

MATH 204: PRINCIPLES OF STATISTICS 2
WINTER 2026
MULTIPLE TESTING CORRECTIONS

Checking the null distributions For a two sample problem, to test the null hypothesis

$$H_0 : \mu_A = \mu_B$$

against a suitable alternative, we often use the test statistic whose observed value is

$$T_{obs} = \frac{\bar{x}_A - \bar{x}_B}{s_P \sqrt{\frac{1}{n_A} + \frac{1}{n_B}}}$$

where $\bar{x}_A$ and $\bar{x}_B$ are sample means from group A and group B, and

$$s_P^2 = \frac{(n_A - 1)s_A^2 + (n_B - 1)s_B^2}{n_A + n_B - 2}$$

is the ‘pooled’ estimate of the sample variance. Under the assumptions that the data are drawn independently from two populations

$$A : Normal(\mu_A, \sigma^2) \quad B : Normal(\mu_B, \sigma^2)$$

with the same variance where The group-specific sample variances are

$$s_A^2 = \frac{1}{n_A - 1} \sum_{i=1}^{n_A} (x_{Ai} - \bar{x}_A)^2 \quad s_B^2 = \frac{1}{n_B - 1} \sum_{i=1}^{n_B} (x_{Bi} - \bar{x}_B)^2$$

the theory tells us that if H_0 is true, the test statistic T satisfies $T \sim t_{n_A+n_B-2}$.

What does this result mean ?

- If we simulate the data under the stated conditions, and compute the test statistic, it should look like an observation from the t-distribution. with $n_A + n_B - 2$ degrees of freedom.
- This theory holds for $n_A, n_B \geq 2$.

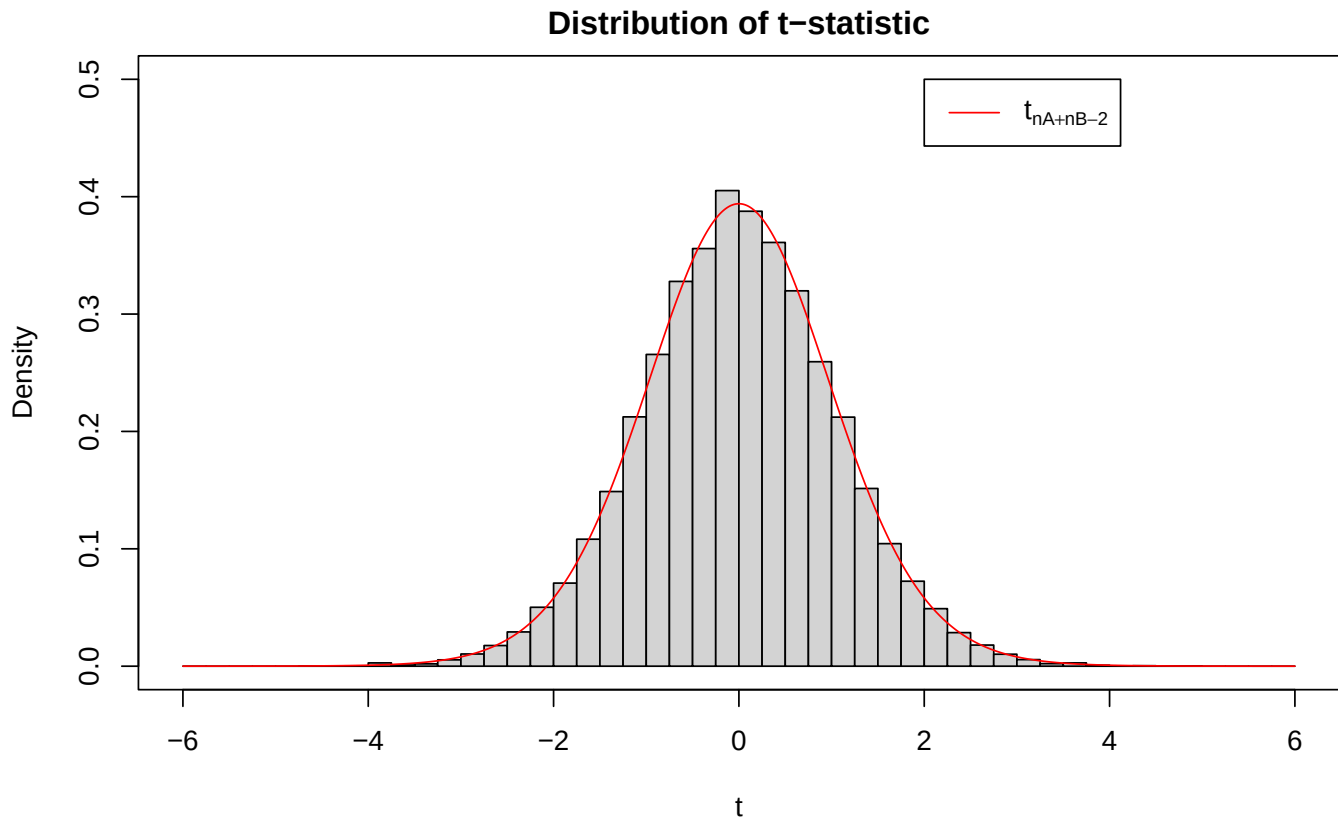
We can verify this theory using a simulation study, which essentially involves simulating a large number of replicate data sets under the stated conditions, and checking the distribution of the generated test statistics.

Here we specify $n_A = 10$, $n_B = 12$ and $\mu_A = \mu_B = 2$ (so that the null hypothesis is true), with $\sigma^2 = 3^2$. We use the `replicate` function in R instead of a loop, based on a specially written function for simulating the data `two.test.sim`. The simulation is based on 20000 replicated data sets.

```

> nreps<-20000                                #Number of replicates
> nA<-10;nB<-12                                #Sample sizes
> muA<-muB<-2; sigma<-3
>
> two.test.sim<-function(nA,nB,muA,muB,sigmaA,sigmaB){
+   xA<-rnorm(nA,muA,sigmaA)
+   xB<-rnorm(nB,muB,sigmaB)
+   vA<-var(xA)
+   vB<-var(xB)
+   vP<-( (nA-1)*vA+(nB-1)*vB)/(nA+nB-2)
+   tval<-(mean(xA)-mean(xB))/sqrt(vP*(1/nA+1/nB))
+ }
>
> t.stats<-replicate(nreps,two.test.sim(nA,nB,muA,muB,sigma,sigma))
> par(mar=c(4,4,2,0))
> br<-seq(-6,6,by=0.25)
> hist(t.stats,freq=FALSE,breaks=br,xlab='t',main='',
+   ylim=range(0,0.5));box()
> xv<-seq(-6,6,by=0.01)
> lines(xv,dt(xv,df=nA+nB-2),col='red')
> legend(2,0.5,c(expression(t[nA+nB-2])),lty=1,col='red')
> title('Distribution of t-statistic')

```



The theory is supported by the simulations.

Now we examine the observed Type I error rate of a test at significance level α .

```
> alpha<-0.05
>
> two.test.sim.result<-function(nA,nB,muA,muB,sigmaA,sigmaB,alpha){
+   xA<-rnorm(nA,muA,sigmaA)
+   xB<-rnorm(nB,muB,sigmaB)
+   vA<-var(xA)
+   vB<-var(xB)
+   vP<-(nA-1)*vA+(nB-1)*vB/(nA+nB-2)
+   tval<-(mean(xA)-mean(xB))/sqrt(vP*(1/nA+1/nB))
+   C.R<-qt(1-alpha/2,nA+nB-2)
+   Result<-abs(tval) >= C.R
+ }
>
> t.results<-replicate(nreps,
+   two.test.sim.result(nA,nB,muA,muB,sigma,sigma,alpha))
>
> sum(t.results)/nreps #Computes the rejection rate
+ [1] 0.0521
```

Of the 20000 replicate data sets simulated under conditions where each time the null hypothesis is true, 1042 resulted in a rejection of the null using the t-test. Thus the observed Type I error rate in this simulation is 0.0521. This is close to the theoretical Type I error rate $\alpha = 0.05$.

Familywise Type I Error Rate: Now suppose we are going to carry out $c = 20$ tests, and want to report the rate at which **at least one** of the c tests yields a rejection of the null hypothesis.

```
> two.test.sim.mult<-function(nt,nA,nB,muA,muB,sigmaA,sigmaB,alpha){
+   t.results<-replicate(nt,
+     two.test.sim.result(nA,nB,muA,muB,sigma,sigma,alpha))
+   Result<-sum(t.results) > 0
+ }
>
> ntests<-20
> mult.t.results<-replicate(nreps,
+   two.test.sim.mult(ntests,nA,nB,muA,muB,sigma,sigma,alpha))
> sum(mult.t.results)/nreps #Computes the Familywise error rate
+ [1] 0.63995
```

Of the 20000 replicate collections of 20 tests carried out under conditions where each time the null hypothesis is true, 12799 resulted in a rejection of the null **at least once** out of the 20 tests. Thus the observed Familywise Type I error rate in this simulation is 0.63995.

From the theory given in lectures, the actual Familywise Type I error rate is

$$\alpha_F = 1 - (1 - \alpha)^c$$

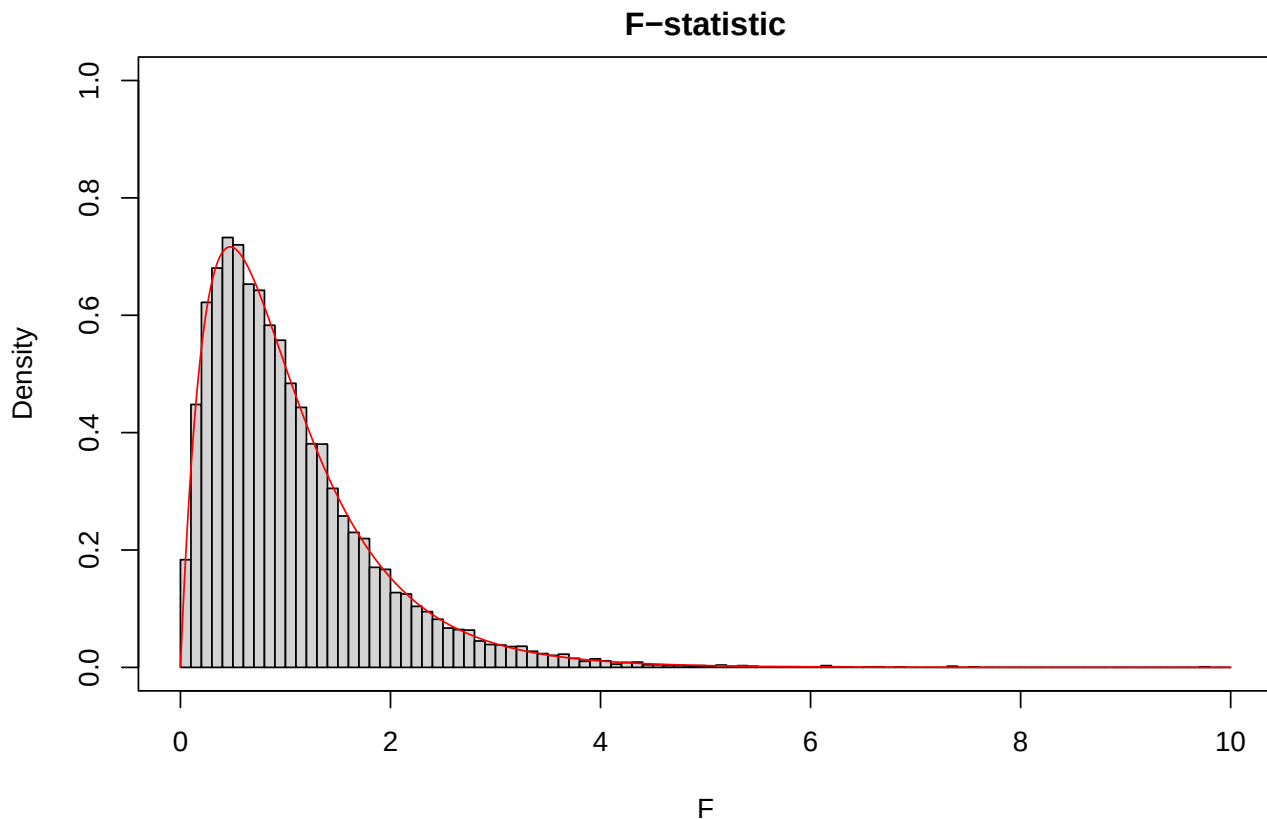
which here is equal to

$$\alpha_F = 1 - (1 - 0.05)^{20} = 0.641514.$$

Post-Hoc Tests in the ANOVA F-test: We now examine the implications for post-hoc testing of group mean differences in the Completely Randomized Design. We take for example $k = 5$ groups, and $n = 40$, with $n_1 = 10, n_2 = 8, n_3 = 6, n_4 = 7, n_5 = 9$, and generate data under the null hypothesis that the treatment group means and variances are equal. Specifically, we take each $\mu_i = 5$ and $\sigma^2 = 1$.

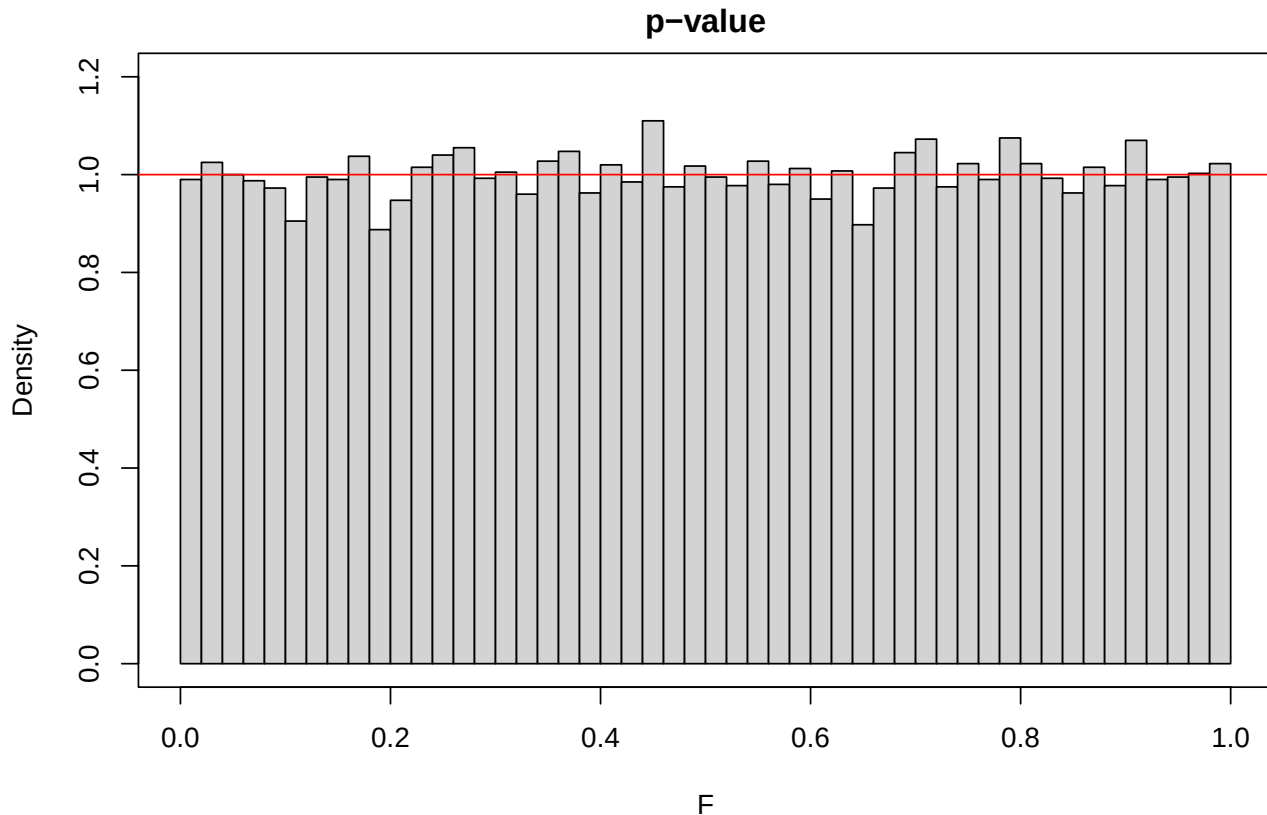
We first examine the behaviour of the ANOVA F-statistic, which in this case, should be a random variable with an $F_{k-1, n-k} = F_{4,35}$ distribution.

```
> myseed<-101
> set.seed(myseed)
> k<-5
> nvec<-c(10,8,6,7,9);n<-sum(nvec)
> tmt<-as.factor(rep(1:5,nvec))
> Fstat<-pval<-rep(0,nreps)
> for(irep in 1:nreps){
+   Y<-rnorm(sum(nvec))
+   ANOVA<-anova(lm(Y~tmt))
+   Fstat[irep]<-ANOVA[1,4]
+   pval[irep]<-ANOVA[1,5]
+ }
> par(mar=c(4,4,2,2))
> hist(Fstat[Fstat<10],breaks=seq(0,10,by=0.1),freq=FALSE,
+      ylim=range(0,1), main='F-statistic',xlab='F')
> box()
> xv<-seq(0,10,by=0.01)
> yv<-df(xv,k-1,n-k)
> lines(xv,yv,col='red')
```



It is also instructive to examine the distribution of p-values from the 20000 replicates.

```
> par(mar=c(4, 4, 2, 2))
> hist(pval, breaks=seq(0, 1, by=0.02), freq=FALSE,
+      ylim=range(0, 1.2), main='p-value', xlab='F')
> box()
> abline(h=1, col='red')
```



This distribution is uniform on the range (0, 1); this is also predicted by theory.

The critical value in the F-test at significance level $\alpha = 0.05$ is

$$C_R = F_{\alpha, 4, 35} = 2.641$$

and we can use this to assess how many times the null hypothesis is actually rejected.

```
> (C.R<-qf(1-alpha, k-1, n-k))
+ [1] 2.64147
> (rejections<-sum(Fstat >= C.R)) #Count the number of rejections
+ [1] 1019
> sum(pval <= alpha)
+ [1] 1019
```

The number of rejections of the null hypothesis is 1019, and so the observed Type I error rate is 0.05095, close to $\alpha = 0.05$. The theory is supported by the simulations.

We now re-run the simulation storing the results of post-hoc testing using the two functions:

- `pairwise.t.test`: this looks at all pairwise comparisons of group means, and carries out a test of the null hypothesis that the group means are equal. The p-values are reported, and can be adjusted to take into account that multiple tests are carried out. The simplest correction is the **Bonferroni correction**, which multiplies the p-values from the direct comparisons by the number of tests carried out, denoted c .

By default, the test statistic for comparing group i with group j is computed as

$$\frac{\bar{x}_i - \bar{x}_j}{s_P \sqrt{\frac{1}{n_i} + \frac{1}{n_j}}}$$

where s_P^2 is a **pooled** estimate of the error variance

$$s_P^2 = MSE = \frac{SSE}{n - k}$$

obtained from **all** the groups. The t-tests are completed, and t-values computed, by comparing against a t-distribution with $n - k$ degrees of freedom.

- `TukeyHSD`: this computes confidence intervals for the differences in group means between all treatment groups, but adjusts for the fact that multiple comparisons are being carried out. This adjustment has the effect of making the confidence intervals **wider**. HSD stands for 'Honest Significant Difference'.

More basic forms of adjustment of confidence intervals can be carried out also. The general form of a confidence interval in a two group comparison is

$$(\bar{x}_i - \bar{x}_j) \pm t_{\alpha/2, \nu} s \sqrt{\frac{1}{n_i} + \frac{1}{n_j}}$$

where $t_{a, \nu}$ is the upper a point of the t-distribution with ν degrees of freedom, and s is an estimate of the random variability present in the data (that is, s^2 is an estimate of σ^2). We might

- set α to be the Familywise Type I error rate
- use the pooled estimate s_P for s and set $\nu = n - k$
- use the pooled estimate for s from the two group comparison

$$s^2 = \frac{(n_i - 1)s_i^2 + (n_j - 1)s_j^2}{n_i + n_j - 2}$$

and set $\nu = n_i + n_j - 2$.

Each choice will make a slight difference to the reported confidence intervals.

In the example, $k = 5$, so the number of tests carried out is $c = k(k - 1)/2 = 10$.

```

> set.seed(myseed)
> tmt<-as.factor(rep(1:5,nvec))
> Y<-rnorm(sum(nvec))
> pairwise.t.test(Y,tmt, p.adjust.method='none')$p.value

+           1           2           3           4
+ 2 0.3148765          NA          NA          NA
+ 3 0.0558803 0.326357          NA          NA
+ 4 0.9559187 0.384039 0.0827069          NA
+ 5 0.4802157 0.750382 0.1968474 0.554909

> pairwise.t.test(Y,tmt, p.adjust.method='bonferroni')$p.value

+           1  2           3  4
+ 2 1.000000 NA          NA NA
+ 3 0.558803  1          NA NA
+ 4 1.000000  1 0.827069 NA
+ 5 1.000000  1 1.000000  1

```

We can see that the p-values using the Bonferroni correction are $c = 10$ times the unadjusted p-values (with values capped at 1). To complete the tests, we compare the p-values (unadjusted or adjusted) with our selected significance level $\alpha = 0.05$, say

We carry out a simulation study to examine the Familywise Type I error rate associated with the pairwise t-testing. In the 20000 replicate data sets, we compute and store the matrix of pairwise p-values, and then count how many times at least one of the null hypotheses is rejected out of the $c = 10$ tests. We then repeat the calculation making the Bonferroni correction by multiplying the p-values by 10.

```

> set.seed(myseed)
> pmat<-array(0,c(nreps,k-1,k-1))
> for(irep in 1:nreps){
+   Y<-rnorm(sum(nvec))
+   pmat[irep,,]<-pairwise.t.test(Y,tmt,p.adjust.method='none')$p.value
+ }
> unadj.count<-sum(apply(pmat,1,function(x){min(x,na.rm=T)}) <= 0.05)
> adj.count<-sum(apply(10*pmat,1,function(x){min(x,na.rm=T)}) <= 0.05)
>
> unadj.count/nreps    #Unadjusted p-value Type I error rate
+ [1] 0.2694

> adj.count/nreps     #Adjusted p-value Type I error rate
+ [1] 0.0384

```

An unadjusted analysis yields a Familywise Type I error rate of 0.2694, whereas the Bonferroni adjusted version yields a rate of 0.0384.

According to the theory, the Familywise Type I error rate, α_F , is given by

$$\alpha_F = 1 - (1 - \alpha)^c$$

so that if we want to fix $\alpha_F = 0.05$, we must choose

$$\alpha = 1 - (1 - \alpha_F)^{1/c}.$$

In this case, this means that in fact we should choose a slightly larger p-value threshold to maintain a Familywise Type I error rate at $\alpha_F = 0.05$.

```
> alpha.S<-1-(1-0.05)^(1/10)
> alpha.S

+ [1] 0.0051162

> sadj.count<-sum(apply(pmat,1,function(x){min(x,na.rm=T)}) <= alpha.S)
>
> sadj.count/nreps      #Adjusted p-value Type I error rate

+ [1] 0.03895
```

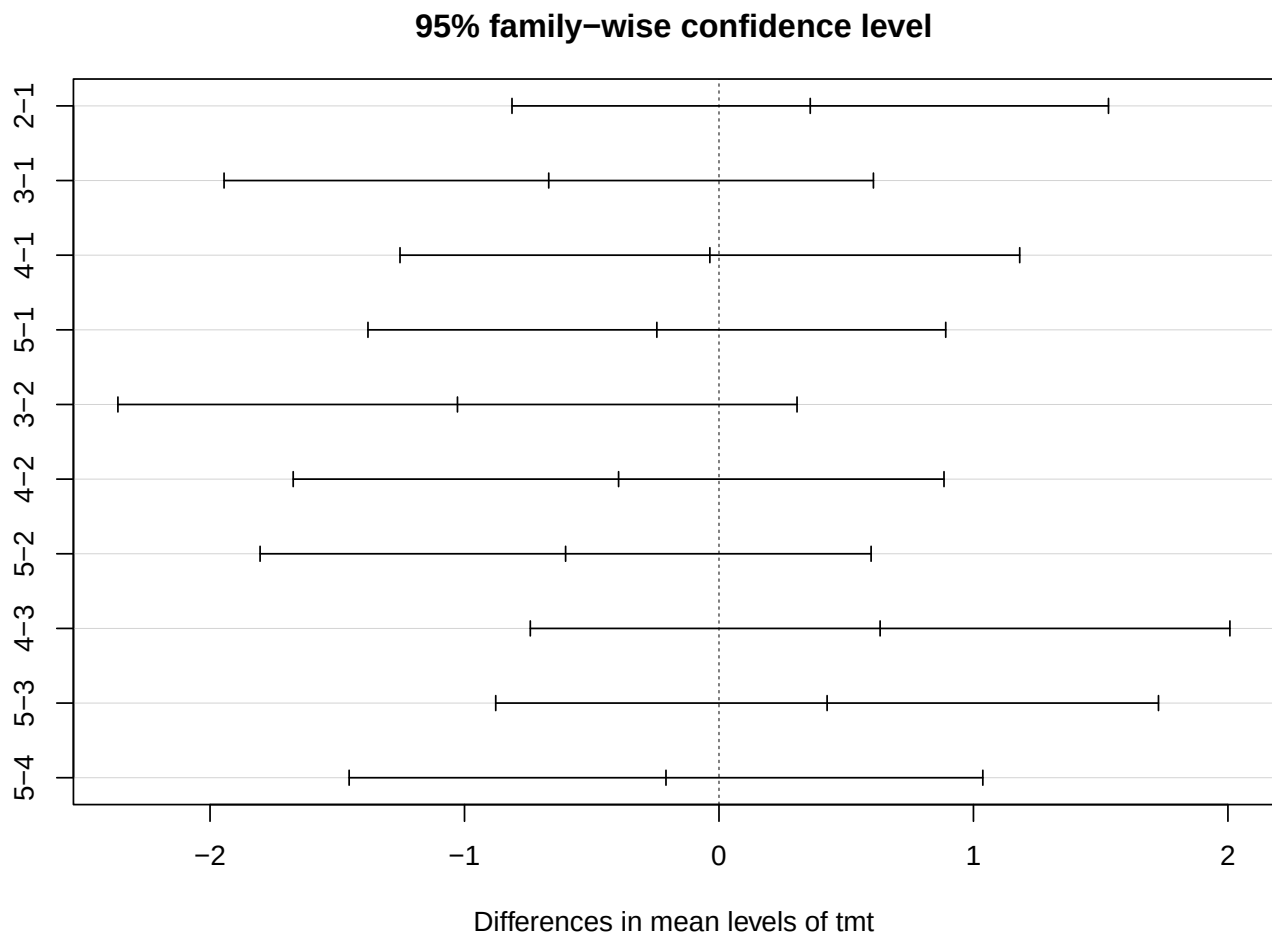
For the Tukey HSD intervals:

```
> TukeyHSD(aov(Y~tmt))

+ Tukey multiple comparisons of means
+ 95% family-wise confidence level
+
+ Fit: aov(formula = Y ~ tmt)
+
+ $tmt
+
+      diff      lwr      upr    p adj
+ 2-1  0.3586500 -0.813344 1.530644 0.902402
+ 3-1 -0.6690004 -1.944906 0.606905 0.564660
+ 4-1 -0.0358418 -1.253455 1.181772 0.999988
+ 5-1 -0.2441061 -1.379351 0.891139 0.971201
+ 3-2 -1.0276505 -2.362024 0.306723 0.198314
+ 4-2 -0.3944919 -1.673242 0.884258 0.899801
+ 5-2 -0.6027561 -1.803339 0.597827 0.604682
+ 4-3  0.6331586 -0.741456 2.007774 0.678270
+ 5-3  0.4248944 -0.877321 1.727110 0.880039
+ 5-4 -0.2082643 -1.453420 1.036892 0.988626
```

In the following plot, the adjusted 95% confidence intervals for the 10 different comparisons are plotted as horizontal bars, with the groups being compared listed on the vertical axis.

```
> par(mar=c(5,4,2,0));plot(TukeyHSD(aov(Y~tmt)))
```



The TukeyHSD intervals, and any other confidence interval, can be assessed to see whether it meets its targeted confidence level. Recall that a $100(1 - \alpha)\%$ confidence interval should contain the true value of the parameter with probability $(1 - \alpha)$, that is, the interval should exhibit **coverage** at rate $(1 - \alpha)$. The TukeyHSD procedure is designed to produce Familywise coverage at the target rate for the family of confidence intervals. In this case the true value of the parameter is always zero for each group wise comparison, so $(1 - \alpha)$ should be the rate at which **all** confidence intervals contain zero. We may test this out in simulation.

In the following examples, we compare four methods of confidence interval construction:

- 1 Tukey HSD method
- 2 **Method 1:** All pairwise CIs, with $\alpha = 0.05$, and a **pooled estimate** of the sample variance.
- 3 **Method 2:** All pairwise CIs, with $\alpha = 0.05/c$, and a **pooled estimate** of the sample variance.
- 4 **Method 3:** All pairwise CIs, with $\alpha = 0.05/c$, and with **separate variance** estimates for each comparison.

```
> set.seed(myseed)
> cil<-ci2<-ci3<-rep(0,k*(k-1)/2)
> Tukey.coverage<-rep(0,nreps)
> method1.coverage<-method2.coverage<-method3.coverage<-rep(0,nreps)
```

```

> for(irep in 1:nreps){
+   Y<-rnorm(sum(nvec))

+   #Extract the Tukey HSD intervals
+   ci<-TukeyHSD(aov(Y~tmt))$tmt[,2:3]
+   Tukey.coverage[irep]<-min(ci[,1]<0 & ci[,2]>0)

+   #Pooled estimate from all groups
+   sP.sq<-summary(lm(Y~tmt))$sigma^2
+   tval<-qt(0.975,n-k);      tval.s<-qt(0.995,n-k)

+   #Compute the coverage other adjustment methods
+   ico<-0
+   for(i in 2:k){
+     xi<-Y[tmt==i]
+     xmi<-mean(xi)
+     ni<-length(xi)
+     for(j in 1:(i-1)){
+       xj<-Y[tmt==j]
+       xmj<-mean(xj)
+       nj<-length(xj)

+       xd<-(xmi-xmj)
+       del<-sqrt(sP.sq*(1/ni+1/nj))
+       ico<-ico+1
+       ci1[ico]<-as.numeric(xd-tval*del<0 & xd+tval*del)
+       ci2[ico]<-as.numeric(xd-tval.s*del<0 & xd+tval.s*del)

+       sP.ij<-((ni-1)*var(xi)+(nj-1)*var(xj))/(ni+nj-2)
+       del<-sqrt(sP.ij*(1/ni+1/nj))
+       tvalij<-qt(0.995,ni+nj-2)
+       ci3[ico]<-as.numeric(xd-tvalij*del<0 & xd+tvalij*del)
+     }
+   }
+   method1.coverage[irep]<-min(ci1)
+   method2.coverage[irep]<-min(ci2)
+   method3.coverage[irep]<-min(ci3)
+ }
> coverage.results<-c(mean(Tukey.coverage),mean(method1.coverage),
+   mean(method2.coverage),mean(method3.coverage))
> coverage.results

+ [1] 0.94965 0.83310 0.96035 0.95795

```

Whereas the Tukey HSD method and Methods 2 and 3 provide coverage at approximately the 95% level, Method 1 has too low a coverage rate of 83.31. This is in part due to the fact that no adjustment is made for multiple testing.