Comparing Groups and Hypothesis Testing

We've mainly reviewed about informally comparing the distribution of data in different groups. Now we want to review the tools you know about how use statistics to make this more formal – specifically to quantify whether the differences we see are due to natural variability or something deeper.

We will first review the techniques from Data 8 in the setting of comparing two groups which you've already seen. This review is to refresh your memory, but also has the following goals to look for:

- abstract the ideas of hypothesis testing, in particular what it means to be "valid", what makes a good procedure
- dig a little deeper as to what assumptions we are making in using a particular test
- Introduce parametric ideas of hypothesis testing

The Question Recall the airline data, with different airline carriers. We could ask the question about whether the distribution of flight delays is different between carriers. If we wanted to ask whether United was more likely to have delayed flights than American Airlines, how might we quantify this?

What happens here when I take the mean of all our observations?

```
flightSubset <- flightSFOSRS[flightSFOSRS$Carrier %in%
    c("UA", "AA"), ]
mean(flightSubset$DepDelay)
```

```
## [1] NA
```

We can use a useful function 'tapply' that will do calculations by group.

```
tapply(X = flightSubset$DepDelay, flightSubset$Carrier,
    mean)
```

```
## AA UA
## NA NA
```

```
tapply(flightSubset$DepDelay, flightSubset$Carrier,
    mean, na.rm = TRUE)
```

```
##        AA        UA
##  7.728294 12.255649
```

```
tapply(flightSubset$DepDelay, flightSubset$Carrier,
    function(x) {
        mean(x, na.rm = TRUE)
    })
```

```
##        AA        UA
##  7.728294 12.255649
```

```
f <- function(x) {
    mean(x, na.rm = TRUE)
}
tapply(flightSubset$DepDelay, flightSubset$Carrier,
    FUN = f)
```

```
##        AA        UA
##  7.728294 12.255649
```

```
tapply(flightSubset$DepDelay, flightSubset$Carrier,
    function(x) {
        median(x, na.rm = TRUE)
    })
```

```
## AA UA
## -2 -1
```

```
tapply(flightSubset$DepDelay, flightSubset$Carrier,
    function(x) {
        sum(x > 0 | is.na(x))/length(x)
    })
```

```
##        AA        UA
## 0.3201220 0.4383791
```

```
tapply(flightSubset$DepDelay, flightSubset$Carrier,
    function(x) {
        sum(x > 15 | is.na(x))/length(x)
    })
```

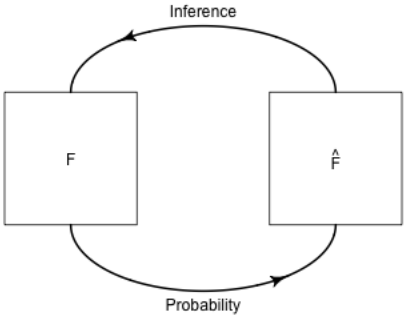
```
##        AA        UA
## 0.1554878 0.2046216
```

```
tapply(flightSubset$Cancelled, flightSubset$Carrier,
    mean)
```

```
##          AA          UA
## 0.005081301 0.007032820
```

These are **statistics** that we can calculate from the data. A statistic is *any* function of the input data sample.

Once we've decided on a statistic, we want to ask whether this is a meaningful difference between our groups. Specifically, with different data samples, the statistic would change. **Inference** is the process of using statistical tools to evaluate whether the statistic observed indicates some kind of actual difference, or whether we could see such a value due to random chance even if there was no difference.



Therefore, to use the tools of statistics – to say something about the generating process – we must have be able to define a random process that we posit created the data.

1 Hypothesis Testing

Recall the components of **hypothesis testing**. Hypothesis testing sets up a **null hypothesis** which describes a random process that could have created any differences we see in our the flight delays (as measured by our statistic). The null hypothesis makes specific the qualitative question "this difference might be just due to chance". There are a lot of ways "chance" could have created differences, and a null hypothesis makes it specific, to the point of defining the specific probability distribution that describes the distribution of the statistic if the null hypothesis of no effect was true (the **null distribution**).

How do we determine whether the statistic is too unlikely under the null distribution? We calculate the probability (under the null distribution) of getting a statistic *as extreme as we saw or more extreme* under the null hyptothesis. This is called a **p-value**.

If the observed statistic is too unlikely under the null hypothesis we can say we **reject the null hypothesis** or that we have a **statistically significant** difference.

1.1 Where did the data come from? Valid tests & Assumptions

Just because a p-value is reported, doesn't mean that you can interpret it as a p-value. You must have a **valid** test. A valid test simply means that the p-value (or level) that you report is accurate. This is only true if the null distribution of the test statistic is correctly identified. To use the tools of statistics, we must assume some kind of random process created the data. When your data violates the assumptions of the data generating process, your p-value can be quite wrong. Sometimes we can know these assumptions are true, but often not; knowing where your data came from and how it is collected is critical for assessing these questions. So we need to always think deeply about where the data come from, how they were collected, etc.

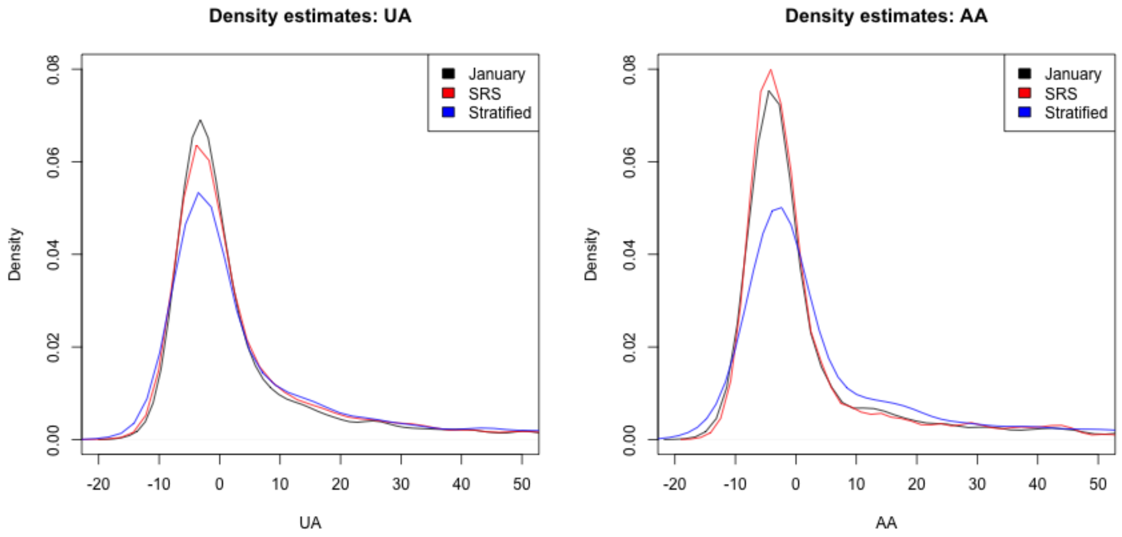
Complete Census For example, for the airline data, we have one dataset that gives *complete* information about the month of January. We can ask questions about flights in January, and get the answer by calculating the relevant statistics. For example, if we want to know whether the average flight is more delayed on United than American, we calculate the means of both groups and simply compare them. End of story. We don't need the inference tools from above

Types of Samples For most of statistical applications, this is not the case. We have a *sample* of the entire population, and want to make statements about the entire population which we don't see. Notice that having a sample does not necessarily mean a random sample. For example, we have all of January which is a sample from the entire year, and there is no randomness involved in how we selected the data from the larger population. Some datasets might be a sample of the population with no easy way to describe the relationship between the sample and the population, for example data from volunteers or other *convenience samples* that pick the easiest to get data rather than randomly sampling from the population. Having such data can make it quite fraught to try to make any conclusions about the population from the sample.

What problems do you have in trying to use the flight data on January to estimate something about the entire year?

What would be a better way to get flight data?

We discussed this issue for estimating histograms, where our histogram is a good estimate of the population when our data is a SRS, and otherwise may be quite off base. For example, here is the difference in our density estimates for three different kinds of sampling:



Recall there, that we said there we could find good estimates for other kind of random samples, though beyond the reach of this course. The key ingredient that is needed is to know the probability mechanism that drew the samples. This is the key difference between a random sample (of any kind) and a sample of convenience.

Assumptions versus reality A prominent statistician, George Box, gave the following famous quote,

All models are wrong but some are useful

All tests have assumptions, and most are often not met in practice. This is a continual problem in interpreting the results of statistical methods. Therefore there is a great deal of interest in understanding how badly the tests perform if the assumptions are violated; this is often called being **robust** to violations. We will try to emphasize both what the assumptions are, and how bad violations to the assumptions are.

For example, in practice, much of data that is available is not a random sample of the population, and therefore a sample of convenience in some sense (there's a reason we call them convenient!). Our goal is not to make say that impossible, but make clear why this might make you want to be cautious about over-interpreting the results.

2 Permutation Tests

The first statistical testing procedure you learned in Data 8 was permutation tests. Suppose we want to compare the median delay time of United and American airlines.

The distribution for the test statistic under the null hypothesis for a permutation tests is determined by assuming

1. There is no difference between the distribution of the flight delays between the two airlines,
   $$H_0 : F_{United} = F_{American}$$
   This is often written as the null hypothesis (H_0) – the main point of the null to be tested
2. The statistic observed is the result of randomly assigning the labels amongst the observed data This is the additional assumption about the random process that allows for calculating a precise null distribution of the statistic

The permutation test uses both of these assumptions to define "by chance" by assuming the data we saw we would have seen anyway even if we changed the labels (i.e. United or American). Therefore, any difference we might see between the groups is due to the luck of random permutation of the labels.

2.1 How do we implement it?

This is just words. We need to actually be able to compute probabilities under a specific distribution. How do you actually determine the null distribution for permutation tests?

First we write a function for doing permutation testing

```
permutation.test <- function(group1, group2, FUN, repetitions) {
    stat.obs <- FUN(group1, group2)
    makePermutedStats <- function() {
        sampled <- sample(1:length(c(group1, group2)),
            size = length(group1), replace = FALSE)
        return(FUN(c(group1, group2)[sampled], c(group1,
            group2)[-sampled]))
    }
    stat.permute <- replicate(repetitions, makePermutedStats())
    p.value <- sum(stat.permute >= stat.obs)/repetitions
    return(list(p.value = p.value, observedStat = stat.obs,
        permutedStats = stat.permute))
}
```

Proportion Later than 15 minutes Now we implement it on our the SRS version of our flight data. Notice that I am going to make the statistic that I compare the absolute difference between the proportion later than 15 minutes. Recall, our summary statistics (only now I'm going to ignore cancelled flights)

```
tapply(flightSFOSRS$DepDelay, flightSFOSRS$Carrier,
    function(x) {
        x <- na.omit(x)
        sum(x > 15)/length(x)
    })[c("AA", "UA")]
```

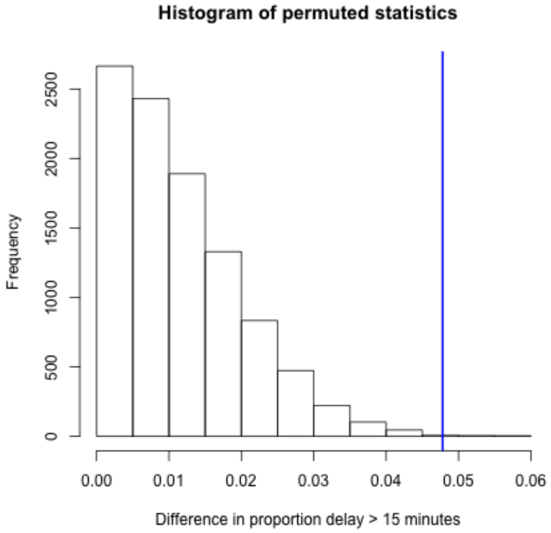
```
##        AA        UA
## 0.1511747 0.1989882
```

Why do I take the absolute difference? What difference does it make if you change the code to be only the difference?

```
set.seed(201728)
diffProportion <- function(x1, x2) {
    x1 <- na.omit(x1)
    x2 <- na.omit(x2)
    prop1 <- sum(x1 > 15)/length(x1)
    prop2 <- sum(x2 > 15)/length(x2)
    return(abs(prop1 - prop2))
}
dataset <- flightSFOSRS
output <- permutation.test(group1 = dataset$DepDelay[dataset$Carrier ==
    "UA"], group2 = dataset$DepDelay[dataset$Carrier ==
    "AA"], FUN = diffProportion, repetitions = 10000)
names(output)
```

```
## [1] "p.value"       "observedStat"  "permutedStats"
```

```
xlim <- range(c(output$observedStat, output$permutedStats))
hist(output$permutedStats, main = "Histogram of permuted statistics",
    xlab = "Difference in proportion delay > 15 minutes",
    xlim = xlim)
abline(v = output$observedStat, col = "blue", lwd = 2)
```



```
cat("pvalue=", output$p.value)
```

```
## pvalue= 8e-04
```

So what conclusions would you draw from this permutation test?

What impact does it have? What conclusions would you be likely to make going forward?

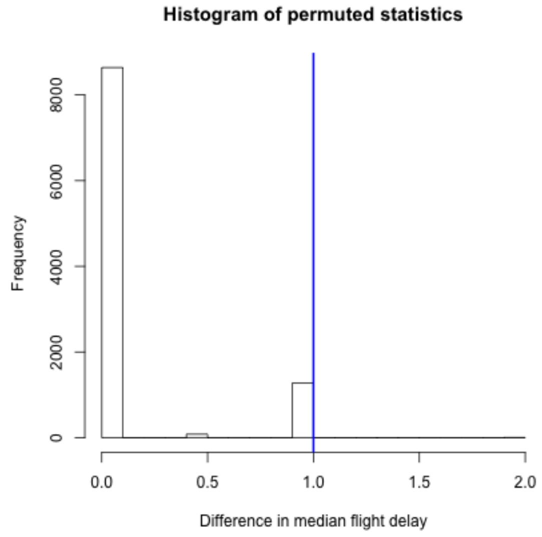
Median difference What about if I look at the median flight delay? The first thing we might note is that there is a very small difference (1 minute). So even if we find something significant, who really cares? That is not going to change any opinions about which airline I fly. Statistical significance is not everything.

```
tapply(flightSFOSRS$DepDelay, flightSFOSRS$Carrier,
    function(x) {
        median(x, na.rm = TRUE)
    })[c("AA", "UA")]
```

```
## AA UA
## -2 -1
```

I can just quickly change the function I look at to be the absolute difference in the median instead of proportion late.

```
set.seed(2843261)
diffMedian <- function(x1, x2) {
    prop1 <- median(x1, na.rm = TRUE)
    prop2 <- median(x2, na.rm = TRUE)
    return(abs(prop1 - prop2))
}
dataset <- flightSFOSRS
output <- permutation.test(group1 = dataset$DepDelay[dataset$Carrier ==
    "UA"], group2 = dataset$DepDelay[dataset$Carrier ==
    "AA"], FUN = diffMedian, repetitions = 10000)
xlim <- range(c(output$observedStat, output$permutedStats))
hist(output$permutedStats, main = "Histogram of permuted statistics",
    xlab = "Difference in median flight delay", xlim = xlim)
abline(v = output$observedStat, col = "blue", lwd = 2)
```



```
cat("pvalue=", output$p.value)
```

```
## pvalue= 0.1279
```

What is going on with our histogram?

What would have happened if we had defined out p-value as the probability of being *greater* rather than *greater than or equal to*? Where in the code was this done, and what happens if you change the code?

Choice of test statistic For simplicitly our null hypothesis was stated quite grandly as whether there was *any difference between distributions*. We picked two different statistics: the median and the proportion late. They made different choices about significance because they were measuring different things about the distributions. In that sense, permutation tests are usually only sensitive do differences between specific aspects of the distribution.

2.2 Assumptions: permutation tests

Let's discuss limitations of the permutation test.

What assumption(s) are we making about the random process that generated this data in determining the null distribution? Does it make sense for our data?

Some datasets have this flavor. For example, if we wanted to decide which of two email solicitations for a political campaign are most likely to lead to someone to donate money, we could assign a sample of people on our mailing list to get one of the two. This would perfectly match the data generation assumed in the null hypothesis.

What if our assumption about random labels is wrong? Clearly random assignment of labels is not a good description for how any of the datasets regarding flight delay data were created. Does this mean the permutation test will be invalid? No, not necessarily. We just need that the null hypothesis have a random process that created the data that leads to a distribution of the permutation test that is equivalent to as if we randomly assigned labels.

Explicitly describing this assumption is beyond the level of this class[1], but an important example where they are valid is if each of your data observations can be considered a random, independent draw from the same distribution (assuming the null is true and the distributions are the same bewteen groups). This is often abbreviated **i.i.d** (independent and identically distributed). This makes sense – the very act of permuting your data implies such an assumption about your data: that you have similar observations and the only thing different about them is which group they were assigned to.

Assuming your data is i.i.d is a common assumption that is thrown around, but is actually rather strong. For example, convenience samples do not have this property, because there is no randomness. However, permutation tests are a pretty good tool even in this setting, however, compared to the alternatives. Also to recognize that actual random assignments of the labels is the strongest such design of how to collect data.

Inferring beyond the sample population Note that the randomness queried by our null hypothesis is all about the specific observations we have. Specifically the randomness is if we imagine that we assigned *these same people* different email solicitations – our null hypothesis asks what variation in our statistic would we expect? However, if we want to extend to the general population, we have to make the assumption that these people's reaction are representative of the greater population.

[1] Namely, if the data can be assumed to be *exchangeable* under the null hypothesis then the permutation test is also a valid test.

For example, in our political email example we described above, if our sample of participants was only women, then the permutation test might have answered the question about any affect seend amongst these women was due to the chance assignment to these women. But that wouldn't answer our question very well about the general population of interest (that presumably includes men). Men might have very different reactions to the same email. Permutation tests do not get around the problem of a poor data sample. Random samples from the population are needed to be able to make the connection back to the general population.

So while permuting your data seems to intuitive and is often thought to make no assumptions, it does have assumptions about where your data are from. The assumptions are much less than some alternative tests (like the parametric tests we'll describe next), but it's useful to realize the limitations even for something as intuitive as non-restrictive as permutation tests.

3 Parametric test: the T-test

Parametric tests perform inference on a *parameter* of the underlying distributions, such as the median or mean.

Parameters Just as a statistic is any function of our observed data, a **parameter** is a function of the true generating distribution F. Parameters are often indicated with greek letters, like θ, α, β, σ. For the normal distribution, for example, there are two parameters that completely define the distribution: the mean μ and the variance σ^2.

Statistics of our data sample are often chosen because they are estimates of our parameter. In that case they are often called the same greek letters as the parameter, only with a "hat" on top of them, e.g. $\hat{\theta}$, $\hat{\alpha}$, $\hat{\beta}$, $\hat{\sigma}$. Sometimes, however, a statistic will just be given a upper-case letter, like T or X, particularly when they are not estimating a parameter of the distribution.

Often (but not always!) a statistic will be the same function of our sample distribution $\hat{F}$ that our parameter θ is of F. For example, the mean of our sample $X_1, \ldots, X_n$ has a distribution $\hat{F}$ and the mean is $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$. If $X_1, \ldots, X_n$ is a sample from a larger population $X_1, \ldots, X_N$, the population has mean $\mu = \frac{1}{N}\sum_{i=1}^{N} X_i$ distribution. Then $\hat{\mu}$ is an estimate of μ.

Null hypothesis for t-test The t-test is a widely used statistical test for comparing two groups that focuses on whether the means of the two distributions are the same. Let μ_{United}, and $\mu_{American}$ be the true medians of the two groups.

$$H_0 : \mu_{AA} = \mu_{UA}$$

This could also be written as

$$H_0 : \mu_{AA} - \mu_{UA} = \delta = 0,$$

so we are testing whether a specific parameter $\delta = 0$.

Let's assume $X_1, \ldots, X_{n_1}$ is the data from United and $Y_1, \ldots, Y_{n_2}$ is the data from American. A natural sample statistic to estimate δ from our data would be

$$\hat{\delta} = \bar{X} - \bar{Y},$$

i.e. the difference in the means of the two groups. This is the statistic of focus for the the t-test

3.1 Distribution of means under Null

To do inference, we need to know the distribution of our statistic of interest

Normality of mean In data 8, you learned that a sample mean of a SRS will have a sampling distribution that is roughly a normal distribution (the Central Limit Theorem) if we have a large enough sample size. Namely, that if $X_1, \ldots, X_n$ are a SRS from a distribution with mean μ and variance σ^2, then $\hat{\mu} = \bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$ will have a roughly normal distribution

$$N(\mu, \frac{\sigma^2}{n}).$$

If we have two groups,

- $X_1, \ldots, X_{n_1}$ a SRS from a distribution with mean μ_1 and variance σ_1^2, and
- $Y_1, \ldots, Y_{n_2}$ a SRS from a distribution with mean μ_2 and variance σ_2^2

Then $\bar{X} - \bar{Y}$ will have a roughly normal distribution

$$N(\mu_1 - \mu_2, \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2})$$

Testing If we want to compare the two groups, under the null $\mu_1 - \mu_2 = 0$, so the distribution of the difference in means is

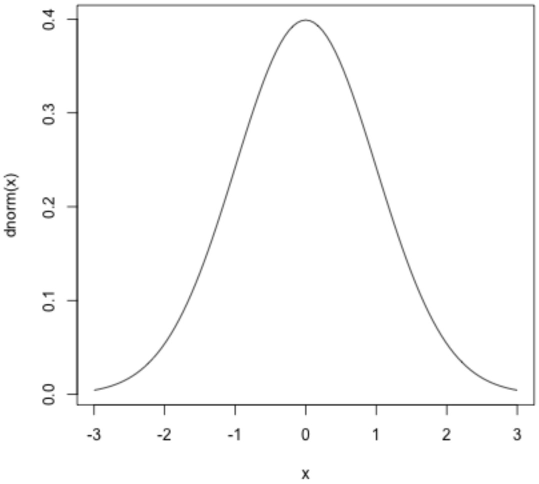
$$\bar{X} - \bar{Y} \sim N(0, \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2})$$

We can thus use that distribution to determine whether the observed $\bar{X} - \bar{Y}$ is unusual (assuming we know σ_1 and σ_2!).

We can also equivalently say,

$$Z = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \sim N(0, 1)$$

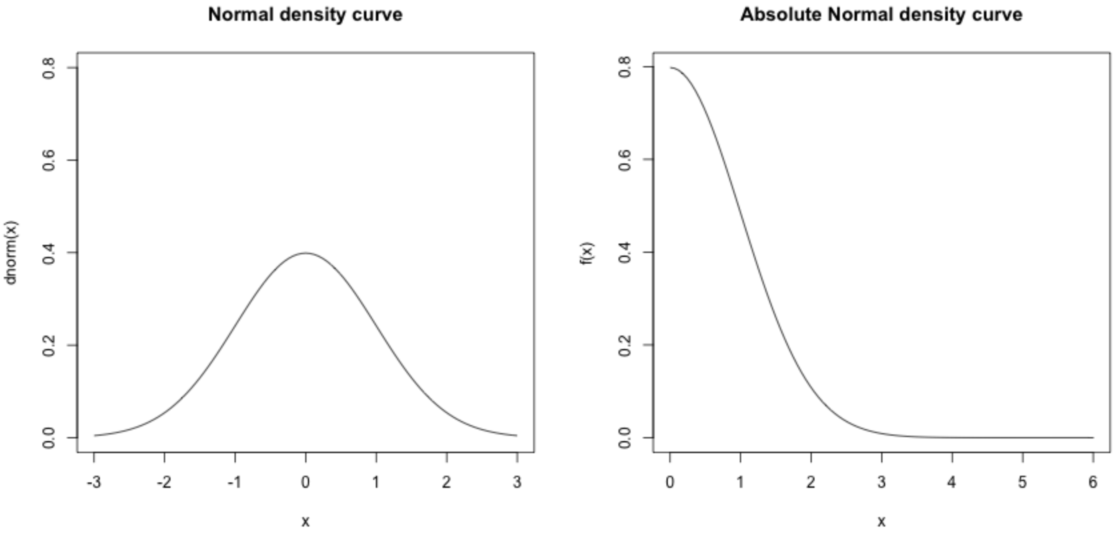
Then if we observe a value $Z = 2$, how would we calculate it's p-value?



Two-sided Tests Going back to our example, $\bar{X} - \bar{Y}$ might correspond to $X_{AA} - Y_{UA}$. But since we picked the order of the difference $\bar{X} - \bar{Y}$ arbitrarily, we also might ask what if $Y_{UA} - X_{AA}$ is large? If we've never looked at the data, either of these questions is equally relevant. Null hypotheses should *always* be based on what the question you want to answer is *before* looking at the data. So a better null hypothesis is whether either $Y_{UA} - X_{AA}$ OR $X_{AA} - Y_{UA}$ is large. This is more succinctly written as whether $|\bar{X} - \bar{Y}|$ is unusually large.

So a better statistic is,

$$Z = \frac{|\bar{X} - \bar{Y}|}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$



With this better Z statistic, what is the p-value if you observe $Z = 2$? How would you calculate this using the normal density curve? With R?

This is often called a 'two-sided' t-statistic, and is the only one that we will consider.

3.2 T-Test

The above is not in fact a statistic because we don't know σ_1 and σ_2. So we can't calculate Z from our data!

Instead you must estimate these unknown parameters with the **sample variance**

$$\hat{\sigma}_1^2 = \frac{1}{n-1} \sum (X_i - \bar{X})^2,$$

and the same for $\hat{\sigma}_2^2$. (Notice how we put a "hat" over a parameter to indicate that we've estimated it from the data.)

But once you must estimate the variance, you are adding additional variability to inference. Namely, before, assuming you knew the variances, you had

$$Z = \frac{|\bar{X} - \bar{Y}|}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}},$$

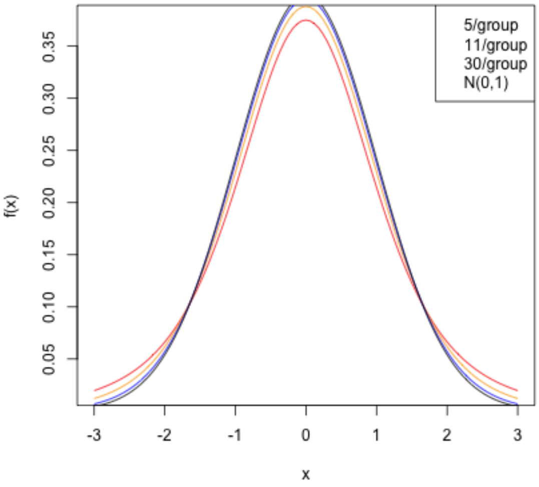
where only the numerator is random. Now we have

$$T = \frac{|\bar{X} - \bar{Y}|}{\sqrt{\frac{\hat{\sigma}_1^2}{n_1} + \frac{\hat{\sigma}_2^2}{n_2}}}.$$

and the denominator is also random. T is called the **t-statistic**.

This additional uncertainty means seeing a large value of T is more likely than of Z. Therefore, T has a different distribution, and it's not $N(0, 1)$.

Unlike the central limit theorem that deals only with the distributions of means, when you add on estimating the variance terms determining even approximately what is the distribution of T is more complicated, and in fact depends on the distribution of the input data X_i and Y_i (unlike the central limit theorem). But if the distributions creating your data are reasonably close to normal distribution, then T follows what is called a t-distribution.



You can see that the t distribution is like the normal, only it has larger "tails" than the normal, meaning seeing large values is more likely than in a normal distribution.

What happens as you change the sample size?

Notice that if you have largish datasets (e.g. $> 30 - 50$ samples in *each* group) then you can see that the t-distribution is basically the normal distribution, so that's why it's usually fine to just use the normal distribution to get p-values. Only in small samples sizes are there large differences.

3.3 Assumptions of the T-test

Parametric tests usually state their assumptions pretty clearly: they generally assume some model that generated the data in order to arrive at the mathematical description of the null distribution. For the t-test, we assume that the data $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$ are normal to get the t-distribution.

What happens if this assumption is wrong? When will it still make sense to use the t-test?

If we didn't have to estimate the variance, the central limit theorem tells us the normality assumption will work for any distribution, *if* we have a large enough sample size.

What about the t-distribution? That's a little tricker. You still need a large sample size; you also need that the distribution of the X_i and the Y_i, while not required to be exactly normal, not be too far from normal. In particular, you want them to be symmetric (unlike our flight data).[2] Generally, the t-statistic is reasonably robust to violations of these assumptions, particularly compared to other parametric tests, if your data is not too skewed and you have a largish sample size (e.g. 30 samples in a group is good). But the permutation test makes far fewer assumptions, and in particular is very robust to assumptions about the distribution of the data.

For small sample sizes (e.g. < 10 in each group), you certainly don't really have any good justification to use the t-distribution unless you have a reason to trust that the data is normally distributed (and with small sample sizes it is also very hard to justify this assumption by looking at the data).

3.4 Flight Data and Transformations

Let's consider the flight data. Recall, the t-statistic focuses on the difference in means. Why might this not be a compelling comparison?

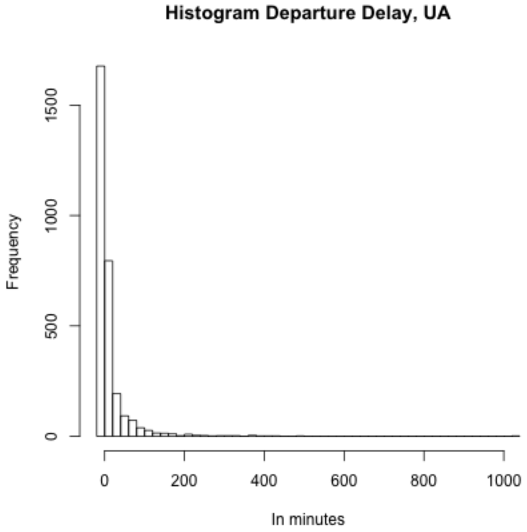
```r
tapply(flightSFOSRS$DepDelay, flightSFOSRS$Carrier,
    function(x) {
        mean(x, na.rm = TRUE)
    })[c("AA", "UA")]
```

[2] Indeed, the central limit theorem requires large data sizes, and how large a sample you need for the central limit theorem to give you a good approximation also depends on things about the distribution of the data, like how symmetric the distribution is.

```
##        AA        UA
##  7.728294 12.255649
```

However, you still – even with larger sample sizes – need to worry about the distribution of the data much more than with the permutation test. Very non-normal input data will not do well with the t-test, particularly if the data is **skewed**, meaning not symmetrically distributed around its mean.

Looking at the flight data, what would you conclude?



Note that nothing stops us from running the test, and it's a simple one-line code:

```
t.test(flightSFOSRS$DepDelay[flightSFOSRS$Carrier ==
    "UA"], flightSFOSRS$DepDelay[flightSFOSRS$Carrier ==
    "AA"])
```

```
##
##  Welch Two Sample t-test
##
## data:  flightSFOSRS$DepDelay[flightSFOSRS$Carrier == "UA"] and flightSFOSRS$DepDel
## t = 2.8325, df = 1703.1, p-value = 0.004673
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.392379 7.662332
## sample estimates:
## mean of x mean of y
## 12.255649  7.728294
```

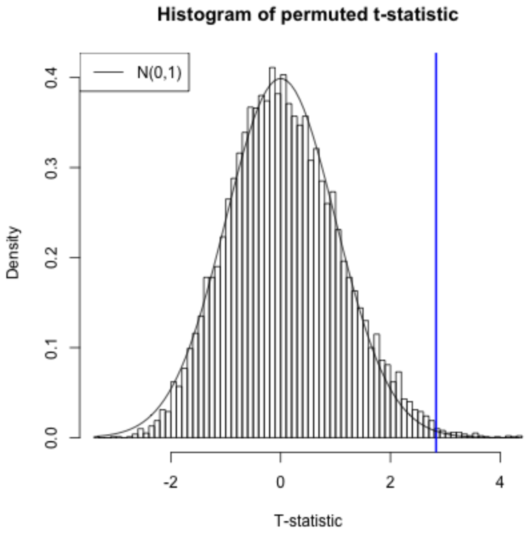
This is a common danger of parametric tests. They are implemented everywhere (there are on-line calculators that will compute this for you; excel will do this calculation), so people are drawn to doing this, while permutation tests are more difficult to find pre-packaged.

We can compare this to permutation test of the mean *using the exact same t-statistic*:

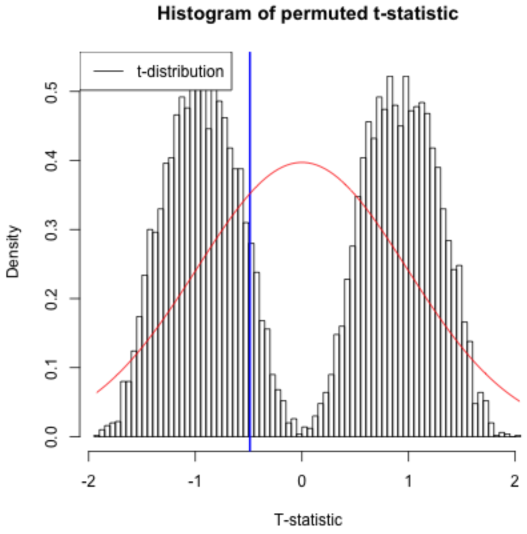
```
set.seed(489712)
tstatFun <- function(x1, x2) {
    abs(t.test(x1, x2)$statistic)
}
dataset <- flightSFOSRS
output <- permutation.test(group1 = dataset$DepDelay[dataset$Carrier ==
    "UA"], group2 = dataset$DepDelay[dataset$Carrier ==
    "AA"], FUN = tstatFun, repetitions = 10000)
cat("pvalue=", output$p.value)
```

```
## pvalue= 0.0059
```

In fact, in this case we get similar answers. We can compare the distribution of the permutation distribution of the t-statistic, and the density of the $N(0,1)$ that the parametric model assumes. We can see that they are quite close, even though our data is very skewed and clearly non-normal. Indeed for larger sample sizes, they will give similar results.



Smaller Sample Sizes If we had a smaller dataset we would not get such nice behavior. We can sample to a smaller sample of the data of size 20 and 30 in each group, and we can see that we do not get a permutation distribution that matches the (roughly) N(0,1) we use for the t-test.



```
cat("pvalue permutation=", outputTPermSmall$p.value,
    "\n")
```

```
## pvalue permutation= 0.5504
```

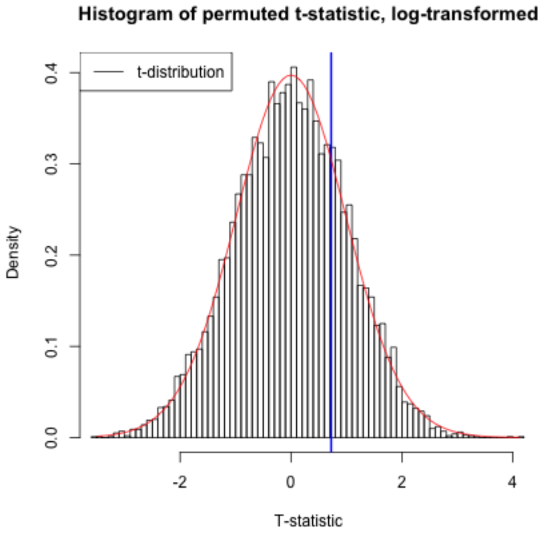
```
cat("pvalue t.test=", toutput$p.value, "\n")
```

```
## pvalue t.test= 0.6273047
```

What different conclusions do you get from the two tests with these smaller datasizes?

Transformations We saw that skewed data could be problematic in visualization of the data, e.g. in boxplots, and transformations are helpful. Transformations can also be helpful for applying the t-test. They can often result in the parametric t-test to work better for smaller datasets.

We've already seen in our flight data, some 'standard' transformations are not super compelling but do distributed the data to be slightly less skewed. We can see if we compare the permutation test versus the t-test on log-transformed data even with the smaller sample sizes, that after transforming the data, we get that the distribution looks much more like the permutation test, and the results are more similar.



```
## pvalue permutation= 0.2448
## pvalue t.test= 0.6273047
```

What does it mean for my null hypothesis to transform to the log-scale? Does this make sense?

3.5 Why parametric models?

We do the comparison of the permutation test to the parametric t-test not to encourage the use of the the t-test in this setting – the data, even after transformation, is pretty skewed and there's no reason to not use the permutation test instead. The permutation test will give pretty similar answers regardless of the transformation[3] and is clearly indicated here.

This exercise was to show the use and limits of using the parametric tests, and particularly transformations of the data, in an easy setting. Historically, t-tests were necessary in statistics because there were not computers to run permutation tests. That's clearly not compelling now! However, it remains that parametric tests are often easier to implement (one-line commands in R, versus writing a function); you will see parametric tests frequently (even when resampling methods like permutation tests and bootstrap would be more justifiable).

The take-home lessons here regarding parametric tests, are that when there are large sample sizes, parametric tests can overcome violations of their assumptions[4] so don't automatically assume parametric tests are completely wrong to use. But a permutation test is the better all-round tool for this question: it is has more minimal assumptions, and can look at how many different statistics we can use.

There are also some important reasons to learn about t-tests, however, beyond a history test. They are the easiest example of a **parameteric** test, where you make assumptions about your data (i.e. $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$ are normally distributed). Parametric tests generally are very important, even with computers. Parametric models are particularly helpful for researchers in data science for the development of new methods, particularly in defining good test statistics, like T.

Parametric models are also useful in trying to understand the limitations of a method, mathematically. We can simulate data under different models to understand how a statistical method behaves.

There are also applications where the ideas of bootstrap and permutation tests are difficult to apply. Permutation tests, in particular, are quite specific. Bootstrap methods, which we'll review in a moment, are more general, but still are not always easy to apply in more complicated settings. A goal of this class is to make you comfortable with parametric models (and their accompanying tests), in addition to the resampling methods you've learned.

[3] In fact, if we were working with the difference in the means, rather than the t-statistics, which estimates the variance, the permutation test would give exactly the same answer.

[4] At least those based on the central limit theorem!

4 Digging into Hypothesis tests

Let's break down some important concepts as to what makes a test. Note that all of these concepts will apply for *any* hypothesis test.

1. A test statistic
2. A null hypothesis of how the data was generated
3. The distribution of the test statistic under the null hypothesis.

As we've seen, different tests can be used to answer the same basic "null" hypothesis – are the two groups "different"? – but the specifics of how that null is defined can be quite different. For any test, you should be clear as to what the answer is to each of these points.

4.1 Significance & Type I Error

The significance refers to measuring how unlikely the null is. There are two terminologies that go along with assessing significance.

p-value One is to report a p-value to demonstrate how unlikely the data is under the null.

Q: Does the p-value give you the probability that the null is true?

Reject/Not reject We can just report the p-value, but it is common to also make an assessment of the p-value and give a final decision as to whether the null hypothesis was too unlikely to have reasonably created the data we've seen. This is a decision approach – either reject the null hypothesis or not. In this case we pick a cutoff, e.g. p-value of 0.05, and report that we reject the null.

You might see sentences like "We reject the null at level 0.05." The **level** chosen for a test is an important concept in hypothesis testing and is the cutoff value for a test to be significant. In principle, the idea of setting a level is that it is a standard you can require before declaring significance; in this way it can keep researchers from creeping toward declaring significance once they see the data and see they have a p-value of 0.07, rather than 0.05. However, in practice it can have the negative result of encouraging researchers to fish in their data until they find *something* that has a p-value less than 0.05.

Commonly accepted cutoffs for unlikely events are 0.05 or 0.01, but these values are too often considered as magical and set in stone. Reporting the actual p-value is more informative than just saying yes/no whether you reject (rejecting with a p-value of 0.04 versus 0.0001 tells you something about your data).

The useful idea about the level of the test is that it defines a repeatable procedure ("reject if p-value is < level"). Then the level actually measures the uncertainity in this procedure. Specifically, with any test, you can make two kinds of mistakes:

- Reject the null when the null is true (**Type I error**)
- Not reject the null when the null is in fact not true (**Type II error**)

Then the level is the probability of this procedure making a type I error: if you always reject at 0.05, then 5% of just tests will wrongly reject the null hypothesis when in fact it is true.

Note that this is no different in concept that our previous statement saying that a p-value is the likelihood under the null of an event as extreme as what we observed. However, it does a clear decision about how willing you are to making Type I Error.

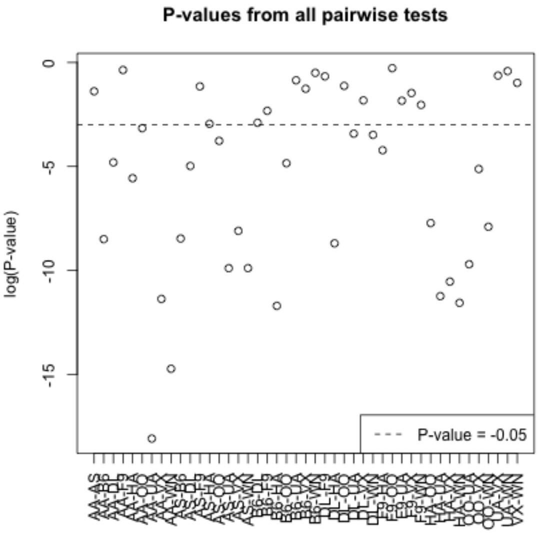
4.2 Type I Error & All Pairwise Tests

Let's make the importance of accounting and measuring Type I error more concrete. We have been considering only comparing the carriers United and American. But in fact there are 10 airlines. What if we want to compare all of them? What might we do?

```
## number of pairs: 45
```

For speed purposes in class, I'll use the t-test to illustrate this idea and calculate the t-statistic and its p-value for every pair of airline carriers (with our transformed data):

```
## Number found with p-value < 0.05:  26
```



What does this actually mean? Is this a lot to find significant?

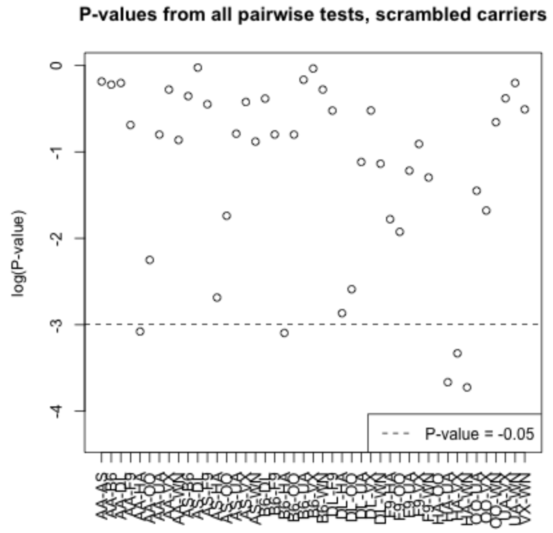
Roughly, if each of these tests has level 0.05, then even if *none* of the pairs are truly different from each other I might expect on average around 2 to be rejected at level 0.05 just because of variation in sampling.[5] This is the danger in asking many questions from your data – something is likely to come up just by chance. [6]

We can consider this by imagining what if I scramble up the carrier labels – randomly assign a carrier to a flight. Then I know there shouldn't be any true difference amongst the carriers. I can do all the pairwise tests and see how many are significant.

```
## Number permuted found with p-value < 0.05:  6
```

[5] In fact, this is not an accurate statement because these tests are reusing the same data, so the data in each test are not independent, and the probabilities don't work out like that. But it is reasonable for understanding the concepts here.

[6] Indeed this true of all of science, which relies on hypothesis testing, so one always has to remember the importance of the iterative process of science to re-examine past experiments.



What does this suggest to you about the actual data?

Multiple Testing Intuitively, we consider that if we are going to do all of these tests, we should have a stricter level so that we do not routinely find pairwise differences when there are none. Does this mean the level should be higher or lower to get a 'stricter' test? What about the p-value?

Making such a change to account for the number of tests run falls under the category of **multiple testing adjustments**, and there are many different flavors beyond the scope of the class. Let's consider the most widely known correction: the **Bonferroni correction**.

Specifically, say we'd like to be able to say "of all the tests I ran, there's only a 5% chance of a type I error". We can do a simple correction where we adjust the level we require for each test. Namely, if we run K tests and we want the overall probability of a single type I error to be 0.05, we can simply run each individual test at level $0.05/N$ – i.e. reduce the level of each test to be more stringent.

In the example of comparing the different airline carriers, the number of tests is 45. So our new level is now 0.0011. Taking our p-values from above, how would this change our conclusions?

```
cat("Number found significant after Bonferonni: ",
    sum(t.testPairs["p.value", ] < 0.05/npairs))
```

```
## Number found significant after Bonferonni:  16
```

```
cat("Number of shuffled differences found significant after Bonferonni: ",
    sum(t.testPairsScramble["p.value", ] < 0.05/npairs))
```

```
## Number of shuffled differences found significant after Bonferonni:  0
```

We can also, equivalently, adjust our p-values by *multiplying* the pvalues by N; these **adjusted p-values** can be compared to the overall level you want (0.05), just like regular p-values, in order to determine significance.

```
t.testPairs <- rbind(t.testPairs, p.value.adj = t.testPairs["p.value",
    ] * npairs)
t.testPairsScramble <- rbind(t.testPairsScramble, p.value.adj = t.testPairsScramble["
    ] * npairs)
t.testPairs[, 1:5]
```

```
##                  AA-AS         AA-B6        AA-DL      AA-F9       AA-HA
## statistic.t  1.1514752 -3.7413417707 -2.648054950 -0.3894014 3.101645905
## p.value      0.2501338  0.0002038769  0.008170586  0.6974224 0.003824936
## p.value.adj 11.2560196  0.0091744583  0.367676386 31.3840059 0.172122129
```

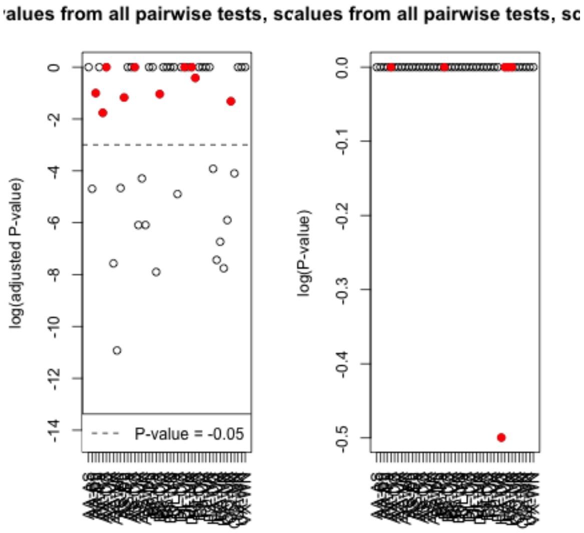
Notice some of these p-values are greater than 1! So in fact, we want to multiply by N, unless the value is greater than 1, in which case we set the p-value to be 1.

```
t.testPairs <- rbind(t.testPairs[1:2, ], p.value.adj = pmin(t.testPairs["p.value",
    ] * npairs, 1))
t.testPairsScramble <- rbind(t.testPairsScramble[1:2,
    ], p.value.adj = pmin(t.testPairsScramble["p.value",
    ] * npairs, 1))
t.testPairs[, 1:5]
```

```
##                 AA-AS         AA-B6        AA-DL      AA-F9       AA-HA
## statistic.t 1.1514752 -3.7413417707 -2.648054950 -0.3894014 3.101645905
## p.value     0.2501338  0.0002038769  0.008170586  0.6974224 0.003824936
## p.value.adj 1.0000000  0.0091744583  0.367676386  1.0000000 0.172122129
```

Now we plot these adjusted values

```
whLostSig <- which(t.testPairs["p.value.adj", ] > 0.05 &
    t.testPairs["p.value", ] <= 0.05)
whLostSigScramble <- which(t.testPairsScramble["p.value.adj",
    ] > 0.05 & t.testPairsScramble["p.value", ] <=
    0.05)
par(mfrow = c(1, 2))
plot(log(t.testPairs["p.value.adj", ]), ylab = "log(adjusted P-value)",
    main = "Adjusted P-values from all pairwise tests, scrambled carriers",
    xaxt = "n", xlab = "")
abline(h = log(0.05), lty = 2)
n <- ncol(t.testPairs)
points((1:n)[whLostSig], log(t.testPairs["p.value.adj",
    whLostSig]), col = "red", pch = 19)
legend("bottomright", legend = "P-value = -0.05", lty = 2)
axis(1, at = 1:ncol(t.testPairs), labels = colnames(t.testPairs),
    las = 2)
plot(log(t.testPairsScramble["p.value.adj", ]), ylab = "log(P-value)",
    main = "Adjusted P-values from all pairwise tests, scrambled carriers",
    xaxt = "n", xlab = "")
points((1:n)[whLostSigScramble], log(t.testPairsScramble["p.value.adj",
    whLostSigScramble]), col = "red", pch = 19)
abline(h = log(0.05), lty = 2)
axis(1, at = 1:ncol(t.testPairs), labels = colnames(t.testPairs),
    las = 2)
```



5 Confidence Intervals

Another approach to inference is with confidence intervals. Confidence intervals focus on parameters of a distribution, though they do not have to require parametric models to do so.

Form of a confidence interval Confidence intervals also do not rely on a specific null hypothesis; instead they give a range of values (based on the data) that are most likely to overlap the true parameter. Confidence intervals take the form of an interval, and are paired with a confidence, like 95% confidence intervals, or 99% confidence intervals.

Which should be wider intervals, a 95% or 99% interval?

5.1 Parametric Confidence Intervals

Confidence Interval for Mean of single group In data 8, you learned that a sample mean of a SRS will have a sampling distribution that is roughly a normal distribution (the Central Limit Theorem). Namely, that if $X_1, \ldots, X_n$ are a SRS from a distribution with mean μ and variance σ^2, then $\hat{\mu} = \bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$ will have a roughly normal distribution

$$N(\mu, \frac{\sigma^2}{n}).$$

Recall from data 9, how would we use the central limit theorem to create a 95% confidence interval for μ based on $\bar{X}$ (if we knew σ^2). We know that for a normal distribution, the probability of being within about 2 standard deviations of the mean is 0.95 – specifically 1.96 standard deviations. This is because these points define the **quantiles** of the normal distribution. Quantiles tell you at what point you will have a particular probability of being less than that value.

So if z is a 0.25 quantile of a normal distribution, it means that

$$P(\text{Normal} \leq z) = 0.25.$$

z is a 0.90 quantile of a normal if $P(\text{Normal} \leq z) = 0.90$.

These numbers can be looked up easily in R for a normal distribution,

```
qnorm(0.2, mean = 0, sd = 1)
```

```
## [1] -0.8416212
```

```
qnorm(0.9, mean = 0, sd = 1)
```

```
## [1] 1.281552
```

```
qnorm(0.0275, mean = 0, sd = 1)
```

```
## [1] -1.918876
```

What is the probability of being between -0.84 and 1.2815516 in a $N(0, 1)$?

Then what would a 95% confidence interval for the mean μ look like?

For the flight data, we can get a confidence interval for the mean of the United flights using `t.test` again. We will work on the log-scale, since we've already seen that makes more sense for parametric tests because our data is skewed:

```
t.test(log(flightSFOSRS$DepDelay[flightSFOSRS$Carrier ==
    "UA"] + addValue))
```

```
##
##  One Sample t-test
##
## data:  log(flightSFOSRS$DepDelay[flightSFOSRS$Carrier == "UA"] + addValue)
## t = 289.15, df = 2964, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  3.236722 3.280920
## sample estimates:
## mean of x
##  3.258821
```

Notice the result is on the (shifted) log scale! Because this is a monotonic function, We can invert this to see what this implies on the original scale:

```
logT <- t.test(log(flightSFOSRS$DepDelay[flightSFOSRS$Carrier ==
    "UA"] + addValue))
exp(logT$conf.int) - addValue
```

```
## [1] 3.450158 4.600224
## attr(,"conf.level")
## [1] 0.95
```

Confidence Interval for Difference in the Means of Two Groups Now lets consider whether the average delay time between the two groups is the same and define the parameter of interest as

$$\delta = \mu_{United} - \mu_{American}.$$

Using the central limit theorem again,

$$\bar{X} - \bar{Y} \sim N(0, \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2})$$

How would we use the central limit theorem to create a 95% confidence interval for $\mu_1 - \mu_2$ if we knew σ_1^2 and σ_2^2?

Of course, we don't know σ_1^2 and σ_2^2, so we will estimate them, as with the t-statistic. This effects our distribution assumptions, but we've already seen that for only moderate sample sizes, the difference between the normal and the t-distribution is not that great, so it is reasonable to use these same confidence intervals only with $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$.

We can get the confidence interval for the difference in our groups using `t.test` as well.

```
t.test(log(flightSFOSRS$DepDelay[flightSFOSRS$Carrier ==
    "UA"] + addValue), log(flightSFOSRS$DepDelay[flightSFOSRS$Carrier ==
    "AA"] + addValue))
```

```
##
```

```
##  Welch Two Sample t-test
##
## data:  log(flightSFOSRS$DepDelay[flightSFOSRS$Carrier == "UA"] + addValue) and log
## t = 5.7011, df = 1800.7, p-value = 1.389e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.07952358 0.16293414
## sample estimates:
## mean of x mean of y
##  3.258821  3.137592
```

What is the problem from this confidence interval on the log-scale that we didn't have before when we were looking at a single group?

5.2 More on deriving confidence intervals

Let's go back to the confidence interval for a single sample. For a $N(\mu, \sigma^2)$ distribution, $\mu - 1.96\sqrt{sigma^2}$ is the 0.025 quantile of the distribution, and $\mu + 1.96\sqrt{sigma^2}$ is the 0.975 quantile of the distribution, so the probability of being between these two is 0.95. Therefore, if $\bar{X}$ has a distribution $N(\mu, \sqrt{\frac{\sigma^2}{n}})$, then the standard deviation is $\sqrt{\frac{\sigma^2}{n}}$. Then for $\bar{X}$

$$\begin{aligned}
0.95 &= P(\mu - 1.96\sqrt{\frac{\sigma^2}{n}} \leq \bar{X} \leq \mu + 1.96\sqrt{\frac{\sigma^2}{n}}) \\
&= P(1.96\sqrt{\frac{\sigma^2}{n}} \leq \bar{X} - \mu \leq 1.96\sqrt{\frac{\sigma^2}{n}}) \\
&= P(1.96\sqrt{\frac{\sigma^2}{n}} - \bar{X} \leq -\mu \leq 1.96\frac{\sigma^2}{n} - \bar{X}) \\
&= P(\bar{X} - 1.96\sqrt{\frac{\sigma^2}{n}} \leq \mu \leq \bar{X} - 1.96\sqrt{\frac{\sigma^2}{n}})
\end{aligned}$$

You can do the same thing for two groups,

$$P((\bar{X} - \bar{Y}) - 1.96\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} \leq \mu_1 - \mu_2 \leq (\bar{X} - \bar{Y}) + 1.96\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}) = 0.95$$

The effect of estimating the variance We know from our t-test that if $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$ are normally distributed, then our t-statistic,

$$T = \frac{|\bar{X} - \bar{Y}|}{\sqrt{\frac{\hat{\sigma_1}^2}{n_1} + \frac{\hat{\sigma_2}^2}{n_2}}}.$$

has actually a t-distribution.

How does this get a confidence interval? We can use the same logic of inverting the equations, only with the quantiles of the t-distribution to get a confidence interval for the difference.

Let $t_{0.025}$ and $t_{0.975}$ be the quantiles of the t distribution. Then,

$$P((\bar{X} - \bar{Y}) - t_{0.975}\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} \leq \mu_1 - \mu_2 \leq (\bar{X} - \bar{Y}) - t_{0.025}\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}) = 0.95$$

Of course, since the t distribution is symmetric, $-t_{0.025} = t_{0.975}$. Why?

5.3 Bootstrap Confidence Intervals

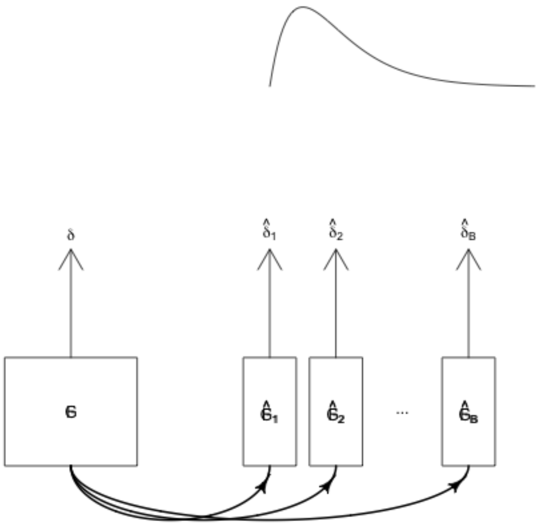
Suppose we are interested instead in whether the median of the two groups is the same. Why might that be a better idea than the mean?

Let θ_{United}, and $\theta_{American}$ be the true medians of the two groups, and now

$$\delta = \theta_{United} - \theta_{American}.$$

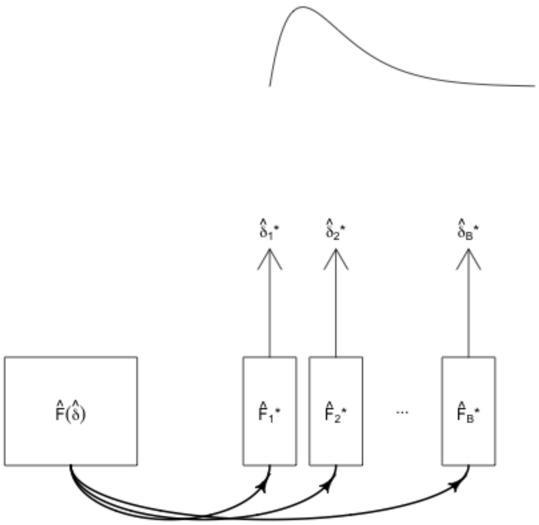
The sample statistic estimating δ would be what?

What we would like to be able to do is collect multiple data samples for any particular δ.



Since we only see one $\hat{\delta}$, this isn't an option. With normal-based confidence intervals (for the mean!), we used the central limit theorem that tells us the mean is approximately normal. For other statistics, like the difference in the median, we could also make assumptions about the distribution of the data to mathematically determine the distribution of $\hat{\delta}$. This can be very difficult to do mathematically for complicated statistics. More importantly, when you go with statistics that are beyond the mean, the mathematics often require more assumptions about the data-generating distribution – the central limit theorem works for most any distribution you can imagine, but that's a special property of the mean.

Rather than try to analyze this process mathematically, the bootstrap tries to estimate this process. Namely, we can't recreate samples from $F(\delta)$, but we can recreate samples from $\hat{F}(\hat{(\delta)})$,



The idea is that if $\hat{F}$ is close to F, then taking the quantiles of $\hat{\delta}^* - \hat{\delta}$ should give us good estimates of the quantiles of $\hat{\delta} - \delta$.

Implementing the bootstrap confidence intervals What does it actually mean to resample from $\hat{F}$? It means to take a sample from $\hat{F}$ just like the kind of sample we took from the actual data generating process, F.

Specifically in our two group setting, say we have a SRS $X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2}$ from an unknown distributions F, G. Then if we resample *with replacement* from the data sample $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$, we will get a new set of samples from each distribution, $X_1^*, \ldots, X_{n_1}^*$ and $Y_1^*, \ldots, Y_{n_2}^*$. These are samples from $\hat{F}$, $\hat{G}$ (the observed distribution of this new sample is what is called $\hat{F}^*$ and $\hat{G}^*$ in above picture). From this sample, we can recalculate the difference of the medians on this sample to get $\hat{\delta}^*$.

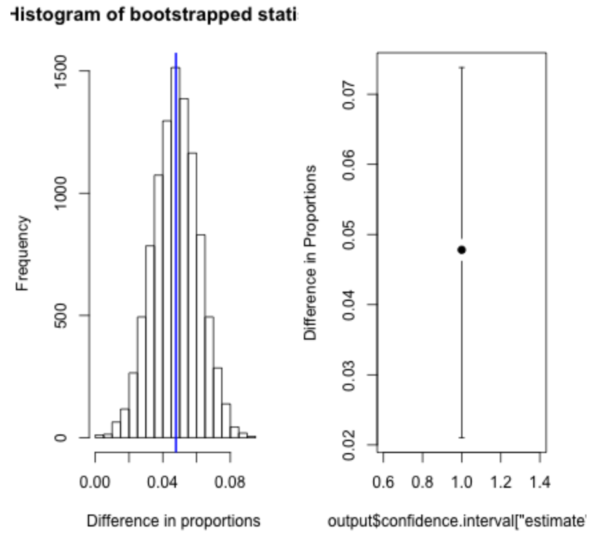
We do this repeatedly, and get a distribution of $\hat{\delta}^*$. Using the resulting distribution of $\hat{\delta}^*$, we use the 0.025 and 0.975 quantiles as the limits of the 95% confidence interval.

Again, we first write a function to do the bootstrap

```
bootstrap.test <- function(group1, group2, FUN, repetitions,
    confidence.level = 0.95) {
    stat.obs <- FUN(group1, group2)
    bootFun <- function() {
        sampled1 = sample(group1, size = length(group1),
            replace = TRUE)
        sampled2 = sample(group2, size = length(group2),
            replace = TRUE)
        FUN(sampled1, sampled2)
    }
    stat.boot <- replicate(repetitions, bootFun())
    level <- 1 - confidence.level
    confidence.interval <- quantile(stat.boot, probs = c(level/2,
        1 - level/2))
    return(list(confidence.interval = c(lower = confidence.interval[1],
        estimate = stat.obs, upper = confidence.interval[2]),
        bootStats = stat.boot))
}
```

Now we apply it to our flight data sets to get a confidence interval for the difference in proportion of late flights. Note how we can reuse our function from before.

```
set.seed(201728)
par(mfrow = c(1, 2))
dataset <- flightSFOSRS
output <- bootstrap.test(group1 = dataset$DepDelay[dataset$Carrier ==
    "UA"], group2 = dataset$DepDelay[dataset$Carrier ==
    "AA"], FUN = diffProportion, repetitions = 10000)
xlim <- range(c(output$confidence.interval["estimate"],
    output$bootStats))
hist(output$bootStats, main = "Histogram of bootstrapped statistics",
    xlim = c(xlim), xlab = "Difference in proportions")
abline(v = output$confidence.interval["estimate"],
    col = "blue", lwd = 2)
require(gplots)
plotCI(x = output$confidence.interval["estimate"],
    ui = output$confidence.interval["upper.97.5%"],
    li = output$confidence.interval["lower.2.5%"],
    ylab = "Difference in Proportions", pch = 19)
```



How do you interpret this confidence interval?

Assumptions: Bootstrap The big assumption of the bootstrap is that our sample distribution $\hat{F}$ is a good estimate of F. We've already seen that will not necessarily the case. Here are some examples of why that might fail:

- Sample size n is too small
- The data is not a SRS

We also need that the parameter θ and the statistic $\hat{\theta}$ to be well behaved in certain ways.

Optional: another bootstrap confidence interval We can actually use the bootstrap to calculate a confidence interval similarly to that of the normal distribution based on estimating the distribution of $\hat{\delta} - \delta$.

Notice with the previous calculation for $\bar{X}$, if I know

$$0.95 = P(1.96\sqrt{\frac{\sigma^2}{n}} \leq \bar{X} - \mu \leq 1.96\sqrt{\frac{\sigma^2}{n}})$$

Then I can invert to get

$$0.95 = P(\bar{X} - 1.96\sqrt{\frac{\sigma^2}{n}} \leq \mu \leq \bar{X} - 1.96\sqrt{\frac{\sigma^2}{n}})$$

So more generally, suppose we have points $z_{0.025}$ and $z_{0.975}$ so that

$$0.95 = P(z_{0.025} \leq \hat{\delta} - \delta \leq z_{0.975})$$

e.g. the 0.025 and 0.975 quantiles of $\hat{\delta} - \delta$. Then I can invert to get

$$0.95 = P(\hat{\delta} - z_{0.975} \leq \delta \leq \hat{\delta} - z_{0.025})$$

So if I can get the quantiles of $\hat{\delta} - \delta$, I can make a confidence interval.

So we could use the bootstrap to get estimates of the distribution of $\hat{\delta}-\delta$ instead of the distribution of $\hat{\delta}$ and use the quantiles of $\hat{\delta} - \delta$ to get confidence intervals that are $(\hat{\delta} - z_{0.975}, \hat{\delta} - z_{0.025})$. This actually gives a different confidence interval, particularly if the distribution of $\hat{\delta}$ is not symmetric. The first method is called (the one you learned in data 8) is called the percentile method, and is most commonly used, partly because it's easier to generalize.[7]

5.4 Thinking about confidence intervals

Suppose you have a 95% confidence interval for δ given by $(5, 15)$. What is wrong with the following statements regarding this confidence interval?

1. δ has a 0.95 probability of being between $(5, 15)$

2. If you repeatedly resampled the data, the difference D would be within $(5, 15)$ 95% of the time.

Confidence Intervals or Hypothesis Testing? Bootstrap inference via confidence intervals is more widely applicable than permutation tests we described above. These hypothesis tests relied on being able to simulate from the null hypothesis, by using the fact that if you detach the data from their labels you can use resampling techniques (either without replacement, i.e permuting, or with replacement from the data) to generate a null distribution. In settings that are more complicated than comparing groups, it can be difficult to find this kind of trick.

[7] If it looks like this method is backward compared to the percentile method, it pretty much is! But both methods are legitimate methods for creating bootstrap intervals.)

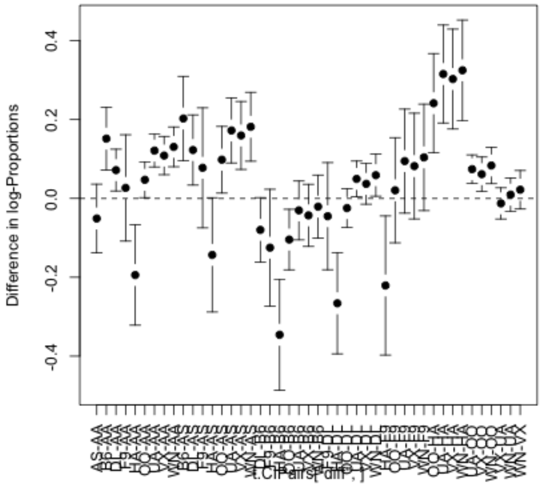
More generally, confidence intervals and hypothesis testing are actually closely intertwined. For example, for the parametric test and the parametric confidence interval, they both relied on the distribution of the same statistics, the t-statistic. If you create a 95% confidence interval, and then decide to reject a specific null hypothesis (e.g. $H_0 : \delta = 0$) only when it does not fall within the confidence interval, then this will exactly correspond to a test with level 0.05. So the same notions of level, and type I error, also apply to confidence intervals

Confidence intervals, on the other hand, give much greater interpretation and understanding about the parameter.

5.5 Revisiting pairwise comparisons

Just as with hypothesis testing, you can have multiple comparison problems with confidence intervals. Consider our pairwise comparisons of the different carriers. We can also create confidence intervals for them all. Again, we will use the t-test on the log-differences to make this go quickly.

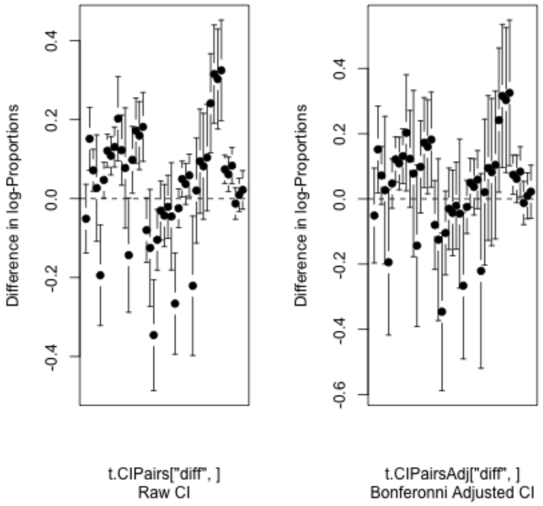
```
ttestCI <- function(x, variableName) {
    tout <- t.test(flightSFOSRS$logDepDelay[flightSFOSRS[,
        variableName] == x[2]], flightSFOSRS$logDepDelay[flightSFOSRS[,
        variableName] == x[1]])
    unlist(tout[c("estimate", "conf.int")])
}
t.CIPairs <- apply(X = pairsOfCarriers, MARGIN = 2,
    FUN = ttestCI, variableName = "Carrier")
colnames(t.CIPairs) <- paste(pairsOfCarriers[2, ],
    pairsOfCarriers[1, ], sep = "-")
t.CIPairs <- rbind(t.CIPairs, diff = t.CIPairs["estimate.mean of x",
    ] - t.CIPairs["estimate.mean of y", ])
require(gplots)
plotCI(x = t.CIPairs["diff", ], li = t.CIPairs["conf.int1",
    ], ui = t.CIPairs["conf.int2", ], ylab = "Difference in log-Proportions",
    pch = 19, xaxt = "n")
axis(1, at = 1:ncol(t.CIPairs), labels = colnames(t.CIPairs),
    las = 2)
abline(h = 0, lty = 2)
```



These confidence intervals suffer from the same problem as the p-values: even if the null value (0) is true in every test, roughly 5% of them will happen to not cover 0 just by chance.

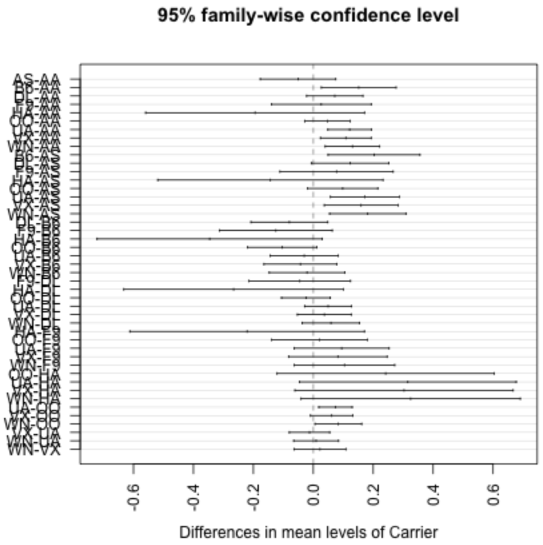
So we can do bonferonni corrections to the confidence intervals. Since a 95% confidence interval corresponds to a level 0.05 test, if we go to a $0.05/K$ level, which is the bonferonni correction, that corresponds to a $100 * (1 - 0.05/K)\%$ confidence interval.

```
ttestCIAdj <- function(x, variableName) {
    tout <- t.test(flightSFOSRS$logDepDelay[flightSFOSRS[,
        variableName] == x[2]], flightSFOSRS$logDepDelay[flightSFOSRS[,
        variableName] == x[1]], conf.level = 1 - 0.05/npairs)
    unlist(tout[c("estimate", "conf.int")])
}
t.CIPairsAdj <- apply(X = pairsOfCarriers, MARGIN = 2,
    FUN = ttestCIAdj, variableName = "Carrier")
colnames(t.CIPairsAdj) <- paste(pairsOfCarriers[2,
    ], pairsOfCarriers[1, ], sep = "-")
t.CIPairsAdj <- rbind(t.CIPairsAdj, diff = t.CIPairsAdj["estimate.mean of x",
    ] - t.CIPairsAdj["estimate.mean of y", ])
par(mfrow = c(1, 2))
plotCI(x = t.CIPairs["diff", ], li = t.CIPairs["conf.int1",
    ], ui = t.CIPairs["conf.int2", ], ylab = "Difference in log-Proportions",
    sub = "Raw CI", pch = 19, xaxt = "n")
abline(h = 0, lty = 2)
plotCI(x = t.CIPairsAdj["diff", ], li = t.CIPairsAdj["conf.int1",
    ], ui = t.CIPairsAdj["conf.int2", ], ylab = "Difference in log-Proportions",
    sub = "Bonferonni Adjusted CI", pch = 19, xaxt = "n")
abline(h = 0, lty = 2)
```



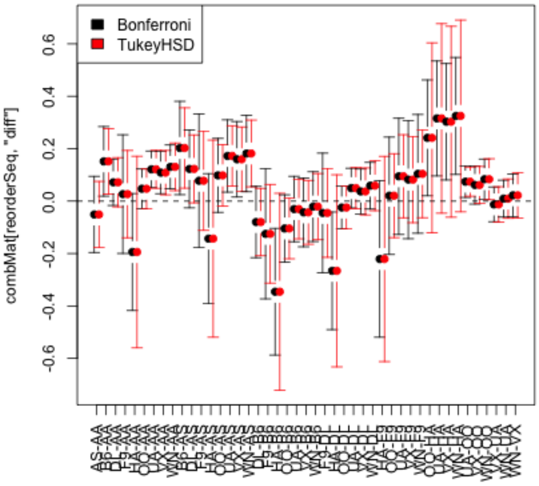
TukeyHSD In fact, as mentioned, there are many ways to do multiple testing corrections, and Bonferonni is the simplest, yet often most crude correction. There is a multiple testing correction just for pairwise comparisons that use the t-test, called the Tukey HSD test.

```
tukeyCI <- TukeyHSD(aov(logDepDelay ~ Carrier, data = flightSFOSRS))
plot(tukeyCI, las = 2)
```



Let’s compare them side-by-side.

```
combMat <- cbind(diff = c(t.CIPairsAdj["diff", ], tukeyCI$Carrier[,
    "diff"]), lwr = c(t.CIPairsAdj["conf.int1", ],
    tukeyCI$Carrier[, "lwr"]), upr = c(t.CIPairsAdj["conf.int2",
    ], tukeyCI$Carrier[, "upr"]))
reorderSeq <- rep(1:ncol(t.CIPairsAdj), each = 2) +
    rep(c(0, ncol(t.CIPairsAdj)), times = ncol(t.CIPairsAdj))
plotCI(x = combMat[reorderSeq, "diff"], li = combMat[reorderSeq,
    "lwr"], ui = combMat[reorderSeq, "upr"], pch = 19,
    xaxt = "n", col = c("black", "red"))
axis(1, at = seq(1.5, 2 * ncol(t.CIPairsAdj), by = 2),
    labels = colnames(t.CIPairsAdj), las = 2)
abline(h = 0, lty = 2)
legend("topleft", c("Bonferroni", "TukeyHSD"), fill = c("black",
    "red"))
```



What differences do you see?

Which to use? The TukeyHSD is a very specific correction – it is only valid for doing pairwise comparisons with the t-test. Bonferonni, on the other hand, can be used with any set of p-values from any test, e.g. permutation, and even if not all of the tests are pairwise comparisons.