When to use t.test(A, B)

If the data are in two separate data frames, where the group membership is defined by which data frame the observation is in, then it's best to use the comma syntax. For example, if you had 1 data frame called Dog_df that looked like

Name.of.pet	DogWeight
Spot	15
Fido	12
Sparky	17

and another data frame called Cat_df that looked like

Name.of.pet	CatWeight
Socks	12
Fluffy	14

then you would use

```
t.test(Dog_df$DogWeight, Cat_df$CatWeight)
```

to test whether the average weight of dogs was different from the average weight of cats.

One-sided, two-sample t-tests

In some cases, you may only be interested in testing whether the average weight of dogs is *higher* than the average weight of cats.

- Maybe you're very confident that dogs are not lighter than cats (they're either the same weight, or dogs are heavier).
    - Note that this should be based on prior experience with dogs and cats, *not* based on looking at the sample means in the current data set.
- Or maybe, if it turned out that cats are heavier than dogs, you would take the same action as if cats and dogs were the same weight. The only result that would make a difference in your behavior is if dogs are heavier than cats.
    - Example: You want to take all of the dogs from your local animal shelter on a boat trip. If the average weight of dogs is the same as, or less than, the average weight of cats, then you can use the same boat as last year, when you took all of the cats from your local animal shelter on a boat trip. If the average weight of dogs is greater than the average weight of cats, then you will need to find a bigger boat.

In situations like this, you can improve your chance of getting a small p-value (your statistical power) using a one-sided t-test. (Note that it's still a two-sample t-test.)

To do this, the approaches

```
t.test(Cat_df$CatWeight, Dog_df$DogWeight, alternative = "less")
t.test(Dog_df$DogWeight, Cat_df$CatWeight, alternative = "greater")
```

are equivalent. With the first one, you would be more likely to phrase the conclusion as "There is/is not enough evidence to claim that the mean weight of cats is less than the mean weight of dogs," while with

the second one, you would be more likely to phrase the conclusion as "There is/is not enough evidence to claim that the mean weight of dogs is greater than the mean weight of cats." But the p-value will be the same either way, and those two ways of writing the conclusion are equivalent.

Note that this only applies when A and B are the two groups you want to compare, and are separated by commas inside the t.test(). If, instead, A contains the data from both groups and B is a vector that tells which group each observation belongs to, and you're using

```
t.test(A ~ B, alternative = "less")
```

then A and B are *not* interchangeable.

- alternative = "greater" is saying that the alternative hypothesis is that the population represented by whichever sample is listed as the first argument has a higher mean than the population represented by the sample in the second argument. So, it doesn't matter whether you list the control or treatment group first, as long as you're thinking about which one you hypothesize will have a higher mean, and using alternative = "greater" or alternative = "less", as appropriate. (In fact, in some analyses it may not make sense to talk about a "control" versus "treatment" group.)

When to use t.test(A ~ B)

When the data are organized as two columns of the same data frame, with 1 column being the quantitative variable (whose mean you're testing) and the other column being the binary variable (that tells which group each observation belongs to), it's best to use the "~" syntax to run the t-test. For example, if the data set was

Name.of.pet	Species	Weight
Spot	Dog	15
Fido	Dog	12
Socks	Cat	12
Fluffy	Cat	14
Sparky	Dog	17

then you would use

```
t.test(Weight ~ Species)
```

to do the *same* test of whether the average weight of dogs was different from the average weight of cats.

One-sided, two-sample tests with ~ syntax

When you're doing a 1-sided t-test with the "~" syntax, you need to be especially careful about the direction of your alternative hypothesis. That's because R will choose an "order" of the two groups for you; by default it chooses alphabetically (so Cat would be the first group and Dog would be the second group). (You can change the order if you want: http://www.cookbook-r.com/Manipulating_data/Changing_the_order_of_levels_of_a_factor/)

So, if you wanted to test whether the average weight of cats is **less than** the average weight of dogs, you would use

```
t.test(Weight ~ Species, alternative = "less")
```

because the alternative hypothesis is that the mean of the first group is less than the mean of the second group.

But, let's say you wanted to do the same test (that cats weigh less than dogs), but now the Species of cats in your data frame is "Feline" instead of "Cat". Now, R would treat Felines as the second group and Dogs as the first group, so your alternative hypothesis would be that
the average weight of dogs is **greater than** the average weight of felines.

To test this, you would use

```
t.test(Weight ~ Species, alternative = "greater")
```

Frequently, I do a test run of

```
t.test(Data ~ Grouping, alternative = "greater")
```

and then check the last 2 lines of the output to see which order R has put the levels in. For example, if I generate some simple data like this:

```
x = rep(c("A","B"), each=10)
y = c(rnorm(10),rnorm(10,1,1))
t.test(y ~ x, alternative = "greater")
```

then the output is

```
 Welch Two Sample t-test


data:  y by x
t = -1.9679, df = 17.902, p-value = 0.9676
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -2.061499       Inf
sample estimates:
mean in group A mean in group B
    -0.03711146      1.05858117
```

Looking at the last 2 lines, I see that the order of the groups is "A" first, then "B". Also, the sample mean of group A was -.037, which is less than the sample mean of group B, which was 1.059. Based on this, I know that if my original plan was to test whether B was greater than A, I had better use

```
t.test(y ~ x, alternative = "less")
```

instead.

- If you’re doing a 1-sided test and get a p-value close to 1, it's always a good idea to double-check that your alternative hypothesis is specified in the correct direction.