Multiple Regression III

Last time...

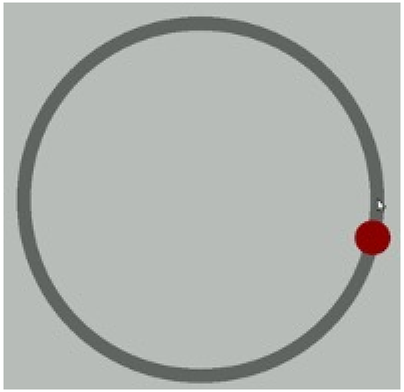
- Model comparison
- Categorical predictors/dummy coding

Today...

ANOVA, the long way

It's important to note that by covering dummy coding a categorical variable, we have already covered ANOVA -- there is nothing more you can learn from this technique that the omnibus test of that model will not tell you.

Eye-hand coordination task while subjected to periodic 3-second bursts of 85 dB white noise played over earphones. They had participants keep a mouse pointer on a red dot that moved in a circle at a rate of 1 revolution per second. Participants performed the task until they allowed the pointer to stray from the rotating dot 10 times. The time (in seconds) at the 10th failure was the outcome measure.



The participants were randomly assigned to one of four noise conditions:

- controllable and predictable noise
- uncontrollable but predictable noise
- controllable but unpredictable noise
- uncontrollable and unpredictable noise.

When noise was *predictable*, the 3-second bursts of noise would occur regularly every 20 seconds.

When noise was *unpredictable*, the 3-second bursts would occur randomly (although every 20 seconds on average).

When noise was *uncontrollable*, participants could do nothing to prevent the noise from occurring.

When noise was *controllable*, participants were shown a button that would prevent the noise, but they were told, "the button is a safety measure, for your protection, but we would prefer that you not use it unless absolutely necessary." No participants actually used the button.

Why is **random assignment** important in this design?

```r
library(here)
rotate = read.csv(here("data/pursuit_rotor.csv"))
head(rotate)
```

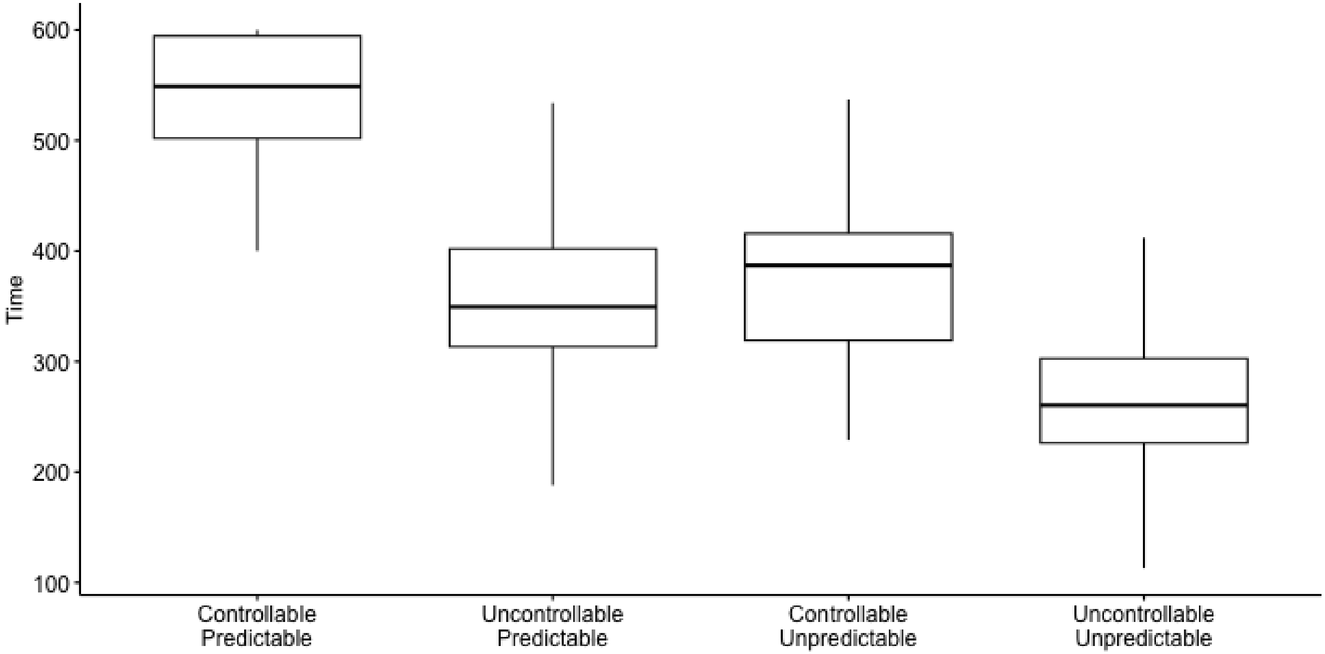
```
##   ID Predict Control Group Time
## 1  1       0       1     3  504
## 2  2       0       0     1  443
## 3  3       1       1     4  292
## 4  4       0       1     3  398
## 5  5       1       1     4  119
## 6  6       0       0     1  545
```

```r
class(rotate$Group)
```

```
## [1] "integer"
```

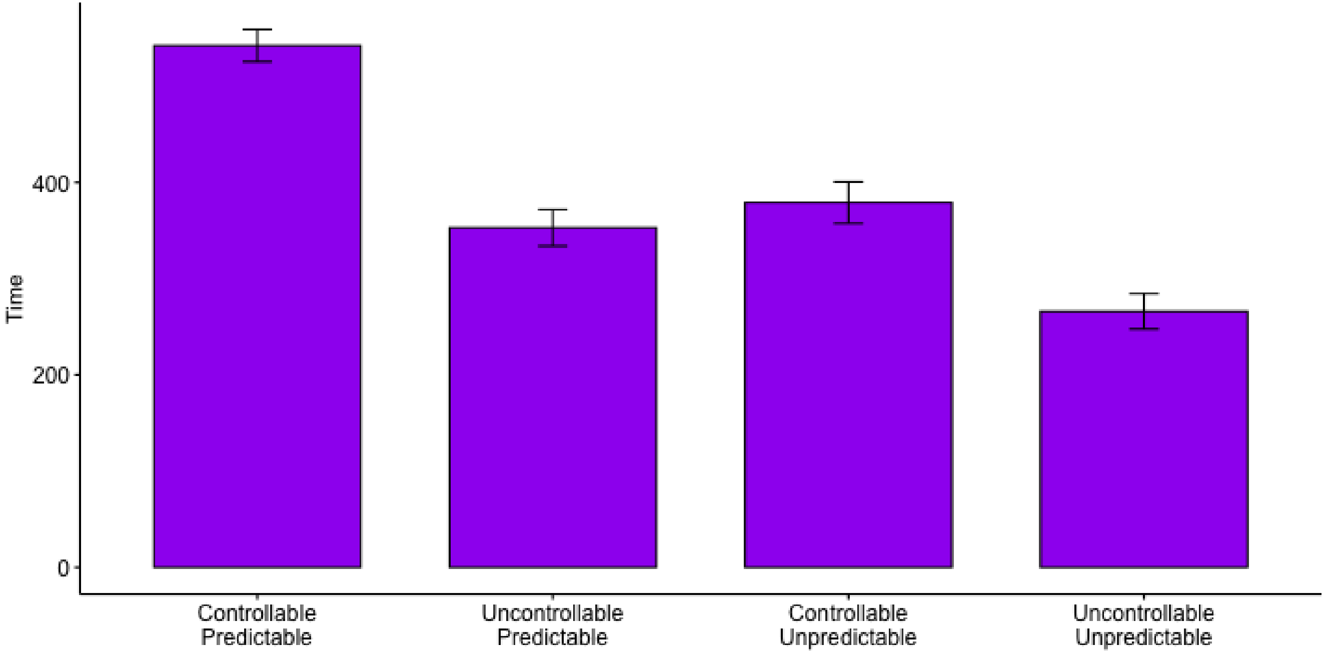
```r
rotate$Group_lab = factor(rotate$Group,
                          labels = c("Controllable\nPredictab
                                     "Uncontrollable\nPredict
                                     "Controllable\nUnpredict
                                     "Uncontrollable\nUnpredi
```

```r
library(ggpubr)
ggboxplot(data = rotate, x = "Group_lab", y = "Time", xlab =
```



Performance degrades with either uncontrollable noise or unpredictable noise. Noise that is both uncontrollable and unpredictable is particularly disruptive.

```
ggbarplot(data = rotate, x = "Group_lab", y = "Time", xlab =
```



Addition of the confidence intervals indicates that the two extreme conditions are likely different from all of the other conditions.

Group_lab	N	Mean	SD
Controllable Predictable	47	542.04	57.11
Uncontrollable Predictable	52	352.85	67.98
Controllable Unpredictable	45	379.04	71.85
Uncontrollable Unpredictable	56	265.98	68.68

The groups differ in their sample sizes, which can easily occur with free random assignment. There are advantages to equal sample sizes, so researchers often restrict random assignment to insure equal sample sizes across conditions.

Hard to tell if the variability in the four groups is homogeneous.

Hypothesis

Unlike regression, the ANOVA framework has a single hypothesis test. This is equivalent to the omnibus test of a multiple regression model.

Regression

$$H_0 : \rho^2_{Y\hat{Y}} = 0$$

$$H_1 : \rho^2_{Y\hat{Y}} > 0$$

ANOVA

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$$

$$H_1 : \text{Not that } \mu_1 = \mu_2 = \mu_3 = \mu_4$$

This can occur in a number of ways.

ANOVA

The total variability of all of the data, regardless of group membership, can be expressed as:

$$Var(Y) = \frac{1}{N}\sum_{k=1}^{G}\sum_{i=1}^{N_k}(Y_{ik} - \bar{Y})^2$$

for G groups and N_k participants within groups.

ANOVA

We are interested in the numerator of this variance equation, known as the **total sum of squares**:

$$SS_{tot} = \sum_{k=1}^{G} \sum_{i=1}^{N_k} (Y_{ik} - \bar{Y})^2$$

It's worth noting that this is just a more complicated way of expressing total sums of squares, but in a way that will be useful for thinking about how we partition sums of squares later.

ANOVA

We already know from regression that the deviation of a score from the grand mean is the sum of two independent parts. In regression these parts are the deviation of the actual score from the predicted score, and the deviation of the predicted value from the grand mean.

$$Y_i - \bar{Y} = (Y_i - \hat{Y}_i) + (\hat{Y} - \bar{Y}_i)$$

In ANOVA, this holds true, but we express these relationships by referring to each Y within a group, and instead of "predicted value" we talk about group means. So now the parts are the deviation of the score from its group mean, and the deviation of that group mean from the grand mean. Why do we substitute "group mean" for "predicted value"?

$$Y_{ik} - \bar{Y} = (Y_{ik} - \bar{Y}_k) + (\bar{Y}_k - \bar{Y})$$

Each deviation score has a within-group part and a between-group part. These separate parts can be squared and summed, giving rise to two other sums of squares.

One part, the within-groups sum of squares, represents squared deviations of scores around the group means:

$$SS_w = \sum_{k=1}^{G} \sum_{i=1}^{N_k} (Y_{ik} - \bar{Y}_k)^2$$

The other, the between-groups sum of squares, represents deviations of the group means around the grand mean:

$$SS_b = \sum_{k=1}^{G} N_k (\bar{Y}_k - \bar{Y})^2$$

Sums of squares are central to ANOVA. They are the building blocks that represent different sources of variability in a research design. They are additive, meaning that the SS_{tot} can be partitioned, or divided, into independent parts.

The total sum of squares is the sum of the between-groups sum of squares and the within-groups sum of squares.

$$SS_{tot} = SS_b + SS_w$$

The relative magnitude of sums of squares, especially in more complex designs, provides a way of identifying particularly large and important sources of variability.

If the null hypothesis is true, then variability among the means should be consistent with the variability of the data.

$$\hat{\sigma}^2_{\bar{Y}} = \frac{\hat{\sigma}^2_Y}{N}$$

In other words, if we have an estimate of the variance of the means, we can transform that into an estimate of the variance of the scores, provided the only source of mean variability is sampling variability (the null hypothesis).

$$N\hat{\sigma}^2_{\bar{Y}} = \hat{\sigma}^2_Y$$

The SS_w is qualitatively giving us information that is similar to

$$\hat{\sigma}^2_Y$$

The SS_b is qualitatively giving us information that is similar to

$$N\hat{\sigma}^2_{\bar{Y}}$$

These are arrived at separately, but under the null hypothesis they should be estimates of the same thing. The only reason that SS_b would be larger than expected is if there are systematic differences among the mean, perhaps created by experimental manipulations.

Because the sums of squares are numerators of variance estimates, we can divide each by their respective degrees of freedom to get variance estimates that, under the null hypothesis, should be approximately the same.

These variance estimates are known as mean squares:

$$MS_w = \frac{SS_w}{df_w} \qquad MS_b = \frac{SS_b}{df_b}$$

$$df_w = N - G \qquad df_b = G - 1$$

ANOVA assumes homogeneity of group variances. Under that assumption, the separate group variances are pooled to provide a single estimate of within-group variability.

$$SS_w = \sum_{k=1}^{G} \sum_{i=1}^{N_k} (Y_{ik} - \bar{Y}_k)^2 = \sum_{k=1}^{G} (N_k - 1)\hat{\sigma}_k^2$$

The degrees of freedom are likewise pooled:

$$df_w = N - G = (N_1 - 1) + (N_2 - 1) + \dots$$

If data are normally distributed, then the variance is χ^2 distributed. The ratio of two χ^2 distributed variables is F distributed.

In ANOVA, the ratio of the mean squares (variance estimates) is symbolized by F and has a sampling distribution that is F distributed with degrees of freedom equal to df_b and df_w. That sampling distribution is used to determine if an obtained F statistic is unusual enough to reject the null hypothesis.

$$F = \frac{MS_b}{MS_w}$$

$$F = \frac{MS_b}{MS_w}$$

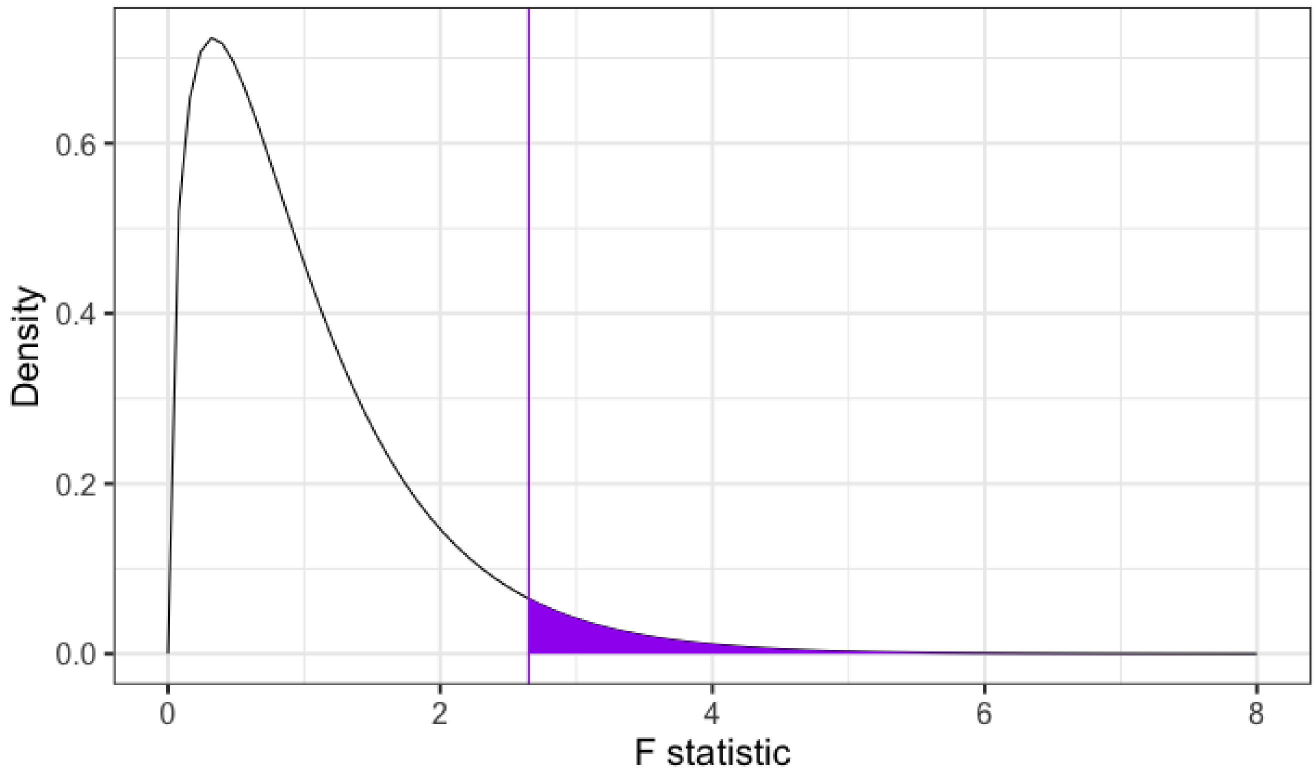
If the null hypothesis is true, F has an expected value of approximately 1.00.

If the null hypothesis is false, the numerator will be larger than the denominator because systematic, between-group differences contribute to the variance of the means, but not to the variance within groups (ideally). If F is "large enough," we will reject the null hypothesis as a reasonable description of the obtained variability.

$$F = \frac{\hat{Q} + \hat{\sigma}^2}{\hat{\sigma}^2}$$

Q is the treatment effect that, if present, increases variability among the means but does not affect the variability of scores within a group (treatment is a constant within any group).

As the treatment effect gets larger, the F statistic gets bigger. If it departs enough, we claim F to be rare or unusual. The shape of the F distribution is defined by its parameters, df_b and df_w.



For 3 and 196 degrees of freedom, an F statistic of 2.65 or greater will indicate between-groups variability relative to within-group variability that is unusual if the null hypothesis is true.

```
##              Df  Sum Sq Mean Sq F value Pr(>F)    
## Group_lab     3 2000252  666751   149.8 <2e-16 ***
## Residuals   196  872288    4450                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F statistic is highly unusual in the F distribution, assuming the null hypothesis is true. We reject the null hypothesis.

What's different?

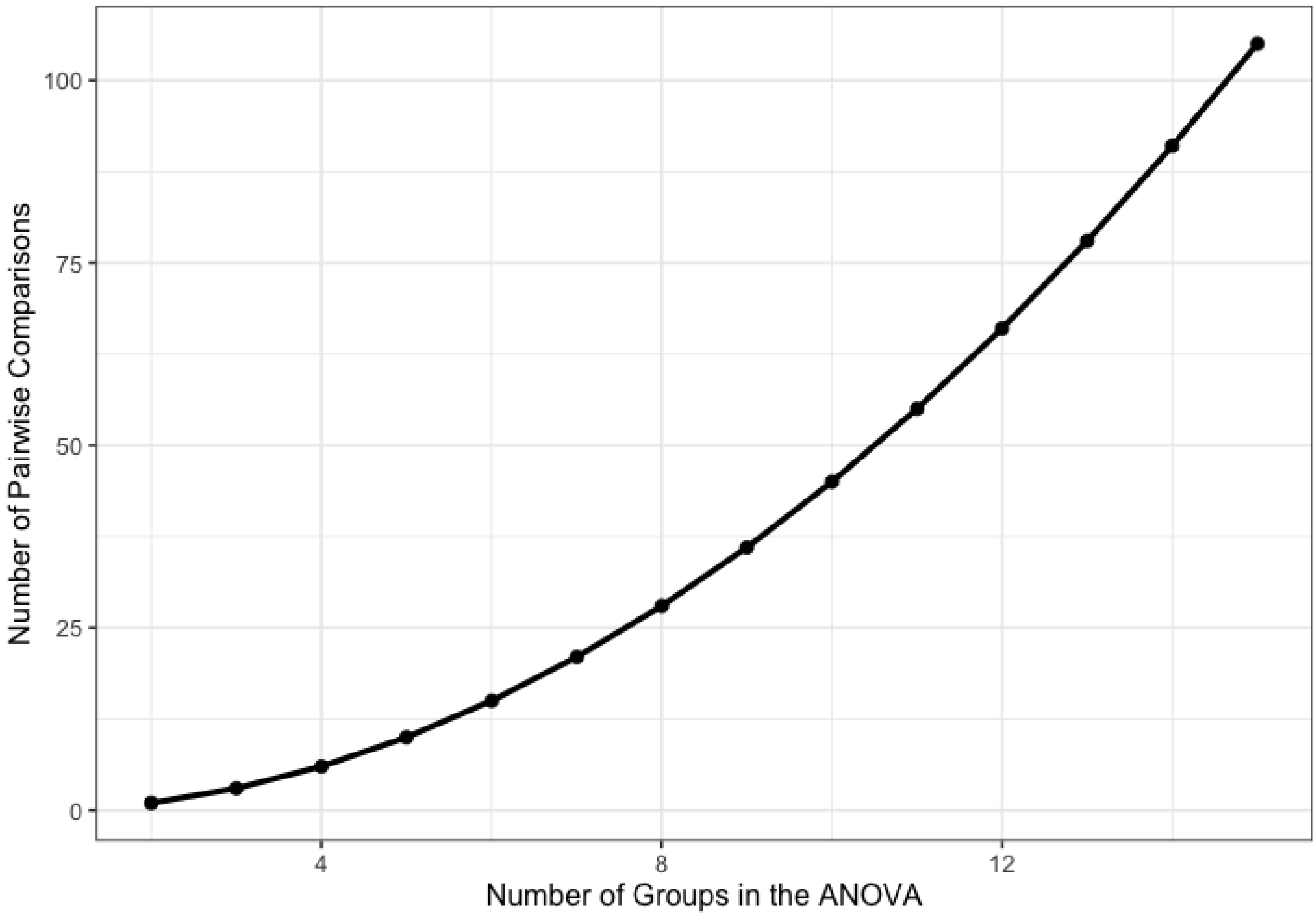
```
##              Df  Sum Sq Mean Sq F value Pr(>F)    
## Group         1 1613798 1613798   253.9 <2e-16 ***
## Residuals   198 1258741    6357                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##              Df  Sum Sq Mean Sq F value Pr(>F)    
## Group_lab     3 2000252  666751   149.8 <2e-16 ***
## Residuals   196  872288    4450                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We know the means are not equal, but the particular source of the inequality is not revealed by the F test.

One simple way to determine what is behind the significant F test is to compare each condition to all other conditions.

These pairwise comparisons can quickly grow in number as the number of conditions increases. With 4 (k) conditions, we have k(k-1)/2 = 6 possible pairwise comparisons.

As the number of groups in the ANOVA grows, the number of possible pairwise comparisons increases dramatically.



As the number of significance tests grows, and assuming the null hypothesis is true, the probability that we will make one or more Type I errors increases. To approximate the magnitude of the problem, we can assume that the multiple pairwise comparisons are independent. The probability that we don’t make a Type I error for just one test is:

$$P(\text{No Type I}, 1 \text{ test}) = 1 - \alpha$$

The probability that we don't make a Type I error for two tests is (note the reason we assume independence):

$$P(\text{No Type I}, 2 \text{ test}) = (1 - \alpha)(1 - \alpha)$$

For C tests, the probability that we make **no** Type I errors is

$$P(\text{No Type I}, C \text{ tests}) = (1 - \alpha)^C$$

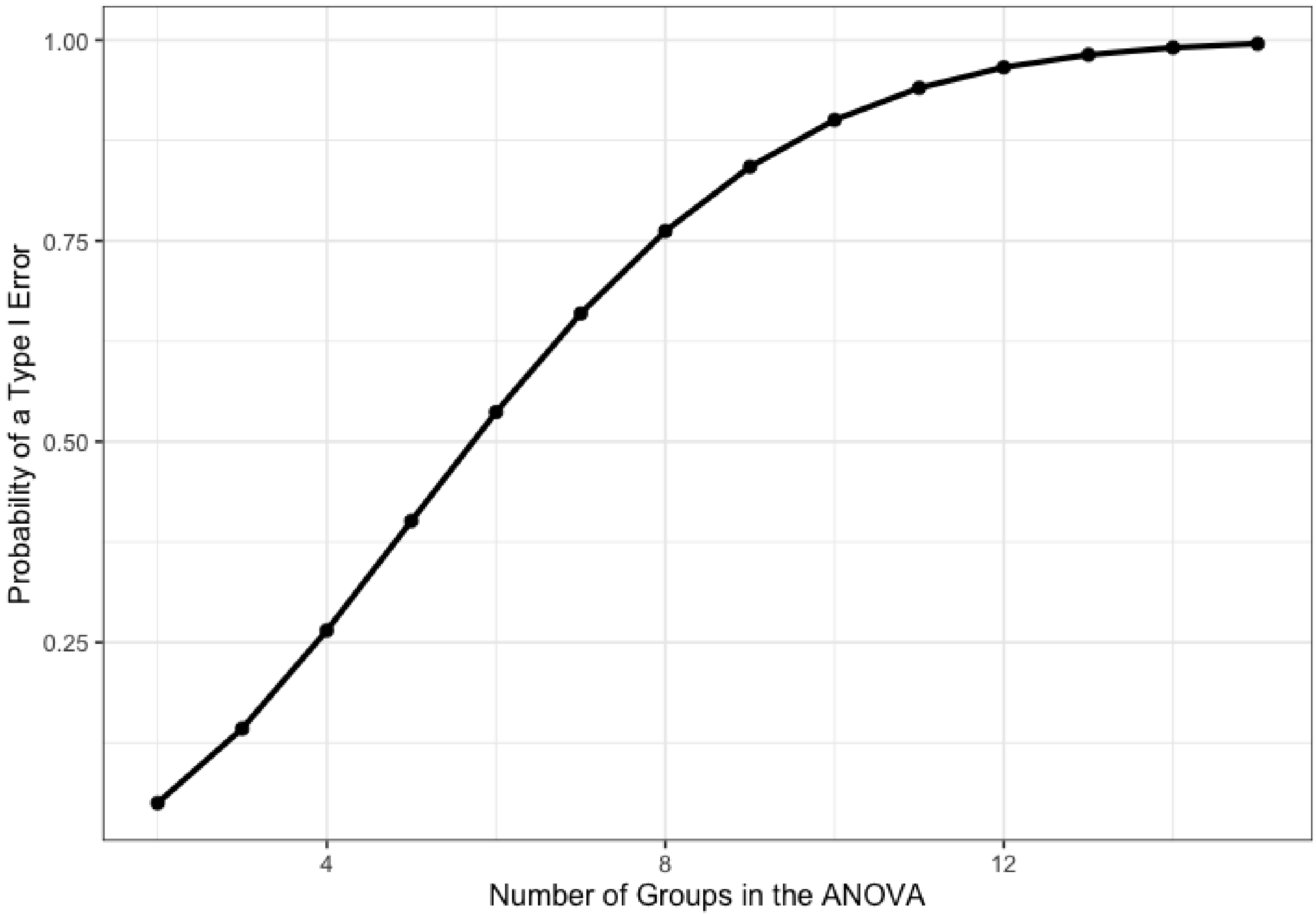
We can then use the following to calculate the probability that we make one or more Type I errors in a collection of C independent tests.

$$P(\text{At least one Type I}, C \text{ tests}) = 1 - (1 - \alpha)^C$$

As the number of comparisons grows, the inflation of the Type I error rate can be substantial.

What is this called?

Need "correction" procedures. Many exist.



Multiple comparisons, each tested with $\alpha_{per-test}$, increases the family-wise α level.

$$\alpha_{family-wise} = 1 - (1 - \alpha_{per-test})^{C}$$

Šidák showed that the family-wise α could be controlled to a desired level (e.g., .05) by changing the $\alpha_{per-test}$ to:

$$\alpha_{per-wise} = 1 - (1 - \alpha_{family-wise})^{\frac{1}{C}}$$

Bonferroni (and Dunn, and others) suggested this simple approximation:

$$\alpha_{per-test} = \frac{\alpha_{family-wise}}{C}$$

```r
library(rstatix)

rotate %>%
  pairwise_t_test(Time ~ Group,
                  p.adjust.method = "bonferroni")
```

```
## # A tibble: 6 × 9
##   .y.   group1 group2    n1    n2        p p.signif    p.adj
## * <chr> <chr>  <chr>  <int> <int>    <dbl> <chr>        <dbl>
## 1 Time  1      2         47    52 1.33e-31 ****      7.99e-31
## 2 Time  1      3         47    45 2.27e-24 ****      1.36e-23
## 3 Time  2      3         52    45 5.52e- 2 ns        3.31e- 1
## 4 Time  1      4         47    56 7.93e-52 ****      4.76e-51
## 5 Time  2      4         52    56 1.53e-10 ****      9.2 e-10
## 6 Time  3      4         45    56 5.92e-15 ****      3.55e-14
## # ℹ 1 more variable: p.adj.signif <chr>
```

The Bonferroni procedure is conservative. The most common other one is the Holm procedure.

The Holm procedure does not make a constant adjustment to each per-test α. Instead it makes adjustments in stages depending on the relative size of each pairwise p-value.

Holm Procedure

1. Rank order the p-values from largest to smallest.
2. Start with the smallest p-value. Multiply it by its rank.
3. Go to the next smallest p-value. Multiply it by its rank. If the result is larger than the previous step, keep it. Otherwise replace with the previous step adjusted p-value.
4. Repeat Step 3 for the remaining p-values.
5. Judge significance of each new p-value against $\alpha = .05$.

Holm Procedure

```
hp = rotate %>%
  pairwise_t_test(Time ~ Group,
                  p.adjust.method = "holm") %>%
  arrange(p) %>%
  mutate(Rank = 6:1, rXp = Rank*p)

hp[,c(2,3,6:11)]
```

```
## # A tibble: 6 × 8
##   group1 group2        p p.signif    p.adj p.adj.signif  Rank
##   <chr>  <chr>     <dbl> <chr>       <dbl> <chr>        <int>
## 1 1      4      7.93e-52 ****     4.76e-51 ****             6
## 2 1      2      1.33e-31 ****     6.66e-31 ****             5
## 3 1      3      2.27e-24 ****     9.07e-24 ****             4
## 4 3      4      5.92e-15 ****     1.78e-14 ****             3
## 5 2      4      1.53e-10 ****     3.07e-10 ****             2
## 6 2      3      5.52e- 2 ns       5.52e- 2 ns               1
## # ℹ 1 more variable: rXp <dbl>
## # A tibble: 6 × 8
##   group1 group2        p p.signif    p.adj p.adj.signif  Rank
##   <chr>  <chr>     <dbl> <chr>       <dbl> <chr>        <int>
## 1 1      4      7.93e-52 ****     4.76e-51 ****             6
## 2 1      2      1.33e-31 ****     6.66e-31 ****             5
## 3 1      3      2.27e-24 ****     9.07e-24 ****             4
## 4 3      4      5.92e-15 ****     1.78e-14 ****             3
## 5 2      4      1.53e-10 ****     3.07e-10 ****             2
## 6 2      3      5.52e- 2 ns       5.52e- 2 ns               1
## # ℹ 1 more variable: rXp <dbl>
```

A single adjusted per-test a would be used with the Bonferroni procedure: .05/6 = .0083. We would compare each original p-value to this new criterion and claim as significant only those that are less. Or, we could multiply the original p-values by 6 and judge them against a =.05.

Next time...

- Exam 1!

(we will cover remaining slides if time)

Contrasts

You can change which group is the reference:

```
rotate$Group = factor(rotate$Group)
contrasts(rotate$Group) = contr.treatment(n = 4, base = 3)
coef(summary(lm(Time~Group, data = rotate)))
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)   379.04444   9.944788 38.114882 1.388406e-92
## Group1        162.99811  13.913633 11.714993 2.267662e-24
## Group2        -26.19829  13.582501 -1.928827 5.519687e-02
## Group4       -113.06230  13.355564 -8.465558 5.923855e-15
```

You're not restricted to simple dummy coding. Choosing another variant of contrast codes allows you to test more specific hypotheses. There are some common coding schemes.

For example, **deviation coding** (aka **"effect coding"**):

```
contr.sum(4)
```

```
##   [,1] [,2] [,3]
## 1    1    0    0
## 2    0    1    0
## 3    0    0    1
## 4   -1   -1   -1
```

```
contrasts(rotate$Group) = contr.sum(n = 4)
coef(summary(lm(Time~Group, data = rotate))) %>% kable(., dig
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	384.98	4.73	81.31	0.000
Group1	157.06	8.35	18.80	0.000
Group2	-32.13	8.08	-3.98	0.000
Group3	-5.93	8.48	-0.70	0.485

You can create a contrast matrix that tests any number of hypotheses, like whether groups 1 and 3 are different from group 4. Rules:

1. The sum of the weights across all groups for each code variable must equal 0. (Sum down the column = 0).

2. The sum of the product of each pair of code variables, C_1C_2, must equal 0. (This is challenging when your groups are unequal sizes).

3. The difference between the value of the set of + weights and the set of neg weights should equal 1 for each column.