R is a very flexible tool and there are a lot of ways to accomplish the same thing. What I am showing you here is just one way to do each test while using R as little as possible. If you aren't confident with R, you can follow the method here exactly (manipulating your data outside of R to get it into the format shown here). If you are confident with R, you can easily find many other ways to accomplish these same things.

These are extremely simple and aren't taking advantage of many of the capabilities of R; I am just providing these examples so you can use them in this class if you need to run t-tests for your activities. I don't recommend using these examples directly beyond this class if you don't understand the code. In general, I do not recommend using any code you don't understand. If you want to use R in your actual research, make sure you take the time to learn it (the "Programming" module in this subject will be a good start), rather than just copying code that you don't fully understand.

Installing the ez package

For this tutorial I am going to use the R function `ezAnova()`, which is not part of the base R installation; it's part of an extra package you need to install.

1. Make sure you have the latest version of R (it's possible to install the package on an older version, but it's more complicated). To do at, either go to the R website and download and install the newest version; or, open R and run

the following commands (just copy and paste them into the R command window), which will automatically update you to the latest version (you will have to click on some pop-up questions that come up:

```
install.packages("installr")
updateR()
```

2. Install the "ez" package, using the following command:

   ```
   install.packages("ez")
   ```

3. Load the package using the following command:

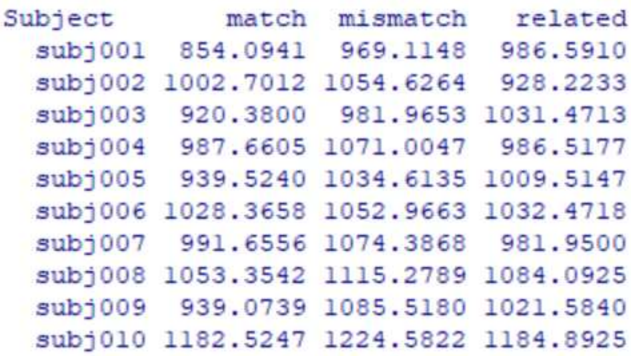
   ```
   library(ez)
   ```

You will only need to do steps 1 and 2 once on a given computer. Step 3 you need to do every time you open R to perform an ANOVA.

Understanding long vs. wide format

For this example I'm going to be using repeated-measures (i.e., paired, dependent) data; the method shown here will also work for independent data.

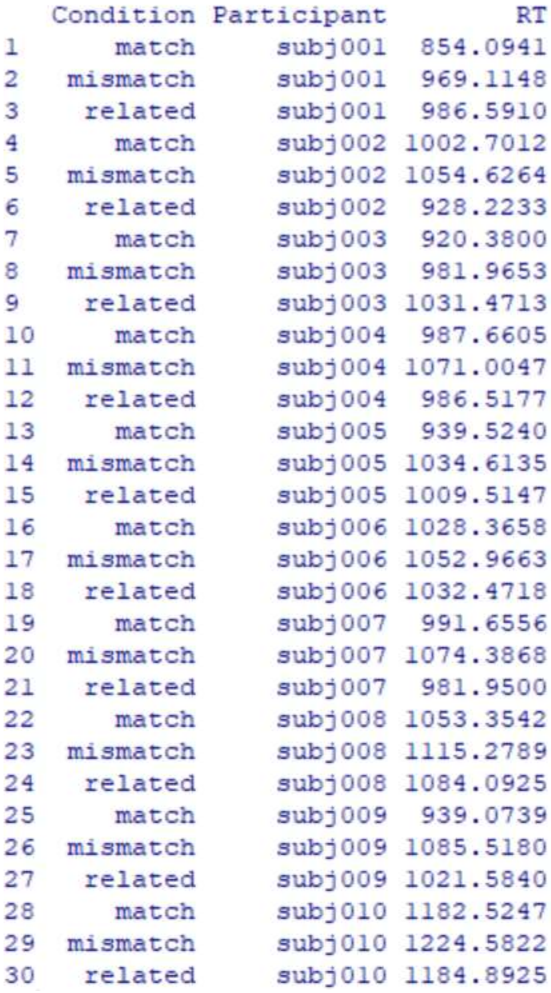
There are two different ways of organizing repeated measures data. One is to make each row be the data of one participant, and each column be their data in a different condition. (This is the typical way to organize data for an ANOVA in SPSS, and when doing data analysis in Excel this format is also convenient.) It looks like this:



```
Subject        match  mismatch   related
  subj001  854.0941  969.1148  986.5910
  subj002 1002.7012 1054.6264  928.2233
  subj003  920.3800  981.9653 1031.4713
  subj004  987.6605 1071.0047  986.5177
  subj005  939.5240 1034.6135 1009.5147
  subj006 1028.3658 1052.9663 1032.4718
  subj007  991.6556 1074.3868  981.9500
  subj008 1053.3542 1115.2789 1084.0925
  subj009  939.0739 1085.5180 1021.5840
  subj010 1182.5247 1224.5822 1184.8925
```

You can see from the above screenshot that each row is data from one participant, and the columns represent data from different conditions (in this experiment participants responded to words in three conditions—"match", "mismatch", and "related"—and I measured how quickly they responded to words in each condition). This is called *wide format*, because it's wide (you can imagine that if there were a lot more conditions and independent variables in this experiment—and thus a lot more columns—this would get very wide).

In R, however, most things are optimized to work with *long format* data. In long format, each column represents one *variable*. (That's different than long-format data; in the long-format example above, the last three columns all represent the dependent variable [how quickly a person responded to words], just in three different conditions.) The same dataset as above, represented in long format rather than wide format, looks like this:



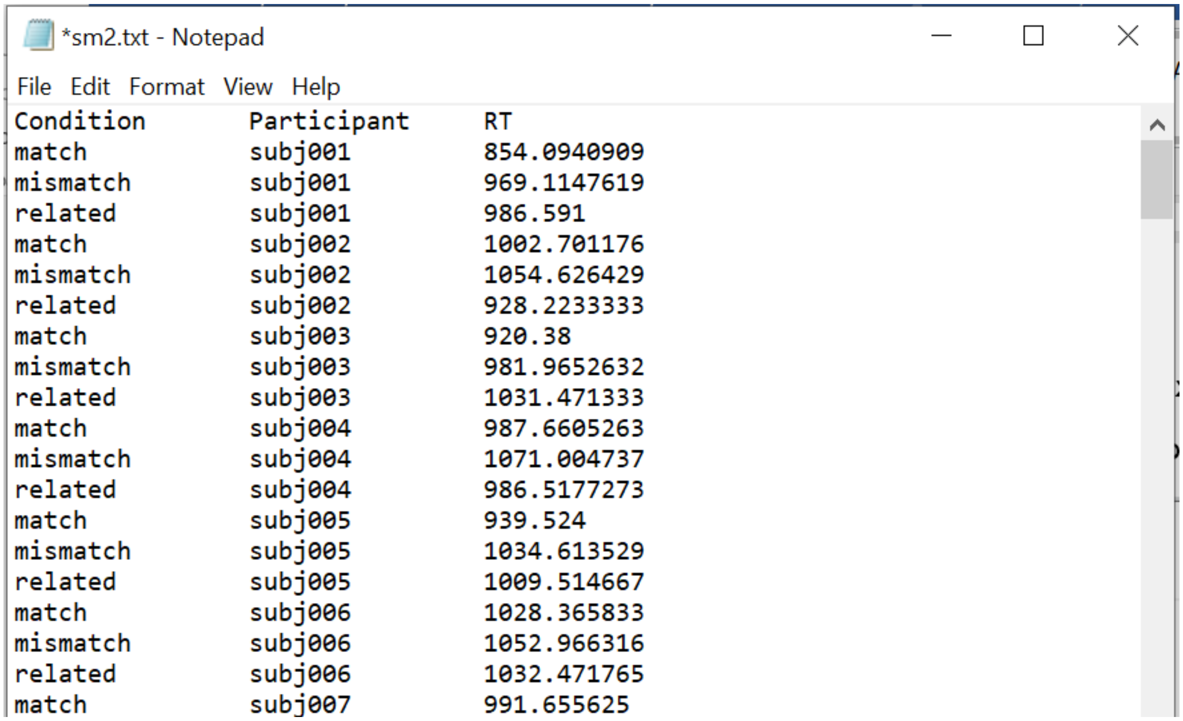
```
   Condition Participant        RT
1      match     subj001  854.0941
2   mismatch     subj001  969.1148
3    related     subj001  986.5910
4      match     subj002 1002.7012
5   mismatch     subj002 1054.6264
6    related     subj002  928.2233
7      match     subj003  920.3800
8   mismatch     subj003  981.9653
9    related     subj003 1031.4713
10     match     subj004  987.6605
11  mismatch     subj004 1071.0047
12   related     subj004  986.5177
13     match     subj005  939.5240
14  mismatch     subj005 1034.6135
15   related     subj005 1009.5147
16     match     subj006 1028.3658
17  mismatch     subj006 1052.9663
18   related     subj006 1032.4718
19     match     subj007  991.6556
20  mismatch     subj007 1074.3868
21   related     subj007  981.9500
22     match     subj008 1053.3542
23  mismatch     subj008 1115.2789
24   related     subj008 1084.0925
25     match     subj009  939.0739
26  mismatch     subj009 1085.5180
27   related     subj009 1021.5840
28     match     subj010 1182.5247
29  mismatch     subj010 1224.5822
30   related     subj010 1184.8925
```

In this format, you can see that each participant's data occupies three rows (because each participant has data in three conditions). However, you can look at the numbers (the reaction times) between the wide-format and long-format examples and confirm that these are the exact same data, just organized in two different ways.

To use the R function described in this tutorial, your data must be in ***long format***. If your data were in wide format, make sure you rearrange them into long format before you continue.

Getting your data into R

Save your data in a tab-delimited text file in multiple columns with labels at the top, which will look something like this:



```
Condition       Participant     RT
match           subj001         854.0940909
mismatch        subj001         969.1147619
related         subj001         986.591
match           subj002         1002.701176
mismatch        subj002         1054.626429
related         subj002         928.2233333
match           subj003         920.38
mismatch        subj003         981.9652632
related         subj003         1031.471333
match           subj004         987.6605263
mismatch        subj004         1071.004737
related         subj004         986.5177273
match           subj005         939.524
mismatch        subj005         1034.613529
related         subj005         1009.514667
match           subj006         1028.365833
mismatch        subj006         1052.966316
related         subj006         1032.471765
match           subj007         991.655625
```

Note: even if your experiment does not have repeated measures (i.e., if every participant only contributes one data point to the experiment, unlike this example where each participant contributes three data points), you still need to include a "Participant" column with a label for each participant. The function we're using in this tutorial requires it.

Open R, and import your data using the following command (you will have to manually navigate to the location where the file is saved):

```
mydata <- read.table( file=file.choose(), header=T, sep="\t" )
```

(the `file=file.choose()` parameter causes R to open a graphical interface where you choose your file; the `header=T` parameter tells R that the first row of your data file is a "header" consisting of labels for the two conditions; and the `sep="\t"` parameter says that the numbers in your text file are separated by tabs).

Running the ANOVA

To run the ANOVA, you issue some version of the following command, but you need to fill in a lot of extra information (which will be described below):

```
ezANOVA(
    mydata
    , RT
    , Participant
    , within = ???
    , between = ???
)
```

Where I wrote "mydata", this should be the name of the dataset you imported (if you copied my code exactly in the previous section, this will still be called "mydata").

Where I wrote "RT", this should be the name of the column that you are using as the dependent variable.

The within= and between= parts are where you specify your independent variables. Recall that ANOVA can have within-participant (i.e., dependent/paired/repeated-measures) variables and it can have between-participant (i.e., independent) variables. A single ANOVA can have just one kind of variable (my example from above only has a within-participants variable and no between-participants variables), or might have both in the same analysis.

If you don't have any within-participant independent variables, you can delete the line ", within = ???".

If you don't have any between-participant independent variables, you can delete the line ", between = ???".

If you have one within-participant independent variable, replace the ??? with the name of that variable (i.e., the column name from your dataset). In my example, this is "Condition".

Likewise for if you have one between-participant independent variable.

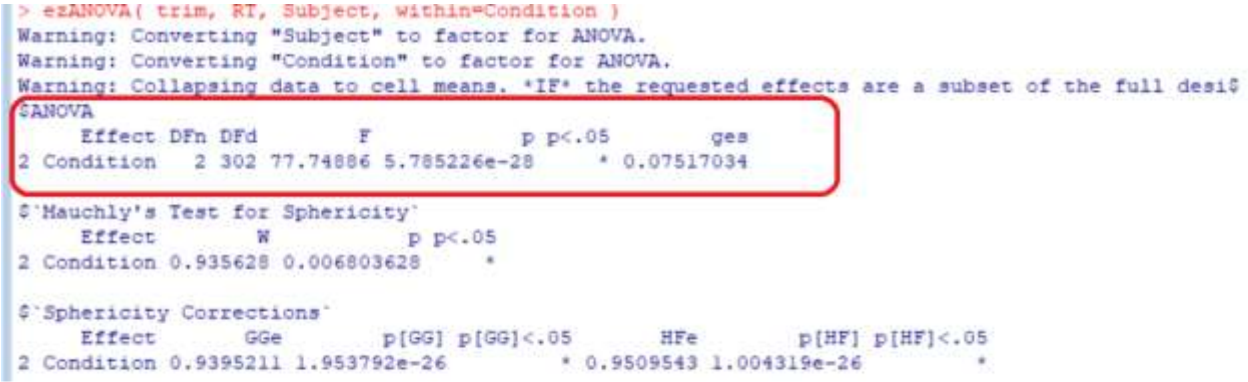
If you have multiple within-participant independent variables (e.g., if you're doing a factorial design), then you have to put their names in a list, separated by commas, within ".()". For example, if I had one variable called Condition and another variable called Test, I would write this as: `within = .(Condition,Test)`

Likewise for if you have multiple between-participant independent variables.

So, if I want to run a one-way ANOVA on the example set of data I showed above, I would write this in R:

```
ezANOVA(
    mydata
    , RT
    , Participant
    , within = Condition
)
```

After you press enter, you will see a lot of results. The only one you need to pay attention to for this module is the first row of effects (circled in the screenshot below). The other stuff is also potentially important, but beyond the scope of what we have time to learn in this class. Here I will repeat the recommendation I made in the t-test module: if you plan to use ANOVA in your real research, make sure you either take more time to study it in detail (when I was a PhD student I took a full semester-long class *just* about ANOVA, whereas in our subject it's crammed into one module of less than a week), or collaborate with someone who knows what they're doing.



```
> ezANOVA( trim, RT, Subject, within=Condition )
Warning: Converting "Subject" to factor for ANOVA.
Warning: Converting "Condition" to factor for ANOVA.
Warning: Collapsing data to cell means. *IF* the requested effects are a subset of the full desi$
$ANOVA
     Effect DFn DFd        F            p p<.05        ges
2 Condition   2 302 77.74886 5.785226e-28     * 0.07517034

$`Mauchly's Test for Sphericity`
     Effect        W           p p<.05
2 Condition 0.935628 0.006803628     *

$`Sphericity Corrections`
     Effect       GGe        p[GG] p[GG]<.05       HFe        p[HF] p[HF]<.05
2 Condition 0.9395211 1.953792e-26         * 0.9509543 1.004319e-26         *
```

In this example there's only one effect, because I only had one independent variable. Of course if you do ANOVA on a factorial design you will see lots more effects, because you'll have multiple variables as well as their interactions.

When reporting ANOVA results, the important stuff is DFn, DFd, F, and *p*. As this is not a statistics class, we won't get into the details of what all that means or how it is typically reported (you can look at papers in your field to see what these reports normally look like). For this module, all you need to look at is *p*.

Be aware that R uses scientific notation for very large or very small numbers. In this example, the *p* value is written as "5.785226e-28". This is an extremely small number (5.785×10^{-28}, i.e., .000000000000000000000000000005785).

Doing ANOVA with more complicated datasets (e.g., factorial designs) follows the same principles as those outlined above.