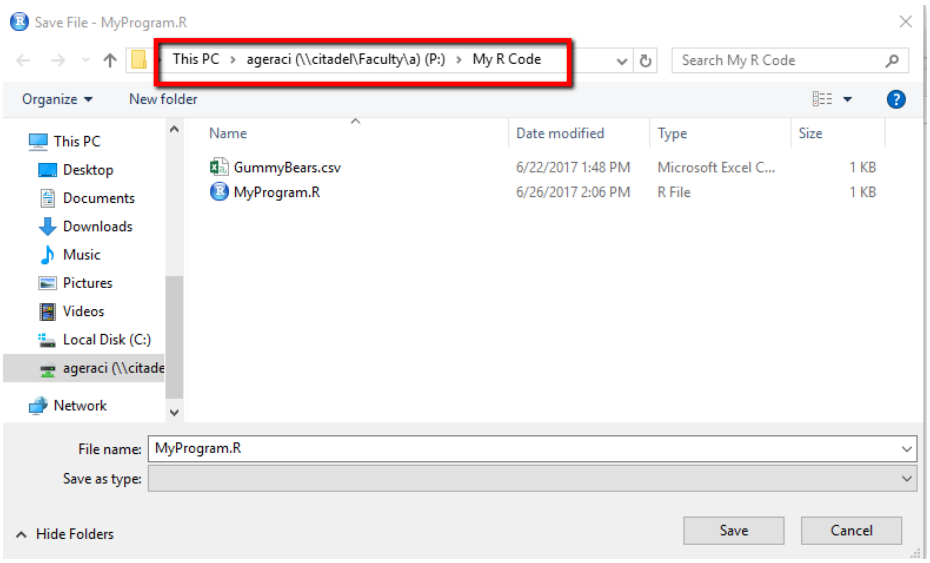


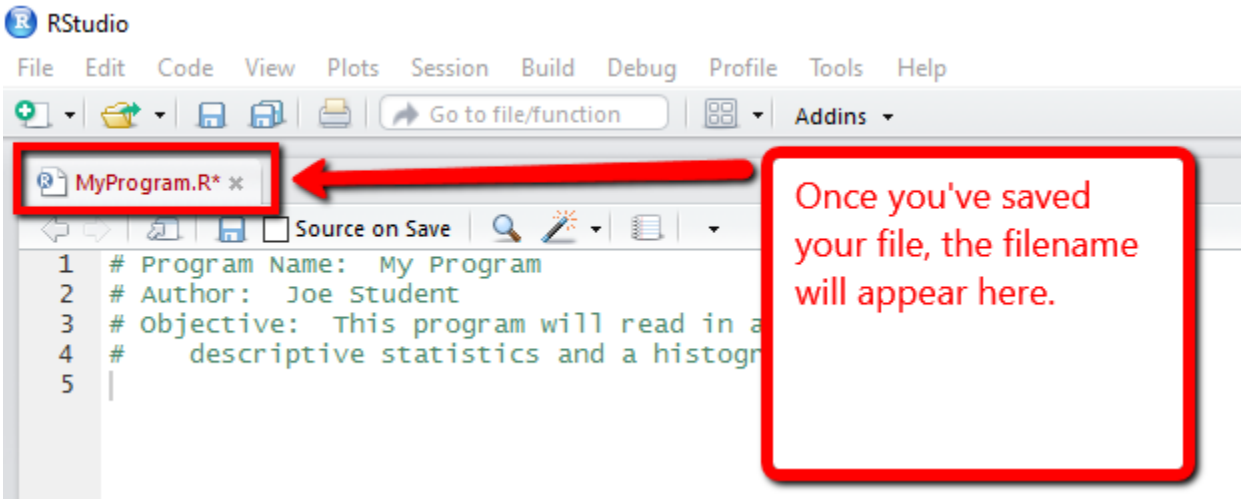
The RSTudio interface consists of several main components sitting below a top-level toolbar and menu bar. By default the four main panels are as follows:

- In the upper left is a **Source** browser for editing files or viewing datasets. This is where you will compose your script programs
- In the lower left is a **Console** for interacting with R. You may use this area to "try out" an R command before including it in your progam.
- In the upper right is a tabbed notebook widget that holds a **Workspace** browser. Use this to view stored data files or other global variables.
- The lower right area contains another tabbed notebook for interacting with **Files** and **Packages**.

To create a new R script, click File>New File>R Script (or CTRL-SHIFT-N).

It is best if you SAVE your file (CTRL S or File>Save) right away and give it a meaningful name. If you don't specify otherwise, R will save your file in your working directory.





11.5 Importing Data into R

Before you use R to analyze your data, you will need to import it. The easiest way to do this is to save the file in Excel using CSV (comma-sepearated values) format, as described in Section 2.6.

Use the following commands to read your Comma-delimited file:

Description	R code snippet

Read a CSV file directly from the web	`df <- read.csv(<URL>`
	
	`# First let's store the URL name`
	`url <= "http://citadel.sjfc.edu/faculty/ageraci/data/GummyBears.csv"`
	`# Now we read the data.`
	`df <- read.csv(url)`

Read a CSV file from your working directory	`setwd(<working directory>)`
	`df <- read.csv(<filename>)`
	
	`# Set the working directory, then read the file`
	`setwd("P:/My R Code")`
	`# This command creates a dataframe called "df"`
	`df <- read.csv("GummyBears.csv")`

Display the variables in a data frame	`str(<data frame>)`
	
	`> str(df)`
	`'data.frame': 6 obs. of 4 variables:`
	` $ Group.ID : int 1 1 1 1 1 1`
	` $ Blocks : int 1 1 5 5 9 9`
	` $ LaunchLoc: Factor w/ 2 levels "B","T": 2 1 2 1 2 1`
	` $ Distance : num 79.2 71.5 124.2 136 162 ...`
	`)`

Display the first 6 observations in a data frame	`head(<data frame>)`
	
	`> head(df)`
	`  Group.ID Blocks LaunchLoc Distance`
	`1 1 1 T 79.25`
	`2 1 1 B 71.50`
	`3 1 5 T 124.25`
	`4 1 5 B 136.00`
	`5 1 9 T 162.00`
	`6 1 9 B 140.75`

11.6 Data Wrangling

One of the strengths of the R environment is the ability to do what is known as "data wrangling" – or the process of cleaning and preparing data that might be messy or incomplete for easy access and data analyais.

Here are the questions you should ask yourself before beginning any statistical analysis of a set of data:

Do I have the data stored using the correct data types?

When you import a data file, be sure to check that the data you stored is in the proper format, using the STR command:

```
> str(df1)
'data.frame':  6384 obs. of  5 variables:
 $ ï..ID   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Day     : Factor w/ 7 levels "D","F","H","M",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ Time    : int  500 530 600 630 700 730 800 830 900 930 ...
 $ Location: Factor w/ 3 levels "Cranny","Hole",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ Count   : int  6 15 44 26 27 26 16 13 9 6 ...
```

In this example, R has interpreted Location and Day as datatype Factor, a categorical variable type. If you wish to force a variable (this is known as "data type conversion") to be a particular datatype, you might find the following functions useful:

```
as.numeric(value)
as.character(value)
as.factor(value)
as.data.frame(matrix of values)
```

Force a variable to be numeric	`df1$newvar <- as.numeric(df1$oldvar)`
	
	`> df1$NewLoc <- as.numeric(df1$Location)`
	`> str(df1$NewLoc)`
	` num [1:6384] 3 3 3 3 3 3 3 3 3 3 ...`

Force a variable to be a factor	`df1$newvar <- as.character(df1$oldvar)`
	
	`> df1$ID <- as.factor(df1$ï..ID)`
	`> str(df1$ID)`
	` Factor w/ 6384 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...`

Do I need any computed variables?

As you will read in Chapter 2 of the textbook, sometimes you need to create *computed variables* – that is, variable that are not collected directly from the problem situation, but computed later based on the *raw data*.

For example, let's say we read in the following data

```
> df2 <- read.csv("C02 SurveyData.csv")
> df2
  ID. DAY TIME NOOK CRANNY HOLE
1  95   F 1400   23     24   16
2  96   S  500    6      7    0
3 149   D 1230   38     43   35
4 150   D 1300   32     38   28
```

We see that NOOK, CRANNY, and HOLE represent the number of customers in each area of our restaurant at that time of day. We might want to know the TOTAL number of customers; computable by adding up the three individual values.

```
> df2$TotalCustomers <- df2$NOOK + df2$CRANNY + df2$HOLE
> str(df2)
'data.frame':  4 obs. of  7 variables:
 $ ID.           : int  95 96 149 150
 $ DAY           : Factor w/ 3 levels "D","F","S": 2 3 1 1
 $ TIME          : int  1400 500 1230 1300
 $ NOOK          : int  23 6 38 32
 $ CRANNY        : int  24 7 43 38
 $ HOLE          : int  16 0 35 28
 $ TotalCustomers: int  63 13 116 98
```

Another example might be that we want to compute a factor variable representing how busy we are based on the number of customers:

```
> df2$HowBusy <- ifelse(df2$TotalCustomers < 51, "Slow",
+        ifelse(df2$TotalCustomers < 100, "Moderate", "Busy"))
> df2$HowBusy <- as.factor(df2$HowBusy)
> str(df2)
'data.frame':  4 obs. of  8 variables:
 $ ID.           : int  95 96 149 150
 $ DAY           : Factor w/ 3 levels "D","F","S": 2 3 1 1
 $ TIME          : int  1400 500 1230 1300
 $ NOOK          : int  23 6 38 32
 $ CRANNY        : int  24 7 43 38
 $ HOLE          : int  16 0 35 28
 $ TotalCustomers: int  63 13 116 98
 $ HowBusy       : Factor w/ 3 levels "Busy","Moderate",..: 2 3 1 2
```

Note that in this case we use the as.factor functionality to ensure that the HowBusy variable is represented as a categorical variable in R.

11.7 Less Volume, More Creativity – the MOSAIC package[3]

One of the reasons why R is becoming so popular in statistics and data science is because of the ability to use the ever-growing number of libraries of open-source functions known as *packages.* In this course we will be using the MOSAIC packages (http://mosaic-web.org/), which will equip us with a simple yet powerful framework to produce all the results necessary for this course.

All of the MOSAIC functions will use the following form:

Formula Template

goal (y ~ x , data = mydata)

Where "goal" is something that you want to do, such as compute a MEAN or produce an XYPLOT. Sometimes the *y* is not needed:

```
goal(~x, data=mydata)
```

and sometimes we may include a third part:

```
goal(y~x|z, data=mydata)
```

But, in general we will always use commands in this form:

```
goal( formula , data=mydata)
```

To apply the template to your needs, answer the following quesitons:

1. What do you want R to do? This is the goal.
2. What must R know to do that? These are the inputs to the function.

[3] Much of this material, including the chapter title, is adapted from *Start Teaching with R*, by R Pruim, N. Norton, and D. Kaplan (Nov 2015), licensed under CC BY 3.0.

The following code snippets will be used frequently in this class to perform common functions. Each MOSAIC **function** is indicated in bold. Examples all use the HELPrct data (Health Evaluation and Linkage to Primary Care).

Description	**R code snippet**
Install (once on each computer) or use (do this each time you run your code) a library	**install.package**("<package name>") **require**("<package name>") require(mosaic)
Get help on a function or dataset	??<function> or ?<function> NOTE: type this in the Console window rather than in a program script. ??histogram ?HELPrct
Compute mean, median, or standard deviation for a variable	**mean**(<formula>, data=MYDATA) **sd**(<formula>, data=MYDATA) **median**(<formula>, data=MYDATA) mean(~cesd, data=HELPrct) sd(~ cesd |sex, data=HELPrct) median(~ cesd | homeless + sex, data=HELPrct)
Compute descriptive statistics for a variable	**favstats**(<formula>, data=MYDATA) favstats(~cesd, data=HELPrct) favstats(~cesd | sex , data=HELPrct) favstats(cesd ~ sex , data=HELPrct) favstats(~cesd | sex + homeless , data=HELPrct)
Create tables	**tally**(<formula>, data=MYDATA) tally(~sex, data=HELPrct) tally(~ sex + homeless, data=HELPrct, margins=TRUE) tally(~ sex + homeless, data=HELPrct, format="proportion") tally(~ sex + homeless, data=HELPrct, format="percent")

Create a histogram	**histogram**(<formula>, data=MYDATA)
	
	histogram(~cesd,data= HELPrct)
	histogram(~cesd, width=5,center=2.5, data= HELPrct)
	histogram(~cesd |sex, data= HELPrct)
	histogram(~cesd |sex, layout=c(1,2), data= HELPrct)
	#Vertical arrangement
	histogram(~cesd | sex + homeless, layout=c(2,2), data=HELPrct)

Create a bargraph (from unsummarized, stacked categorical data)	**bargraph**(<formula>, data=MYDATA) bargraph(~anysub, data=HELPrct) bargraph(~substance, data=HELPrct, horizontal=TRUE) bargraph(~substance, data=HELPrct, scales=list(x=list(rot=45))) bargraph(~ substance, data=HELPrct, groups= sex, auto.key=TRUE) bargraph(~ substance, data=HELPrct, groups= sex, auto.key=list(space="right")) bargraph(~ substance | sex , data=HELPrct, auto.key=list(space="right")) bargraph(~ substance, groups = homeless, auto.key=TRUE,data = HELPrct %>% filter(sex == "male"))

Create a boxplot	**bwplot**(<formula>, data=MYDATA)
	
	# Box-whisker plot with a line for MEDIAN
	bwplot(~cesd,pch = "|", data=HELPpct)
	bwplot(~cesd | sex, pct="|",data=HELPpct)
	bwplot(~cesd | sex, layout=c(1,2), data=HELPrct, pch="|")

Compute linear correlation	**cor**(<formula>, use="complete.obs", data=MYDATA)
	
	cor(cesd ~ mcs + dayslink, data=HELPrct, use="complete.obs")
	#Correlation matrix of 4 numeric vars
	cor(select(HELPrct, cesd, mcs, daysanysub, dayslink))
	# What if we want to use all numeric variables?
	cor(HELPrct[,sapply(HELPrct,is.numeric)], data=HELPrct, use="complete.obs")

Create a scatterplot	**xyplot**(<formula>, xlab="<label>", ylab = "<label>",
	main = "<label>", data=MYDATA)
	
	xyplot(cesd ~ sex , data=HELPrct)
	xyplot(sex ~ cesd , data=HELPrct)
	xyplot(sex ~ cesd , alpha = .6, cex = 1.4, data=HELPrct)
	#Scatterplot with labels xyplot(cesd ~ mcs, xlab="CESD - depressive symptoms",
	ylab="SF-36 Mental Component Score",
	main = "Depression vs. Mental State",data=HELPrct)

Create a scatterplot with trendline and groups	same format as above PLUS: ...., groups = <variable>, auto.key=TRUE, type = c("p", "r"), data=MYDATA) xyplot(cesd ~ mcs, xlab="CESD - depressive symptoms", ylab="SF-36 Mental Component Score", main = "Depression vs. Mental State", groups = sex, auto.key=TRUE, type=c("p","r"),data=HELPrct)

Compute a linear regression Note: lm is actually part of the base R packages, but uses the same formula syntax as MOSAIC does	**lm**(<formula>, data=HELPrct) -> m; **summary**(m) lm(cesd ~ daysanysub, data=HELPrct) -> m; summary(m) lm(cesd ~ daysanysub + drugrisk + dayslink + age + i1 + indtot, data=HELPrct) -> m; summary(m) lm(cesd ~ daysanysub + drugrisk + i1 + indtot + sex, data=HELPrct) -> m; summary(m) lm(cesd ~ daysanysub + drugrisk + i1 + indtot + sex + racegrp, data=HELPrct) -> m; summary(m) lm(cesd ~ daysanysub + indtot + sex + substance, data=HELPrct) -> m; summary(m) lm(cesd ~ daysanysub + indtot + daysanysub*sex, data=HELPrct) -> m; summary(m)
Retrieve just the coefficients for a model	**lm**(<formula>, data=HELPrct) -> m; **coef**(m) > lm(cesd ~ mcs, data=HELPrct) -> m; coef(m) (Intercept) mcs 53.902188 -0.664669
Compute confidence intervals for the coefficients	**lm**(<formula>, data=HELPrct) -> m; **confint**(m) > lm(cesd ~ mcs, data=HELPrct) -> m; confint(m) 2.5 % 97.5 % (Intercept) 51.6475997 56.1567763 mcs -0.7306418 -0.5986962
Create Regression Diagnostic Graphs	Perform lm function, as described above, then ... **xyplot**(resid(m) ~ predict(m), xlab="Fitted Values", ylab="Residual Values", main = "Residual vs. Fitted Diagnostic Graph",data=HELPrct) **xyplot**(predict(m) ~ cesd, xlab="Observed Values", ylab="Fitted Values", main = "Fitted vs. Actual Diagnostic Graph", type = c("p", "r"), data=HELPrct)
Use Regression model to make predictions	lm(cesd ~ daysanysub, data=HELPrct) -> m; summary(m) # Predict the CESD for a value of daysanysub = 70 **predict(m, data.frame(daysanysub = 70))** # .... same computation using the lm parameters 32.41366 + 0.00409 * (70)

11.8 Four things to know about R

1. As with any computer program, **R will only do what you tell it to**. Before asking for help on any problem, ask yourself these two questions:

   - What do you want R to do? This will generally determine which function to use.

   - What must R know in order to do that? This will determine the inputs to the function

2. **R is case sensitive**. If you mis-capitalize something in R it won't do what you want. Pay careful attention to the spelling and capitalization of variables and datasets.

   ```
   Mydata is not the same as mydata
   ```

3. Functions in R use the following syntax:

   ```
   Functionname( argument1, argument2, ...)
   ```

   - The agumnents are <u>always</u> surrounded by (round) parenthases and separated by commas.

   - If you type a function name without the parentheses, you will see the *code* for that function (this generally isn't what you want).

4. TAB completion and arrows can improve your typing speed and accuracy.

   - If you begin a command and hit the TAB key, R will show you a list of possible ways to complete the command.

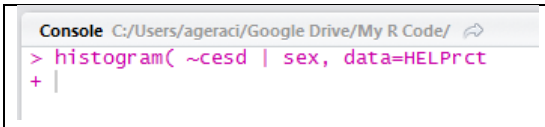
   - If you hit TAB after the opening parenthasis of a fucntion, R will display the list of argumnents it expects

5. Data stored in R is not "recomputed" dynamically in the same way it is in Excel. If you compute the TOTAL of two variables, but then change the value of one of the variables, the TOTAL variable will not change unless you re-compute it.

11.9 Common R Errors

Unfortunately, error messages in R can be difficult to understand. First of all, look at the following:

- Did you spell your variables correctly? Remember that a variable named "Gender" is not the same as a variable named "gender".
- Did you use a comma in between each argument of the function?
- Did you remember the "~" in front of the variable, as in histogram(~ cesd ...

	If you see a + prompt in the Console, it means R is waiting for more input. Often this means that you have forgotten a closing parenthesis or made dome other syntax error. ? How to fix it: Press ESC key to return to the > prompt and start the command fresh.
`histogram(~ aeg, data= HELPrct)` `Error in xxxxxxxxx : object 'aeg' not found`	R was not expecting to find "xxxx" where it did. You probably either misspelled something, forgot a comma, or put the arguments in the incorrect order.
`text3 <- hello` `Error in eval(expr, envir, enclos): object 'hello' not found`	You used unquoted text, hello, instead of quoted text, "hello". R thinks that hello is a defined variable

`c(1,2 3)`
 	
`Error in c(): unexpected numeric constant in "c(1,2 3"`
	This vector is missing a comma between the 2 and the 3.

`time[3]`
 	
`Error in time[3]: object of type 'closure' is not subsettable.`
	The data frame time has not been defined. Did you mean to use the time() function?

Appendix A. Sample Data

mosaicdata – Alcohol

These data provide per capita alcohol consumption values for many countries in 2005 and 2008. There are also a few countries for which there are data in other years

```
country country name
year year
alcohol per capita alcohol consumption
```

mosaicdata – Gestation

Birth weight, date, and gestational period collected as part of the Child Health and Development

Studies in 1961 and 1962. Information about the baby's parents — age, education, height, weight, and whether the mother smoked is also recorded.

```
id - identification number
plurality - 5 = single fetus
outcome 1 = live birth that survived at least 28 days
date birth date where 1096=January 1, 1961
gestation - length of gestation (in days)
sex infant's sex (1=male, 2=female)
wt birth weight (in ounces)
parity total number of previous pregnancies (including fetal deaths and still births)
race mother's race: 0-5=white 6=mex 7=black 8=asian 9=mixed
age mother's age in years at termination of pregnancy
ed mother's education: 0= less than 8th grade, 1 = 8th -12th grade - did not graduate, 2=
HS graduate-no other schooling, 3= HS+trade, 4=HS+some college, 5=College graduate,
6=Trade school, 7=HS unclear
ht mother's height in inches to the last completed inch
wt.1 mother's prepregnancy weight (in pounds)
drace father's race (a factor with levels equivalent to mother's race)
dage father's age (in years)
ded father'ed education (same coding as mother's education)
dht father's height in inches to the last completed inch
dwt father's weight (in pounds)
marital marital status: 1=married, 2=legally separated, 3=divorced, 4=widowed, 5=never married
inc family yearly income in $2500 increments: 0=under 2500,1=2500-4999,...,8=12,500-14,999, 9=15000+
smoke does mother smoke? 0=never,1=smokes now,2=until current pregnancy,3=once did, not now time
   time since quitting smoking: 0=never smoked,1=still smokes,2=during current preg,3=within 1
   yr,4=1-2 years ago, 5=2-3 yr ago, 6=3-4yrs ago, 7=5-9yrs ago,8=10+yrs ago, 9=quit and don't know
number number of cigs smoked per day for past and current smokers 0=never, 1=1-4, 2=5-9,3=10-14,
   4=15-19, 5=20-29, 6=30-39, 7=40-60, 8=60+, 9=smoke but don't know
```

mosaicdata - HELPrct: Health Evaluation and Linkage to Primary Care

The HELP study was a clinical trial for adult inpatients recruited from a detoxification unit. Eligible subjects were adults, who spoke Spanish or English, reported alcohol, heroin or cocaine as their first or second drug of choice, resided in proximity to the primary care clinic to which they would be referred or were homeless. Subjects were interviewed at baseline during their detoxification stay and follow-up interviews were undertaken every 6 months for 2 years.

Source: http://www.math.smith.edu/help

Data frame with 453 observations on the following variables.

- `age` subject age at baseline (in years)
- `anysub` use of any substance post-detox: a factor with levels `no` `yes`
- `cesd` Center for Epidemiologic Studies Depression measure at baseline (high scores indicate more depressive symptoms)
- `d1` lifetime number of hospitalizations for medical problems (measured at baseline)
- `daysanysub` time (in days) to first use of any substance post-detox
- `dayslink` time (in days) to linkage to primary care
- `drugrisk` Risk Assessment Battery drug risk scale at baseline
- `e2b` number of times in past 6 months entered a detox program (measured at baseline)
- `female` 0 for male, 1 for female
- `sex` a factor with levels `male` `female`
- `g1b` experienced serious thoughts of suicide in last 30 days (measured at baseline): a factor with levels `no` `yes`
- `homeless` housing status: a factor with levels `housed` `homeless`
- `i1` average number of drinks (standard units) consumed per day, in the past 30 days (measured at baseline)
- `i2` maximum number of drinks (standard units) consumed per day, in the past 30 days (measured at baseline)
- `id` subject identifier
- `indtot` Inventory of Drug Use Consequences (InDUC) total score (measured at baseline)
- `linkstatus` post-detox linkage to primary care (0 = no, 1 = yes)
- `link` post-detox linkage to primary care: `no` `yes`
- `mcs` SF-36 Mental Component Score (measured at baseline, lower scores indicate worse status)
- `pcs` SF-36 Physical Component Score (measured at baseline, lower scores indicate worse status)
- `pss_fr` perceived social support by friends (measured at baseline, higher scores indicate more support)
- `racegrp` race/ethnicity: levels `black` `hispanic` `other` `white`
- `satreat` any BSAS substance abuse treatment at baseline: `no` `yes`
- `sexrisk` Risk Assessment Battery sex risk score (measured at baseline)
- `substance` primary substance of abuse: `alcohol` `cocaine` `heroin`
- `treat` randomized to HELP clinic: `no` `yes`

mosaicdata – Kidsfeet

These data were collected by a statistician, Mary C. Meyer, in a fourth grade classroom in Ann Arbor, MI, in October 1997. They are a convenience sample — the kids who were in the fourth grade.

```
name a factor with levels corresponding to the name of each child
birthmonth the month of birth
birthyear the year of birth
length length of longer foot (in cm)
width width of longer foot (in cm)
sex a factor with levels  B G
biggerfoot a factor with levels L R
domhand a factor with levels L R
```

mosaicdata – Marriage

Marriage records from the Mobile County, Alabama, probate court.

```
bookpageID a factor with levels for each book and page (unique identifier)
appdate a factor with levels corresponding to each of the dates on which the application was
filed (in the form MO/DY/YY, e.g. 1/22/99 represents January 22, 1999)
ceremonydate a factor with levels corresponding to the date of the ceremony
delay number of days between the application and the ceremony
officialTitle a factor with levels BISHOP CATHOLIC PRIEST CHIEF CLERK CIRCUIT JUDGE ELDER MARRIAGE OFFICIAL
   MINISTER PASTOR REVEREND
person a factor with levels Bride Groom
dob a factor with levels corresponding to the date of birth of the person
age age of the person (in years)
race a factor with levels American Indian Black Hispanic White
prevcount the number of previous marriages of the person, as listed on the application
prevconc the way the last marriage ended, as listed on the application hs the number of years of
   high school education, as listed on the application
college the number of years College education, as listed on the application.
dayOfBirth the day of birth, as a number from 1 to 365 counting from January 1
sign the astrological sign, with levels Aquarius Aries Cancer Capricorn Gemini Leo Libra Pisces Saggitarius
   Scorpio Taurus Virgo
```

mosaicdata – Riders

The Pioneer Valley Planning Commission (PVPC) collected data north of Chestnut Street in Florence, MA for ninety days from April 5, 2005 to November 15, 2005. Data collectors set up a laser sensor, with breaks in the laser beam recording when a rail-trail user passed the data collection station.

```
date - date of data collection (POSIXct)
day - a factor with levels (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and, Sunday)
highT - high temperature for the day (in degrees Fahrenheit)
lowT - low temperature for the day (in degrees Fahrenheit)
hi -shorter name for  highT
lo shorter name for lowT
precip inches of precipitation
clouds measure of cloud cover (in oktas)
riders estimated number of trail crossings that day (number of breaks recorded)
ct shorter name for riders
weekday type of day: a factor with levels N (weekend or holiday), Y (non-holiday weekday)
wday shorter name for weekday
```

mosaic – SaratogaHoues

Data on houses in Saratoga County, New York, USA in 2006

```
price price (1000s of US dollars)
lotSize size of lot (square feet)
age age of house (years)
landValue value of land (1000s of US dollars)
livingArea living are (square feet)
pctCollege percent of neighborhood that graduated college
bedrooms number of bedrooms
firplaces number of fireplaces
bathrooms number of bathrooms (half bathrooms have no shower or tub)
rooms number of rooms
heating type of heating system
fuel fuel used for heating
sewer type of sewer system
waterfront whether property includes waterfront
newConstruction whether the property is a new construction
centralAir whether the house has central air
```

mosaicdata – Utilities

Data from utility bills at a residence.

```
month month (coded as a number)
day day of month on which bill was calculated
year year of bill
temp average temperature (F) for billing period
kwh electricity usage (kwh)
ccf gas usage (ccf)
thermsPerDay a numeric vector
billingDays number of billing days in billing period
totalbill total bill (in dollars)
gasbill gas bill (in dollars)
elecbill exectric bill (in dollars)
notes notes about the billing period
```

(good example of computed variables – ccfpday, kwhpday, gassbillpday, elecbillpday, totalbillpday, thermsperday, monthssinceY2K) – used in Kaplans "Statistical Modeling: A fresh Approach, 2009)

fivethirtyeight – airline_safety

The raw data behind the story "Should Travelers Avoid Flying Airlines That Have Had Crashes in the Past?" http://fivethirtyeight.com/features/should-travelers-avoid-flying-airlines-that-have-had-crashes-in-the-past/.

```
Classes 'tbl_df', 'tbl' and 'data.frame':56 obs. of  9 variables:
 $ airline               : chr  "Aer Lingus" "Aeroflot" "Aerolineas Argentinas" "Aeromexico" ...
 $ incl_reg_subsidiaries : logi  FALSE TRUE FALSE TRUE FALSE FALSE ...
 $ avail_seat_km_per_week: num  3.21e+08 1.20e+09 3.86e+08 5.97e+08 1.87e+09 ...
 $ incidents_85_99       : int  2 76 6 3 2 14 2 3 5 7 ...
 $ fatal_accidents_85_99 : int  0 14 0 1 0 4 1 0 0 2 ...
 $ fatalities_85_99      : int  0 128 0 64 0 79 329 0 0 50 ...
 $ incidents_00_14       : int  0 6 1 5 2 6 4 5 5 4 ...
 $ fatal_accidents_00_14 : int  0 1 0 0 0 2 1 1 1 0 ...
 $ fatalities_00_14      : int  0 88 0 0 0 337 158 7 88 0 ...
```

fivethirtyeight – bad_drivers

The raw data behind the story "Dear Mona, Which State Has The Worst Drivers?"http://fivethirtyeight.com/datalab/which-state-has-the-worst-drivers/

```
Classes 'tbl_df', 'tbl' and 'data.frame':51 obs. of  8 variables:
 $ state              : chr  "Alabama" "Alaska" "Arizona" "Arkansas" ...
 $ num_drivers        : num  18.8 18.1 18.6 22.4 12 13.6 10.8 16.2 5.9 17.9 ...
 $ perc_speeding      : int  39 41 35 18 35 37 46 38 34 21 ...
 $ perc_alcohol       : int  30 25 28 26 28 28 36 30 27 29 ...
 $ perc_not_distracted: int  96 90 84 94 91 79 87 87 100 92 ...
 $ perc_no_previous   : int  80 94 96 95 89 95 82 99 100 94 ...
 $ insurance_premiums : num  785 1053 899 827 878 ...
 $ losses             : num  145 134 110 142 166 ...
```

fivethirtyeight – bechdel

The raw data behind the story "The Dollar-And-Cents Case Against Hollywood's Exclusion of Women" http://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-exclusion-of-women/.

```
Classes 'tbl_df', 'tbl' and 'data.frame':1794 obs. of  15 variables:
 $ year         : int  2013 2012 2013 2013 2013 2013 2013 2013 2013 2013 ...
 $ imdb         : chr  "tt1711425" "tt1343727" "tt2024544" "tt1272878" ...
 $ title        : chr  "21 & Over" "Dredd 3D" "12 Years a Slave" "2 Guns" ...
 $ test         : chr  "notalk" "ok-disagree" "notalk-disagree" "notalk" ...
 $ clean_test   : chr  "notalk" "ok" "notalk" "notalk" ...
 $ binary       : chr  "FAIL" "PASS" "FAIL" "FAIL" ...
 $ budget       : int  13000000 45000000 20000000 61000000 40000000 225000000 92000000 12000000
   13000000 130000000 ...
 $ domgross     : num  25682380 13414714 53107035 75612460 95020213 ...
 $ intgross     : num  4.22e+07 4.09e+07 1.59e+08 1.32e+08 9.50e+07 ...
 $ code         : chr  "2013FAIL" "2012PASS" "2013FAIL" "2013FAIL" ...
 $ budget_2013  : int  13000000 45658735 20000000 61000000 40000000 225000000 92000000 12000000
   13000000 130000000 ...
 $ domgross_2013: num  25682380 13611086 53107035 75612460 95020213 ...
 $ intgross_2013: num  4.22e+07 4.15e+07 1.59e+08 1.32e+08 9.50e+07 ...
 $ period_code  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ decade_code  : int  1 1 1 1 1 1 1 1 1 1 ...
```

fivethirtyeight – biopics

The raw data behind the story "'Straight Outta Compton' Is The Rare Biopic Not About White Dudes" http://fivethirtyeight.com/features/straight-outta-compton-is-the-rare-biopic-not-about-white-dudes/.

```
Classes 'tbl_df', 'tbl' and 'data.frame':761 obs. of  14 variables:
 $ title             : chr  "10 Rillington Place" "12 Years a Slave" "127 Hours" "1987" ...
 $ site              : chr  "tt0066730" "tt2024544" "tt1542344" "tt2833074" ...
 $ country           : chr  "UK" "US/UK" "US/UK" "Canada" ...
 $ year_release      : int  1971 2013 2010 2014 1998 2008 2002 2013 1994 1987 ...
 $ box_office          : num  NA 56700000 18300000 NA 537000 81200000 1130000 95000000 19600000
   1080000 ...
 $ director          : chr  "Richard Fleischer" "Steve McQueen" "Danny Boyle" "Ricardo Trogi" ...
 $ number_of_subjects: int  1 1 1 1 1 1 1 1 1 2 ...
 $ subject           : chr  "John Christie" "Solomon Northup" "Aron Ralston" "Ricardo Trogi" ...
 $ type_of_subject   : chr  "Criminal" "Other" "Athlete" "Other" ...
 $ race_known        : chr  "Unknown" "Known" "Unknown" "Known" ...
 $ subject_race      : chr  NA "African American" NA "White" ...
 $ person_of_color   : logi  FALSE TRUE FALSE FALSE FALSE TRUE ...
 $ subject_sex       : chr  "Male" "Male" "Male" "Male" ...
 $ lead_actor_actress: chr  "Richard Attenborough" "Chiwetel Ejiofor" "James Franco" "Jean-Carl
   Boucher" ...
```

fivethirtyeight – classic_rock_raw_data

The raw data behind the story "Why Classic Rock Isn't What It Used To Be"http://fivethirtyeight.com/features/why-classic-rock-isnt-what-it-used-to-be/.

(Songs played by Classic Rock radio stations during one week in June, 2014)

```
Classes 'tbl_df', 'tbl' and 'data.frame':37673 obs. of  7 variables:
 $ song     : chr  "Caught Up in You" "Caught Up in You" "Caught Up in You" "Caught Up in You" ...
 $ artist   : chr  ".38 Special" ".38 Special" ".38 Special" ".38 Special" ...
 $ callsign : chr  "KGLK" "KGB" "KGB" "KGLK" ...
 $ time      : int  1402943314 1403398735 1403243924 1403470732 1403380737 1403105300 1402970932
   1403456303 1403056697 1403179167 ...
 $ date_time: POSIXct, format: "2014-06-16 14:28:34" "2014-06-21 20:58:55" "2014-06-20 01:58:44"
   ...
 $ unique_id: chr  "KGLK1536" "KGB0260" "KGB0703" "KGLK0036" ...
 $ combined : chr  "Caught Up in You by .38 Special" "Caught Up in You by .38 Special" "Caught Up
   in You by .38 Special" "Caught Up in You by .38 Special" ...
```

fivethirtyeight – college_all_ages

The raw data behind the story "The Economic Guide To Picking A College Major"http://fivethirtyeight.com/features/the-economic-guide-to-picking-a-college-major/..

```
Classes 'tbl_df', 'tbl' and 'data.frame':173 obs. of  11 variables:
 $ major_code                 : int  1100 1101 1102 1103 1104 1105 1106 1199 1301 1302 ...
 $ major                      : chr  "General Agriculture" "Agriculture Production And Management"
   "Agricultural Economics" "Animal Sciences" ...
 $ major_category               : chr  "Agriculture & Natural Resources" "Agriculture & Natural
   Resources" "Agriculture & Natural Resources" "Agriculture & Natural Resources" ...
 $ total                       : int  128148 95326 33955 103549 24280 79409 6586 8549 106106 69447
   ...
 $ employed                    : int  90245 76865 26321 81177 17281 63043 4926 6392 87602 48228 ...
 $ employed_fulltime_yearround: int  74078 64240 22810 64937 12722 51077 4042 5074 65238 39613 ...
 $ unemployed                  : int  2423 2266 821 3619 894 2070 264 261 4736 2144 ...
 $ unemployment_rate           : num  0.0261 0.0286 0.0302 0.0427 0.0492 ...
 $ p25th                        : num  34000 36000 40000 30000 38500 35000 39400 35000 38000 40500
   ...
 $ median                       : num  50000 54000 63000 46000 62000 50000 63000 52000 52000 58000
   ...
 $ p75th                        : num  80000 80000 98000 72000 90000 75000 88000 75000 75000 80000
   ...
```

fivethirtyeight – daily_show_guests

The raw data behind the story "Every Guest Jon Stewart Ever Had On 'The Daily Show'" http://fivethirtyeight.com/datalab/every-guest-jon-stewart-ever-had-on-the-daily-show/.

```
Classes 'tbl_df', 'tbl' and 'data.frame':2693 obs. of  5 variables:
 $ year                       : int  1999 1999 1999 1999 1999 1999 1999 1999 1999 1999 ...
 $ google_knowledge_occupation: chr  "actor" "comedian" "television actress" "film actress" ...
 $ show                        : Date, format: "1999-01-11" "1999-01-12" "1999-01-13" "1999-01-14"
   ...
 $ group                      : chr  "Acting" "Comedy" "Acting" "Acting" ...
 $ raw_guest_list              : chr  "Michael J. Fox" "Sandra Bernhard" "Tracey Ullman" "Gillian
   Anderson" ...
```

fivethirtyeight – drinks

The raw data behind the story "Dear Mona Followup: Where Do People Drink The Most Beer, Wine And Spirits?" http://fivethirtyeight.com/datalab/dear-mona-followup-where-do-people-drink-the-most-beer-wine-and-spirits/.

```
Classes 'tbl_df', 'tbl' and 'data.frame':193 obs. of  5 variables:
 $ country                     : chr  "Afghanistan" "Albania" "Algeria" "Andorra" ...
 $ beer_servings               : int  0 89 25 245 217 102 193 21 261 279 ...
 $ spirit_servings             : int  0 132 0 138 57 128 25 179 72 75 ...
 $ wine_servings               : int  0 54 14 312 45 45 221 11 212 191 ...
 $ total_litres_of_pure_alcohol: num  0 4.9 0.7 12.4 5.9 4.9 8.3 3.8 10.4 9.7 ...
```

fivethirtyeight – drug_use

The raw data behind the story "How Baby Boomers Get High"http://fivethirtyeight.com/datalab/how-baby-boomers-get-high/. It covers usage of 13 drugs in the past 12 months across 17 age groups.

```
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':17 obs. of  28 variables:
 $ age               : Factor w/ 17 levels "12","13","14",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ n                 : int  2798 2757 2792 2956 3058 3038 2469 2223 2271 2354 ...
 $ alcohol_use       : num  3.9 8.5 18.1 29.2 40.1 49.3 58.7 64.6 69.7 83.2 ...
 $ alcohol_freq      : num  3 6 5 6 10 13 24 36 48 52 ...
 $ marijuana_use     : num  1.1 3.4 8.7 14.5 22.5 28 33.7 33.4 34 33 ...
 $ marijuana_freq    : num  4 15 24 25 30 36 52 60 60 52 ...
 $ cocaine_use       : num  0.1 0.1 0.1 0.5 1 2 3.2 4.1 4.9 4.8 ...
 $ cocaine_freq      : num  5 1 5.5 4 7 5 5 5.5 8 5 ...
 $ crack_use         : num  0 0 0 0.1 0 0.1 0.4 0.5 0.6 0.5 ...
 $ crack_freq        : num  NA 3 NA 9.5 1 21 10 2 5 17 ...
 $ heroin_use        : num  0.1 0 0.1 0.2 0.1 0.1 0.4 0.5 0.9 0.6 ...
 $ heroin_freq       : num  35.5 NA 2 1 66.5 64 46 180 45 30 ...
 $ hallucinogen_use  : num  0.2 0.6 1.6 2.1 3.4 4.8 7 8.6 7.4 6.3 ...
 $ hallucinogen_freq : num  52 6 3 4 3 3 4 3 2 4 ...
 $ inhalant_use      : num  1.6 2.5 2.6 2.5 3 2 1.8 1.4 1.5 1.4 ...
 $ inhalant_freq     : num  19 12 5 5.5 3 4 4 3 4 2 ...
 $ pain_releiver_use : num  2 2.4 3.9 5.5 6.2 8.5 9.2 9.4 10 9 ...
 $ pain_releiver_freq: num  36 14 12 10 7 9 12 12 10 15 ...
 $ oxycontin_use     : num  0.1 0.1 0.4 0.8 1.1 1.4 1.7 1.5 1.7 1.3 ...
 $ oxycontin_freq    : num  24.5 41 4.5 3 4 6 7 7.5 12 13.5 ...
 $ tranquilizer_use  : num  0.2 0.3 0.9 2 2.4 3.5 4.9 4.2 5.4 3.9 ...
 $ tranquilizer_freq : num  52 25.5 5 4.5 11 7 12 4.5 10 7 ...
 $ stimulant_use     : num  0.2 0.3 0.8 1.5 1.8 2.8 3 3.3 4 4.1 ...
 $ stimulant_freq    : num  2 4 12 6 9.5 9 8 6 12 10 ...
 $ meth_use          : num  0 0.1 0.1 0.3 0.3 0.6 0.5 0.4 0.9 0.6 ...
 $ meth_freq         : num  NA 5 24 10.5 36 48 12 105 12 2 ...
 $ sedative_use      : num  0.2 0.1 0.2 0.4 0.2 0.5 0.4 0.3 0.5 0.3 ...
 $ sedative_freq     : num  13 19 16.5 30 3 6.5 10 6 4 9 ...
```

fivethirtyeight – flying

The raw data behind the story "41 Percent Of Fliers Think You're Rude If You Recline Your Seat" http://fivethirtyeight.com/datalab/airplane-etiquette-recline-seat.

```
Classes 'tbl_df', 'tbl' and 'data.frame':1040 obs. of  27 variables:
 $ respondent_id       : num  3.44e+09 3.43e+09 3.43e+09 3.43e+09 3.43e+09 ...
 $ gender              : chr  NA "Male" "Male" "Male" ...
 $ age                 : Factor w/ 4 levels "18-29","30-44",..: NA 2 2 2 2 2 2 2 NA 2 ...
 $ height              : Factor w/ 20 levels "Under 5 ft.",..: NA 17 10 13 9 11 16 14 14 8 ...
 $ children_under_18   : logi  NA TRUE FALSE FALSE FALSE TRUE ...
 $ household_income    : Factor w/ 5 levels "$0 - $24,999",..: NA NA 4 1 3 2 NA 1 NA 1 ...
 $ education           : Factor w/ 5 levels "Less than high school degree",..: NA 5 4 4 4 5 3 4 NA
   4 ...
 $ location            : chr  NA "Pacific" "Pacific" "Pacific" ...
 $ frequency           : Factor w/ 6 levels "Never","Once a year or less",..: 2 2 2 2 3 2 3 2 2 2
   ...
 $ recline_frequency   : Factor w/ 5 levels "Never","Once in a while",..: NA 3 4 5 3 4 2 2 2 1 ...
 $ recline_obligation  : logi  NA TRUE TRUE FALSE FALSE TRUE ...
 $ recline_rude        : Factor w/ 3 levels "No","Somewhat",..: NA 2 1 1 1 1 2 1 1 3 ...
 $ recline_eliminate   : logi  NA FALSE FALSE FALSE FALSE FALSE ...
 $ switch_seats_friends: Factor w/ 3 levels "No","Somewhat",..: NA 1 1 2 1 2 2 1 NA 3 ...
 $ switch_seats_family : Factor w/ 3 levels "No","Somewhat",..: NA 1 1 1 1 1 1 1 NA 3 ...
 $ wake_up_bathroom    : Factor w/ 3 levels "No","Somewhat",..: NA 1 1 1 2 2 1 1 NA 3 ...
 $ wake_up_walk        : Factor w/ 3 levels "No","Somewhat",..: NA 1 2 2 2 3 1 2 NA 3 ...
 $ baby                : Factor w/ 3 levels "No","Somewhat",..: NA 1 2 2 2 3 1 2 NA 3 ...
 $ unruly_child        : Factor w/ 3 levels "No","Somewhat",..: NA 1 3 3 3 3 2 3 NA 3 ...
 $ two_arm_rests       : chr  NA "The arm rests should be shared" "Whoever puts their arm on the
   arm rest first" "The arm rests should be shared" ...
 $ middle_arm_rest     : chr  NA "The arm rests should be shared" "The arm rests should be shared"
   "The arm rests should be shared" ...
 $ shade               : chr  NA "Everyone in the row should have some say" "The person in the
   window seat should have exclusive control" "Everyone in the row should have some say" ...
 $ unsold_seat         : Factor w/ 3 levels "No","Somewhat",..: NA 1 1 1 1 2 1 1 1 3 ...
 $ talk_stranger       : Factor w/ 3 levels "No","Somewhat",..: NA 1 1 1 1 1 2 1 1 3 ...
 $ get_up              : Factor w/ 6 levels "It is not okay to get up during flight",..: NA 3 4 4
   3 2 3 5 5 1 ...
 $ electronics         : logi  NA FALSE FALSE FALSE TRUE FALSE ...
 $ smoked              : logi  NA FALSE FALSE FALSE FALSE FALSE ...
```

fivethirtyeight – police_locals

The raw data behind the story "Most Police Don't Live In The Cities They Serve"http://fivethirtyeight.com/datalab/most-police-dont-live-in-the-cities-they-serve/.

```
Classes 'tbl_df', 'tbl' and 'data.frame':75 obs. of  8 variables:
 $ city      : chr  "New York" "Chicago" "Los Angeles" "Washington" ...
 $ force_size: int  32300 12120 10100 9340 7700 6045 4475 4460 3605 3265 ...
 $ all       : num  0.618 0.875 0.228 0.116 0.292 ...
 $ white     : num  0.4464 0.872 0.1528 0.0568 0.1737 ...
 $ non_white : num  0.764 0.877 0.264 0.157 0.399 ...
 $ black     : num  0.771 0.897 0.387 0.17 0.366 ...
 $ hispanic  : num  0.7629 0.8398 0.2177 0.0899 0.4571 ...
 $ asian     : num  0.749 0.967 0.305 0.231 0.408 ...
```