ENTMLGY 6707 Entomological Techniques and Data Analysis

R activity 4: *t*-tests

1 *t*-tests

We use *t*-tests to compare a continuous variable (e.g., yield, tree diameter at breast height) between two groups (e.g., no fertilizer vs. fertilizer, tree species X vs. tree species Y).

1.1 Assumptions

Every quantitative analysis requires us to make assumptions about our data. If we are to trust the results of an analysis, we need to be reasonably sure that its assumptions are met. For a *t*-test, we assume the:

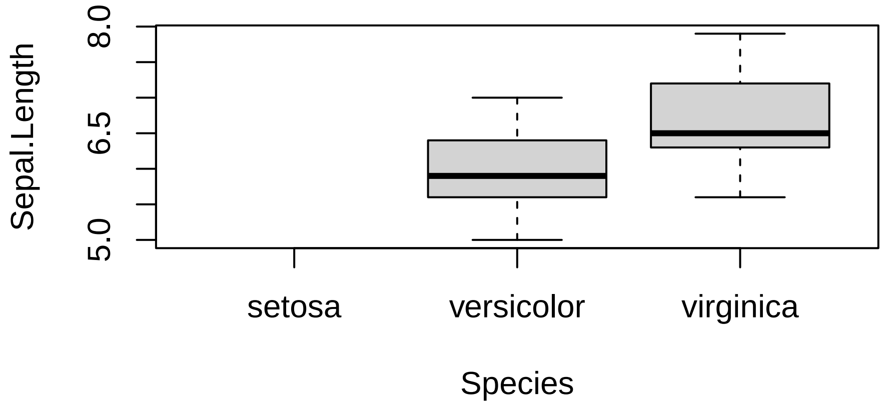
1. Response (aka dependent) variable (the one we are comparing between groups) is continuous
2. Observations are independent (typically meaning they comprise a random sample)
3. Response variable is normally distributed
4. Variances of the response variable are equal across groups (= homogeneity of variances)

We will again use the `iris` data for this tutorial. First, we need to load in the data. We have been doing this manually, but this week we will just load it in directly using the `datasets` package. That is, R has a package that loads in this famous data set (and many others) for us.

```
library(datasets)
library(tidyverse)
```

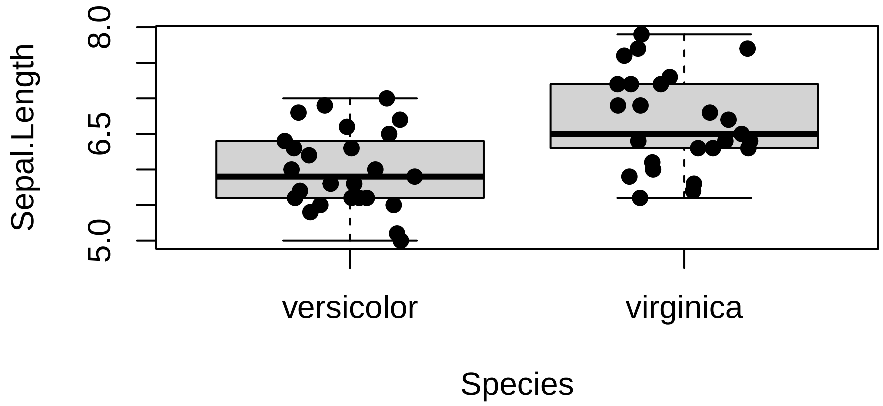
Since we only want to work with two groups for a *t*-test, let's subset the data by removing one of the `Species` (pretend this species was colonized by a pathogen and the anatomical measurements are no longer reliable, so we exclude it from our analysis). Notice this code does not fully achieve our goal... R still remembers we had a level of `setosa` in our data.

```
iris_t <- iris %>% filter(Species == c("versicolor", "virginica"))
boxplot(Sepal.Length~Species, data=iris_t)
```



Adding the `droplevels()` argument to the pipeline essentially erases the memory of `setosa` (the species we removed) from the data set. We have overlaid the data points onto the boxplot, and jittered them slightly so they don't overlap. This provides additional information about the spread of the data.

```
iris_t <- iris %>% filter(Species == c("versicolor", "virginica")) %>%  droplevels()
boxplot(Sepal.Length~Species, data=iris_t)
stripchart(Sepal.Length~Species, data=iris_t, pch = 19, add = TRUE, vertical = TRUE,
           method = "jitter", jitter = 0.2)
```



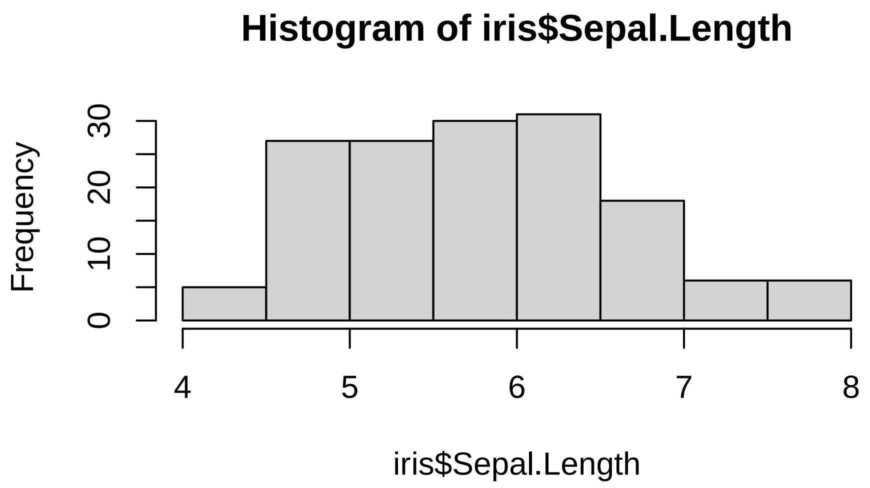
1.2 Checking assumptions 1 (response is continuous) and 2 (data are independent)

We want to compare `Sepal.Length` between `Species`. You can assess the first assumption by knowing the form of your response variable (i.e., we are hopeful you now feel confident determining if your data are continuous vs. categorical) and assess the second assumption by understanding the experimental design and data collection protocols. Thus, you typically know if the first and second assumptions are met before you even start collecting the data. We did not collect the `iris` data, but we know the first assumption is met and we will assume the second is as well.

1.3 Checking assumption 3 (normality)

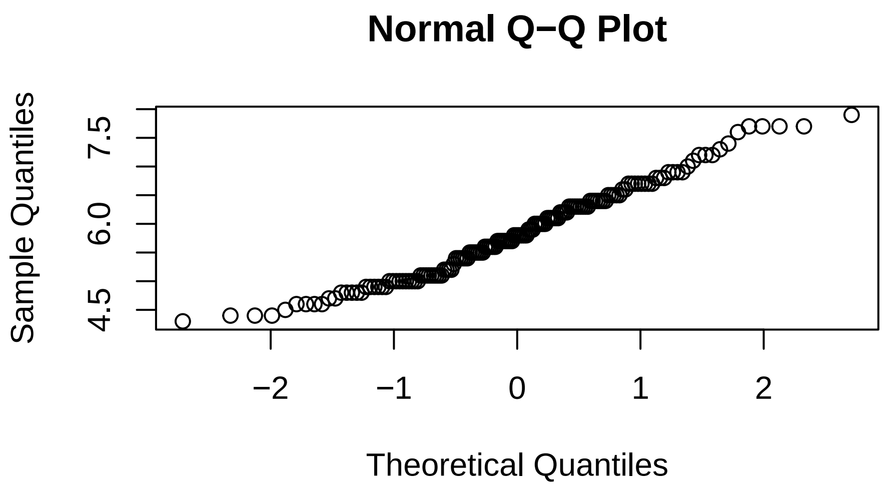
There are formal tests to assess normality (e.g., Shapiro-Wilk Test), but these tend to be quite sensitive. We generally rely on graphical inspection using a histogram

```
hist(iris$Sepal.Length)
```



and/or a Quantile-Quantile (Q-Q) plot. See the companion document "Q-Q Plots: Nuts and Bolts" for a detailed explanation of Q-Q plots. In short, you want the data to fall on a straight line.

```
qqnorm(iris$Sepal.Length)
```



1.4 Checking assumption 4 (homogeneity of variance)

We can use a boxplot (like the one we created earlier in this document) to check the assumption of equal variances. We are proponents of graphical inspection of data for this assumption, but some folks may disagree and want you to use a formal test (e.g., Levene's Test). You could also try comparing the variances directly using a ratio. There are rules of thumb floating around (e.g., a ratio of variances within 0.5-2 is okay).

```r
var_check <- iris_t %>% group_by(Species) %>% summarise(Variance = var(Sepal.Length))
var_check$Variance[2]/var_check$Variance[1]# variance ratio
```

```
## [1] 1.446457
```

If you do have an issue with homogeneity of variances, square-root or log-transformations can often help. Here, the variances look pretty equal and, in practice, we would move on to the *t*-test at this stage.

1.5 Conducting the *t*-test

We will conduct the test using `t.test()`. Here is a good time to check `?t.test()` to make sure the default arguments reflect your goals - for example, are you conducting a two-sided or one-sided *t*-test? Are the observations paired? There is also an argument to specify whether the variances are equal, meaning there are statistical adjustments for *t*-tests when that assumption is violated. Said another way, if that assumption is violated and a transformation did not help, you can specify `var.equal = F` and conduct the test.

```
t.test(Sepal.Length~Species, data=iris_t,
       alternative = "two.sided", paired=F, var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  Sepal.Length by Species
## t = -3.9099, df = 48, p-value = 0.0002896
## alternative hypothesis: true difference in means between group versicolor and group virginica is not
## 95 percent confidence interval:
##  -1.0296886 -0.3303114
## sample estimates:
## mean in group versicolor  mean in group virginica
##                    5.992                    6.672
```

Here is an example write-up: Sepal lengths (cm ± SE) of *Iris versicolor* (5.99 ± 0.11) were significantly shorter than those of *I. virginica* (6.67 ± 0.13) by an average of 0.68 cm (t_{48} = -3.91, p < 0.001).

Notice the test statistic has two decimal places and the *p*-value has **three**. For non-significant *p*-values, two decimal places are typically reported.

When writing up results for *t*-tests, we prefer to report the raw means and standard errors (rounded to two significant digits), which can be calculated using the code below. R does not have a built in standard error function, so we use one from the `plotrix` package.

```
library(plotrix)
```

```
iris_t %>%
  group_by(Species) %>%
  summarise(means = round(mean(Sepal.Length),2), SE = round(std.error(Sepal.Length),2))
```

```
## # A tibble: 2 x 3
##   Species    means    SE
##   <fct>      <dbl> <dbl>
## 1 versicolor  5.99  0.11
## 2 virginica   6.67  0.13
```

2 R Activity

The response variable is the length (picometre, pm) of odontoblasts (cells responsible for tooth growth) in 60 guinea pigs. There is one measurement per pig and each animal received a low (<1 mg/day) or high (≥ 1) dose of vitamin C. There was another treatment in these data, but we are ignoring it for this activity.

1. Download the `pig_teeth` data set and load it into R. Name the dataset `pigs` in R. Convert the `dose` column to a factor using `pigs$dose <- as.factor(pigs$dose)` and provide a `summary()` of the resulting data set.

2. We are going to compare `len` (length of the cells) by dose levels. First, we need to graph the data (if possible, always always always graph your data before analyzing it!). Make a boxplot with `ggplot2` with `Length of cells` as the y-axis and `Dose` on the x-axis. Use the theme `theme_classic()`.

3. Look at the graph you just created. Do the variances look equal? If not, what might you do to fix the issue? Please write a response - we are not asking for an example using R code for this question.

4. That leaves one assumption left: normality. Check the normality of `len` using a method of your choosing. Is this assumption met? Write one sentence explaining your reasoning.

5. Conduct a two-sided *t*-test comparing `len` between your two levels of `dose`. Set `var.equal=T` in the *t*-test function (we are pretending the assumptions are met here, but *t*-tests are actually pretty robust to violations of normality).

6. Write a one sentence conclusion of your *t*-test and be sure to include means, SEs, and summary statistics.