

Analysis of Variance (ANOVA)

Henry Scharf

MATH/STAT 571B

Module Goals:

Ch. 3 [DAE]:

Students will be able to:

1. Explain how to decompose total sum of squares into treatment and error sums of squares.
2. Conduct an analysis of variance in R including:
 - ▶ Testing $H_0 : \mu_1 = \mu_2 = \dots = \mu_a$.
 - ▶ Estimating effect sizes and interpreting in the context of the application.
 - ▶ Checking model adequacy using visual diagnostic tools.
3. Explain the relationship between an ANOVA and linear regression.
4. Estimate the sample size required to detect effects of a minimum size.

ANalysis Of VAriance (ANOVA)

ANalysis Of VAriance (ANOVA)

- ▶ We have seen several ways to compare two populations, but when we have a factor with $a > 2$ levels, something else is required.
- ▶ The naive approach of testing all possible pairs of factor levels has two big issues:
 - ▶ The number of comparisons, $\frac{a(a-1)}{2}$, can get big quickly.
 - ▶ When we conduct multiple tests, the probability of *any one of them* having a type I error can be very different than the nominal level for each test.
- ▶ ANOVA tests the null hypothesis that the factor does not have any effect on the response.
 - ▶ $H_0 : \mu_1 = \mu_2 = \dots = \mu_a$ vs. $H_1 : \mu_i \neq \mu_j$ for some i and j

Example: Plasma Etching Experiment¹

- ▶ “An engineer is...investigating the relationship between the **RF power setting** and the **etch rate**...”
- ▶ “The 20 runs should be made in random order.”
- ▶ The `sample()` function in R is a quick way to randomly order an index.

```
order <- sample(x = 1:20, size = 20, replace = FALSE)
```

```
order
```

```
## [1] 13 10 11 6 2 20 18 17 3 15 12 8 5 7 1 9 4 16 19 14
```

¹Section 3.1 DAE

Example: Plasma Etching Experiment

- ▶ The `read.csv()` function reads comma-spaced values into a dataframe. `read.table()` can be used for more general tabular formats, and there are analogous functions for other data formats.
- ▶ The `head()/tail()` function is useful for looking at the first/last few lines of the dataframe.

```
etching <- read.csv("etching.csv")
```

```
head(etching)
```

```
##   power rate  
## 1    160  575  
## 2    160  542  
## 3    160  530  
## 4    160  539  
## 5    160  570  
## 6    180  565
```

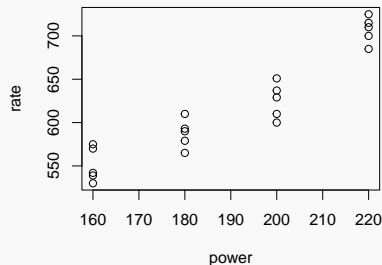
Example: Plasma Etching Experiment

- The following uses of `plot()` are equivalent.

```
plot(rate ~ power, data = etching)
plot(x = etching$power, y = etching$rate)
```

- The first uses a syntax called a “formula”, which we will use for ANOVA too.
- `boxplot()` can be used analogously.

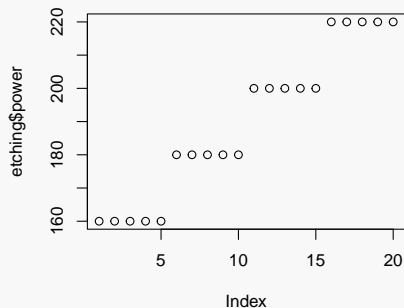
```
boxplot(rate ~ power, data = etching)
```



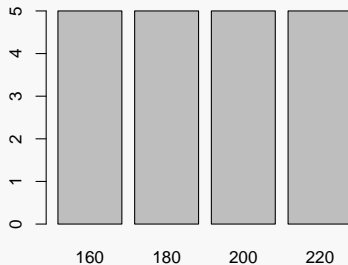
Example: Plasma Etching Experiment

- ▶ We are treating **RF power** as a categorical predictor, not a continuous one. Each level of power has its own independent etching rate.
 - ▶ $y_{ij} = \mu_i + \epsilon_{ij}$, $i = 1, \dots, a$; $j = 1, \dots, n$
- ▶ The `as.factor()` function is a way to make this explicit for variables that could be mistakenly treated as continuous.

```
plot(etching$power)
```



```
plot(as.factor(etching$power))
```



ANalysis Of VAriance (ANOVA)

- Decomposition of sums of squares:²

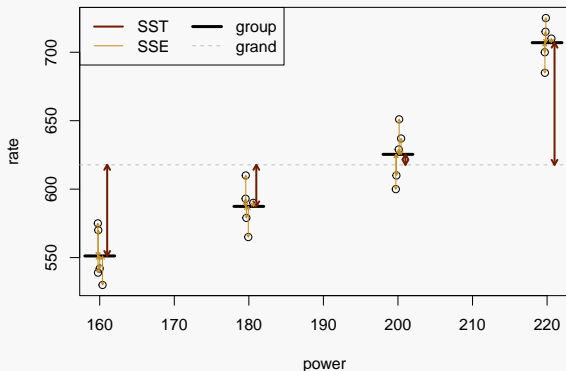
$$\underbrace{SS_T}_{SSTO} = \underbrace{SS_{\text{Treatments}}}_{SSR} + \underbrace{SS_E}_{SSE}$$
$$\sum_{i=1}^a \sum_{j=1}^n (y_{ij} - \bar{y}_{..})^2 = n \sum_{i=1}^a (\bar{y}_{i.} - \bar{y}_{..})^2 + \sum_{i=1}^a \sum_{j=1}^n (y_{ij} - \bar{y}_{i.})^2$$

- This relationship is special and a consequence of using the specific estimators $\hat{\mu}_i = \bar{y}_{i.}$ and the *squared* differences.

²with alternate notation from Kutner et al. (2004). *ALRM*.

ANalysis Of VAriance (ANOVA)

- ▶ Decomposition of sums of squares: $SS_T = SS_{\text{Treatments}} + SS_E$
- ▶ If $\frac{SS_{\text{Treatments}}/(a-1)}{SS_E/(N-a)}$ is large, then there is more variation *between* the estimated effects than *within* them.
- ▶ We have a ratio of two sums of squares, so we should expect an F sampling distribution.



Example: Plasma Etching Experiment

- The ANOVA table³ output and p-value for H_0 is easy to get with R.

```
etching_aov <- aov(rate ~ as.factor(power), data = etching)
summary(etching_aov)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
as.factor(power)	3	66871	22290	66.8	2.88e-09 ***
Residuals	16	5339	334		

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(2) What is the MSE?

³Compare with Table 3.4 in DAE.

Example: Plasma Etching Experiment (INCORRECT!)

- ▶ This code gives the wrong answer!
- ▶ **RF power** is mistakenly treated as continuous factor with a single overall trend.

```
etching_aov_WRONG <- aov(rate ~ power, data = etching)
summary(etching_aov_WRONG)
##              Df Sum Sq Mean Sq F value    Pr(>F)
## power          1  63857   63857   137.6 7.26e-10 ***
## Residuals     18   8352     464
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- (3) What is something about the summary output that would let you know something was not right?

ANOVA as linear regression

- An alternative way to denote the model $y_{ij} = \mu_i + \epsilon_{ij}$ is

$$y_{ij} = x_{1ij}\mu_1 + x_{2ij}\mu_2 + \cdots + x_{a ij}\mu_a + \epsilon_{ij}$$

$$x_{kij} = \begin{cases} 1, k = i \\ 0, k \neq i \end{cases}$$

- Let $\mu_i = \beta_i$, $\mathbf{x}'_{ij} = (0 \dots 0 \underbrace{1}_i 0 \dots 0)$, $\boldsymbol{\beta}' = (\beta_1, \dots, \beta_a)$

$$y_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta} + \epsilon_{ij}$$

- Using full matrix notation:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \mathbf{X}_{N \times a} = \begin{pmatrix} \mathbf{x}_{11} \\ \mathbf{x}_{12} \\ \vdots \\ \mathbf{x}_{an} \end{pmatrix}, \quad \boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_N)$$

ANOVA as linear regression: Plasma Etching Experiment

- ▶ The function `lm()` fits a linear regression model using least squares.
- ▶ The summary output includes the overall test of a relationships between at least one predictor and the response, which is equivalent to the ANOVA approach.

```
fit_lm <- lm(rate ~ as.factor(power), data = etching)
summary(fit_lm)
## Call:
## lm(formula = rate ~ as.factor(power), data = etching)
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      551.200      8.169  67.471  < 2e-16 ***
## as.factor(power)180   36.200     11.553   3.133  0.00642 **
## as.factor(power)200   74.200     11.553   6.422  8.44e-06 ***
## as.factor(power)220  155.800     11.553  13.485  3.73e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.27 on 16 degrees of freedom
## Multiple R-squared:  0.9261, Adjusted R-squared:  0.9122
## F-statistic: 66.8 on 3 and 16 DF, p-value: 2.883e-09
```

ANOVA as linear regression: Plasma Etching Experiment

- ▶ The standard linear regression assigns the intercepts to one category and estimates the differences between that reference category and all others.
- ▶ In this case the reference category is for **RF power** = 160W.⁴

```
coef(fit_lm)
##           (Intercept) as.factor(power)180 as.factor(power)200 as.factor(power)220
##           551.2           36.2           74.2           155.8
coef(fit_lm)[1] + coef(fit_lm)[2:4]
## as.factor(power)180 as.factor(power)200 as.factor(power)220
##           587.4           625.4           707.0
```

$$y_{ij} = \underbrace{\mu_1 + \alpha_i}_{\mu_i} + \epsilon_{ij}, \quad \alpha_i = \mu_i - \mu_1$$

⁴Verify results with Example 3.3 on p. 79 of DAE.

ANOVA as linear regression: Plasma Etching Experiment

- The following formula can be used to get the the effects τ_i in the model⁵

$$y_{ij} = \mu + \tau_i + \epsilon_{ij}$$

$$\mu_i = \mu + \tau_i = \alpha_i + \mu_1 \implies \tau_i = \alpha_i + \mu_1 - \mu$$

- (4) What indices (integers) should go in the blanks to calculate the point estimate for τ_3 (**RF power** = 200W)?

```
coef(fit_lm)[___] + coef(fit_lm)[___] - mean(etching$rate)
```

```
## as.factor(power)200
```

```
## 7.65
```

⁵eq. 3.2 in DAE; example 3.3 on p.79

Model Adequacy Checking

Model Adequacy Checking

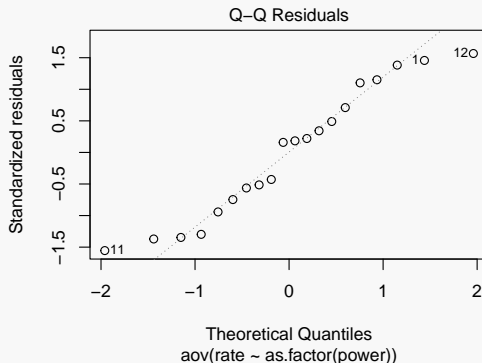
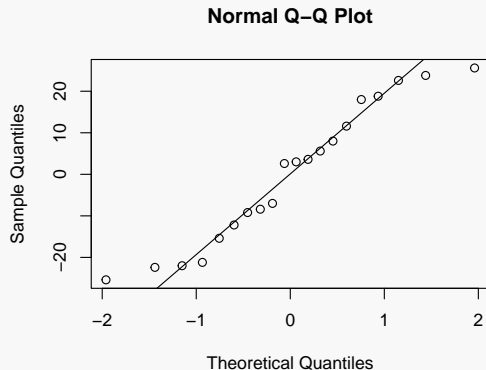
- ▶ Recall our modeling assumptions:
 1. ϵ_{ij} are **normally distributed** with
 2. constant variance, σ^2 , and
 3. are mutually **independent**.
- ▶ Define residuals $e_{ij} = y_{ij} - \hat{y}_{ij}$, where $\hat{y}_{ij} = \hat{\mu}_i = \bar{y}_{i.}$
- ▶ e_{ij} are therefore also mean-zero normal (but *not* independent).
- ▶ Residuals should be **structureless**: no obvious patterns, and no correlation related to time/space/other indices.

Model Adequacy Checking: Normality

- A QQ-plot with a normal reference distribution (“normal probability plot” in DAE) is a good way to visually evaluate the assumption of normality.

```
qqnorm(resid(etching_aov))  
qqline(resid(etching_aov))
```

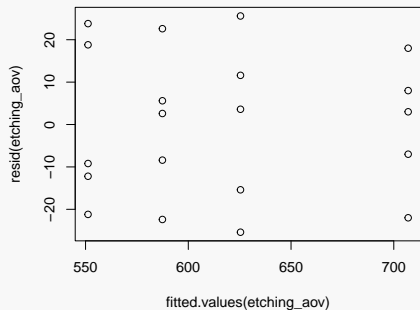
```
plot(etching_aov, which = 2)
```



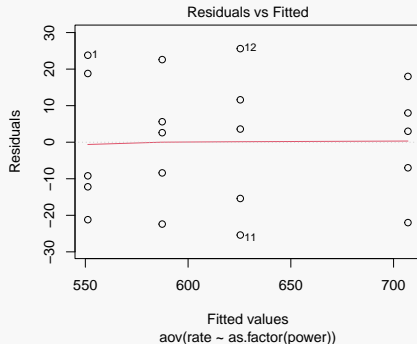
Model Adequacy Checking: Constant Variances

- ▶ Plotting residuals on the vertical axis against various other variables on the horizontal axis can reveal structure/patterns that conflict with model assumptions.
- ▶ No obvious signs of heteroskedasticity from these figures.

```
plot(fitted.values(etching_aov),  
     resid(etching_aov))
```



```
plot(etching_aov, which = 1)
```

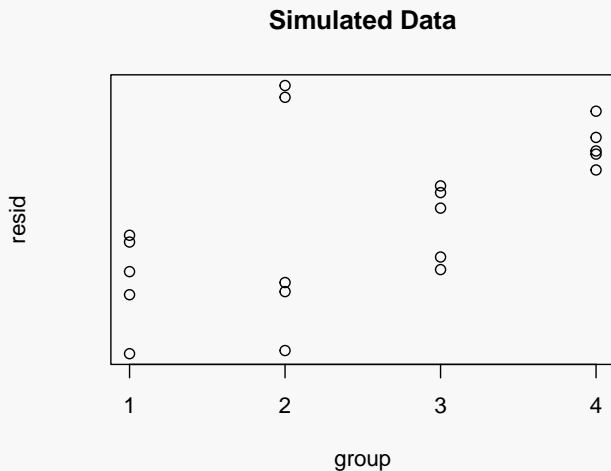


Model Adequacy Checking: Constant Variances

- ▶ Visual diagnostics are often sufficient, but formal tests for model assumptions are sometimes available.
- ▶ Bartlett's test is based on a statistic with a sampling distribution that is approximately chi-square if normality can be assumed.
- ▶ The modified Levene test is robust to the assumption of normality.

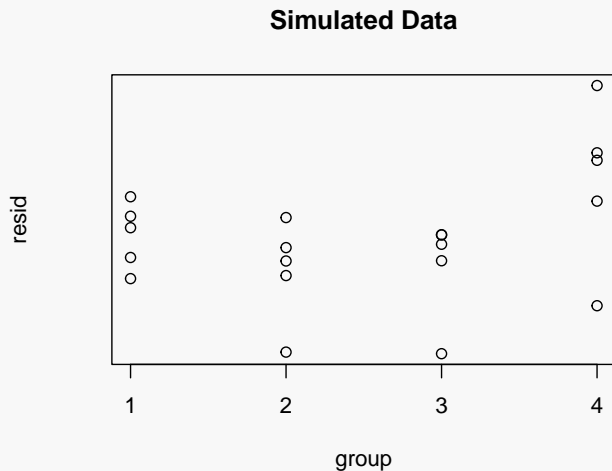
```
bartlett.test(rate ~ as.factor(power), data = etching)
##
##  Bartlett test of homogeneity of variances
##
## data:  rate by as.factor(power)
## Bartlett's K-squared = 0.43349, df = 3, p-value = 0.9332
library(car)
## Loading required package: carData
leveneTest(rate ~ as.factor(power), data = etching)
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value Pr(>F)
## group    3  0.1959 0.8977
##           16
```

Can you spot the heterogeneity better than Bartlett or Levene?



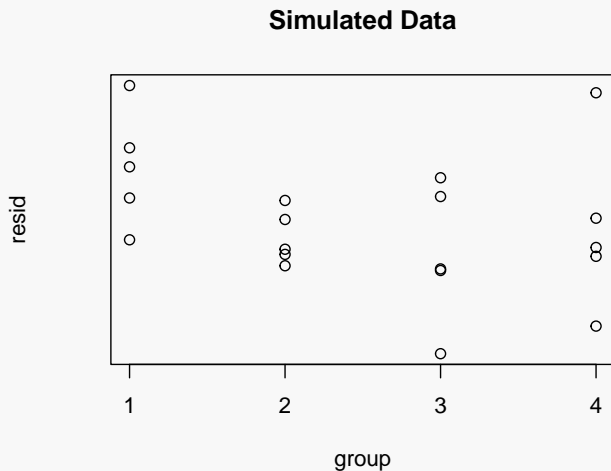
(5) Do you see evidence of heteroskedasticity (non-constant variance)?

Can you spot the heterogeneity better than Bartlett or Levene?



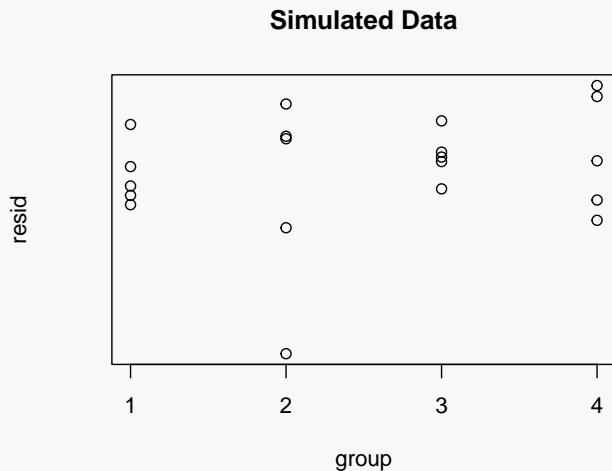
(6) Do you see evidence of heteroskedasticity (non-constant variance)?

Can you spot the heterogeneity better than Bartlett or Levene?



(7) Do you see evidence of heteroskedasticity (non-constant variance)?

Can you spot the heterogeneity better than Bartlett or Levene?



(8) Do you see evidence of heteroskedasticity (non-constant variance)?

Effect on inference?

Repeat the following 5000 times:

1. Simulate data from null model with $\sigma_1^2 = \sigma_3^2 = 10^2$ and $\sigma_2^2 = \sigma_4^2 = 20^2$.
2. Compute ANOVA-based p-value (which assumes common variance).

Then:

3. Determine proportion of times H_0 was incorrectly rejected at level $\alpha = 0.05$.
4. Compare to nominal type I error rate.

```
sigmas <- c(10, 20, 10, 20)
set.seed(2023)
typeI <- sapply(1:5000, function(rep){
  fake_data <- data.frame(y = c(sapply(1:4, function(a) rnorm(n_reps, sd = sigmas[a]))),
                           group = as.factor(rep(1:n_groups, rep(n_reps, n_groups))))
  summary(aov(y ~ group, data = fake_data))[[1]]$`Pr(>F)`[1] < 0.05
})
mean(typeI)
## [1] 0.0706
```

- (9) Is our test too liberal (rejects too often) or too conservative (fails to reject too often)?

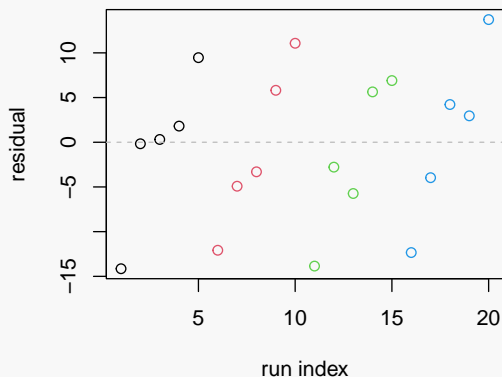
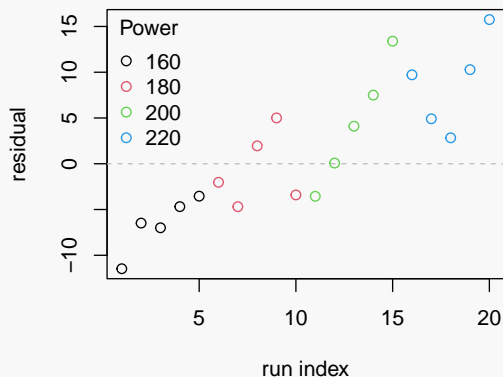
Model Adequacy Checking: Independence

- ▶ “If data have been collected on any other variables that might possibly affect the response, the residuals should be plotted against these variables.”⁶
- ▶ Example: Suppose the order of the etchings had not been randomized and the experimenter had done all runs for each power level together. Confounding effects could invalidate the independence assumption, and structure would likely be visible in the residuals plotted against run order.

⁶Ch. 3.4.4 DAE

Model Adequacy Checking: Independence

- Example: Suppose the order of the etchings had not been randomized...



(10) What potential issue do you see with each hypothetical figure?

- Non-independence can be very hard to adequately address.

Remedial Measures: Transformations of Response

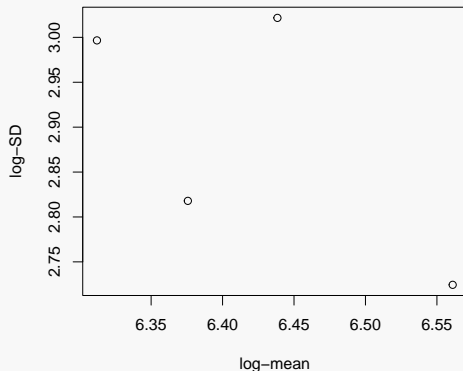
- ▶ One easy to implement remedial measure that can address heterogeneity is a transformation of the response.
- ▶ Common transformations to try: \sqrt{y} , $\log(y)$, $1/y$
- ▶ Remember that effects must be interpreted for transformed response.
- ▶ Plotting $\log(\bar{y}_i)$ on the horizontal axis and $\log(S_i)$ on the vertical axis can (sometimes) help with the choice of transformation.

```
sds <- aggregate(rate ~ power, data = etching, FUN = sd)
sds
##    power    rate
## 1    160 20.01749
## 2    180 16.74216
## 3    200 20.52559
## 4    220 15.24795
means <- aggregate(rate ~ power, data = etching, FUN = mean)
```

Remedial Measures: Transformations of Response

- ▶ Plotting $\log(\bar{y}_{i.})$ on the horizontal axis and $\log(S_i)$ on the vertical axis can (sometimes) help with the choice of transformation.
- ▶ $y^* = y^{1-\text{slope}}$

```
par(mar = c(4, 4, 1, 1))  
plot(log(means$rate), log(sds$rate), xlab = "log-mean", ylab = "log-SD")
```



Remedial Measures: Non-parametric Inference

- ▶ **Non-parametric** tests generally trade some power for a relaxation of assumptions.
- ▶ If parametric assumptions are in fact valid, then non-parametric inference is usually **conservative** (sometimes VERY conservative).
- ▶ **Kruskal–Wallis** is a non-parametric test for $H_0 : \mu_1 = \dots = \mu_a$: No assumption of normality.
- ▶ Based on *ranks* instead of raw observations.
- ▶ Overall F -test is fairly robust to assumptions, especially under a balanced design, but other aspects of analysis may not be.

```
kruskal.test(rate ~ as.factor(power), data = etching)
##
##  Kruskal-Wallis rank sum test
##
## data:  rate by as.factor(power)
## Kruskal-Wallis chi-squared = 16.907, df = 3, p-value = 0.0007386
```

Practical Interpretation

Group Effects

- ▶ The `predict()` method in R is very useful in many contexts.
- ▶ Takes two arguments:
 - ▶ `object`: fitted model (e.g., `etching_aov`)
 - ▶ `newdata`: dataframe with columns exactly named to match model variables

```
pred_data <- data.frame(power = c(160, 180, 200, 220))
effects_CI <- predict(etching_aov, newdata = pred_data, interval = "confidence")
effects_CI
```

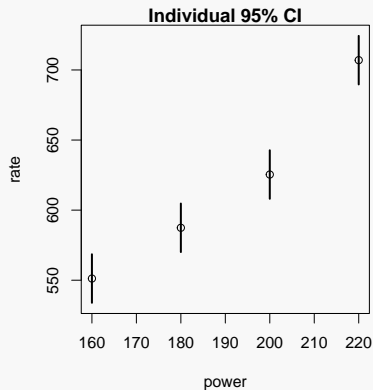
	<i>fit</i>	<i>lwr</i>	<i>upr</i>
## 1	551.2	533.8815	568.5185
## 2	587.4	570.0815	604.7185
## 3	625.4	608.0815	642.7185
## 4	707.0	689.6815	724.3185

(11) Why do you think the follow code throws an error?

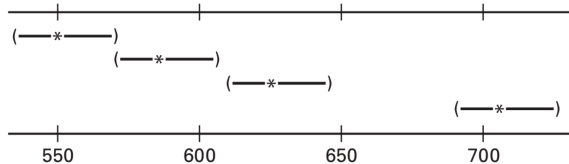
```
pred_data <- data.frame(power = c(160, 180, 200, 220, 240))
effects_CI <- predict(etching_aov, newdata = pred_data, interval = "confidence")
```

Group Effects: Figures

```
par(mar = c(4, 4, 1, 1)) ## default margins in R are generous
plot(pred_data$power, effects_CI[, 'fit'], ylim = range(effects_CI),
     xlab = "power", ylab = "rate", main = "Individual 95% CI")
segments(x0 = pred_data$power, y0 = effects_CI[, 'lwr'],
        y1 = effects_CI[, 'upr'], lwd = 2)
```



Individual 95% CIs For Mean Based on Pooled StDev



DAE p.104

Comparing Pairs of Treatment Means

- ▶ “Snooping”⁷ through the results to find standout groups has to be done carefully.
- ▶ “Significant” differences are often interpreted with respect to individual error rates, but in fact should be interpreted with respect to **experimentwise** error rates.
- ▶ There are several methods for addressing the issue of “multiple comparisons” such as:
 - ▶ Fisher’s Least Significant Difference
 - ▶ Bonferroni correction
 - ▶ Tukey’s Honest Significant Difference

⁷In my experience, this term generally always has a negative connotation.

Fisher's Least Significant Difference

- ▶ Do a bunch of pairwise t -tests (without pooling variances).
- ▶ Does **NOT** control experimentwise error rate!

```
t.test(rate ~ power, data = etching[etching$power %in% c(160, 180), ], conf.level = 0.99)
## Welch Two Sample t-test
##
## data:  rate by power
## t = -3.1018, df = 7.7575, p-value = 0.01519
```

```
t.test(rate ~ power, data = etching[etching$power %in% c(160, 200), ], conf.level = 0.99)
## t = -5.787, df = 7.995, p-value = 0.0004122
```

```
t.test(rate ~ power, data = etching[etching$power %in% c(180, 200), ], conf.level = 0.99)
## t = -3.2079, df = 7.6894, p-value = 0.01313
```

```
t.test(rate ~ power, data = etching[etching$power %in% c(180, 220), ], conf.level = 0.99)
## t = -11.81, df = 7.9311, p-value = 2.601e-06
```

```
t.test(rate ~ power, data = etching[etching$power %in% c(200, 220), ], conf.level = 0.99)
## t = -7.136, df = 7.3842, p-value = 0.0001454
```

Fisher's Least Significant Difference

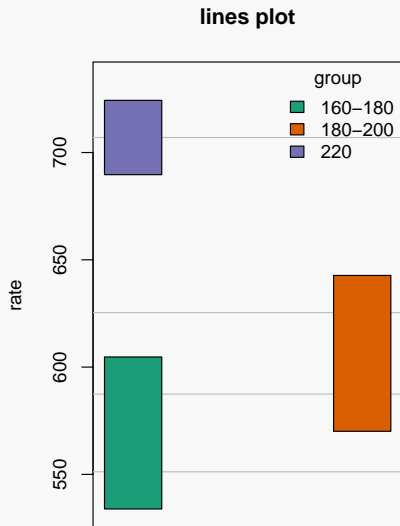
Pair	p-value
160-180	0.015
160-200	0.00041
180-200	0.013
180-220	0.0000026
200-220	0.00015

- (12) Why do we not need to see the p-value for the pair 160-220?
- (13) How would you “collect” these four levels based on the p-values?

“Lines plot”: A useful qualitative figure?

- ▶ Popularized by SAS, these plots give a qualitative visualization groups means by whether or not they are significantly different from each other.
- ▶ No simple implementation in R.
 - ▶ Maybe for good reason? The SAS ones *look* quantitative, but they're really not.
- ▶ Plot is based on a particular method for pairwise comparisons (e.g., Fisher's LSD).
- ▶ Version to the right uses extremes of CIs for each level in collection.
 - ▶ No sound statistical basis for doing this that I know of.
 - ▶ Code is a little clunky.^a

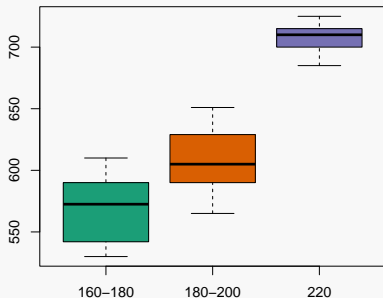
^abut available upon request



“Lines plot”

- A simpler to code and probably equally effective version.

```
colors <- RColorBrewer::brewer.pal(3, "Dark2")  
par(mar = c(4, 4, 1, 1))  
boxplot(list("160-180" = etching$rate[etching$power %in% c(160, 180)],  
            "180-200" = etching$rate[etching$power %in% c(180, 200)],  
            "220" = etching$rate[etching$power %in% c(220)]), col = colors)
```



Bonferroni

- ▶ Controls experimentwise error rate, but can be very conservative.
- ▶ $\binom{a}{2} = \frac{a(a-1)}{2}$ possible pairwise tests $\implies \alpha^* = \frac{\alpha}{a(a-1)/2}$.
- ▶ $0.01/6 \approx 0.00167$

Pair	p-value
160-180	0.015
160-200	0.00041
180-200	0.013
180-220	0.0000026
200-220	0.00015

(14) How would our conclusions change about pairwise differences after using Bonferroni correction?

Tukey's HSD

- Controls experimentwise error rate and less conservative than Bonferroni.

```
etching_hsd <- TukeyHSD(etching_aov, plot(etching_hsd),  
                        conf.level = 0.99)
```

```
etching_hsd
```

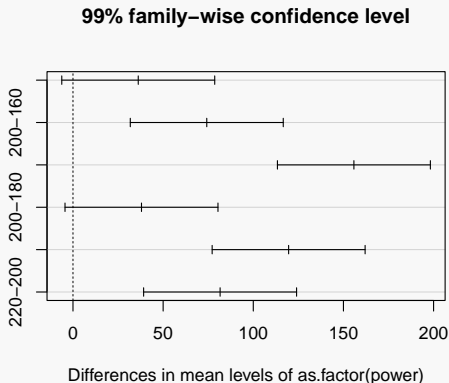
```
##    Tukey multiple comparisons of means
```

```
##      99% family-wise confidence level
```

```
##
```

```
## $`as.factor(power)`
```

##		diff	lwr	upr	p adj
##	180-160	36.2	-6.21498	78.61498	0.0294279
##	200-160	74.2	31.78502	116.61498	0.0000455
##	220-160	155.8	113.38502	198.21498	0.0000000
##	200-180	38.0	-4.41498	80.41498	0.0215995
##	220-180	119.6	77.18502	162.01498	0.0000001
##	220-200	81.6	39.18502	124.01498	0.0000146



Dunnett's procedure

- ▶ For comparing means to a single control level (DAE p.101).
- ▶ Implemented in `DescTools` package: `DunnettTest()`
- ▶ Demo: scrutinizing packages
 - ▶ `Version`: fine-grain numbering suggests regular maintenance
 - ▶ `Published`: within the last year or two is best
 - ▶ `Author`: recognize anyone? many authors sometimes a good sign. Google them :)
 - ▶ `URL`: sometimes gives more info about package and team
 - ▶ `Vignettes`: good sign if these exist and are carefully prepared

Contrasts

- ▶ More general comparing group means, we might be interested in comparing linear combinations of group means.
- ▶ Example: $H_0 : \frac{\mu_1 + \mu_2}{2} = \frac{\mu_3 + \mu_4}{2}$
 - ▶ Equivalent representation: $\mu_1 + \mu_2 - \mu_3 - \mu_4 = \sum_{i=1}^4 c_i \mu_i = \mathbf{c}' \boldsymbol{\mu} = 0$
where $\mathbf{c}' = (1, 1, -1, -1)$
- ▶ When $\sum_{i=1}^a c_i = 0$, this is called a contrast.

Scheffé's Method

- ▶ Controls experimentwise error rate for *all contrasts*.
- ▶ More conservative than Tukey's HSD for pairwise means only.
- ▶ $C = \sum_{i=1}^a c_i \bar{y}_i$.
- ▶ $S_C = \sqrt{MSE \sum_{i=1}^a (c_i^2 / n_i)}$
- ▶ Compare $|C|$ to $S_C \sqrt{(a-1)F_{\alpha, a-1, N-a}}$

```
c <- c(1, -1, 0, 0)
means <- aggregate(rate ~ power, data = etching, FUN = mean)
ybars <- means$rate
C <- sum(c * ybars)
S_C <- sqrt(sigma(etching_aov)^2 * sum(c^2 / 5))
C
## [1] -36.2
S_C * sqrt(3 * qf(1 - 0.01, 3, 16))
## [1] 46.03492
```

Power Analysis: Determining Sample Size

Determining Sample Size

- ▶ Heuristic: Run a hypothetical experiment + analysis and see how it goes.
- ▶ Brute force:
 - ▶ Simulate many (thousands, maybe much more) hypothetical data sets,
 - ▶ analyze each one,
 - ▶ compute proportion of data sets for which type II error occurs.
 - ▶ Adjust sample size until satisfactory power is reached.
- ▶ To avoid all the computation, need to know sampling distribution under particular alternative hypothesis (often “non-central” distributions).

Simulation: DAE Example 3.10 p.107

- ▶ Desired power (probability of detecting difference) is 0.90.
- ▶ Hypothetical group means $\mu_1 = 575$, $\mu_2 = 600$, $\mu_3 = 650$, $\mu_4 = 675$ and variance $\sigma = 25$.
- ▶ Maximum error type I rate of $\alpha = 0.01$.

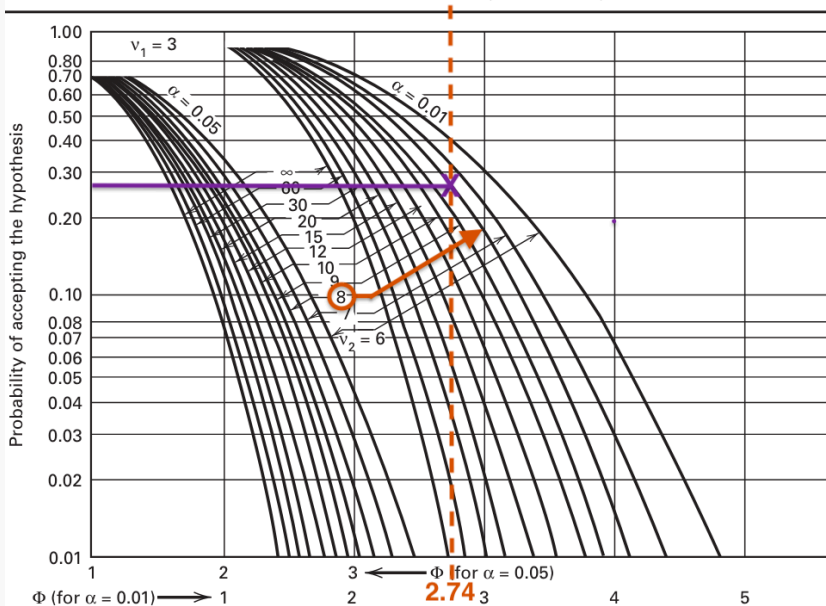
```
n <- 3
mus <- c(mu1 = 575, mu2 = 600, mu3 = 650, mu4 = 675)
alpha <- 0.01
sigma <- 25
typeII <- sapply(1:2000, function(i){
  sim_data <- data.frame(power = as.factor(rep(mus, rep(n, 4))),
                        rate = c(sapply(mus, function(mu) rnorm(n, mu, sd = sigma))))
  sim_aov <- aov(rate ~ power, data = sim_data)
  summary(sim_aov)[[1]]$`Pr(>F)`[1] < alpha
})
mean(typeII)
## [1] 0.7495
```

Simulation: DAE Example 3.10 p.107

```
n <- 4
typeII <- sapply(1:2000, function(i){
  sim_data <- data.frame(power = as.factor(rep(mus, rep(n, 4))),
                        rate = c(sapply(mus, function(mu) rnorm(n, mu, sd = sigma))))
  sim_aov <- aov(rate ~ power, data = sim_data)
  summary(sim_aov)[[1]]$`Pr(>F)`[1] < alpha
})
mean(typeII)
## [1] 0.9545
```

```
n <- 5
typeII <- sapply(1:2000, function(i){
  sim_data <- data.frame(power = as.factor(rep(mus, rep(n, 4))),
                        rate = c(sapply(mus, function(mu) rnorm(n, mu, sd = sigma))))
  sim_aov <- aov(rate ~ power, data = sim_data)
  summary(sim_aov)[[1]]$`Pr(>F)`[1] < alpha
})
mean(typeII)
## [1] 0.9975
```


V Operating Characteristic Curves for the Fixed Effects Model Analysis of Variance (*Continued*)



Power calculations in R

```
power.anova.test(groups = 4, n = 4, between.var = var(mus),  
                 within.var = sigma^2, sig.level = 0.01)  
  
##  
##      Balanced one-way analysis of variance power calculation  
##  
##           groups = 4  
##           n = 4  
##      between.var = 2083.333  
##      within.var = 625  
##      sig.level = 0.01  
##           power = 0.9621239  
##  
## NOTE: n is number in each group
```

Details

Exactly one of the parameters `groups`, `n`, `between.var`, `power`, `within.var`, and `sig.level` must be passed as `NULL`, and that parameter is determined from the others. Notice that `sig.level` has non-`NULL` default so `NULL` must be explicitly passed if you want it computed.

Power calculations in R

```
power.anova.test(groups = 4, between.var = var(mus),  
                  within.var = sigma^2, sig.level = 0.01, power = 0.90)  
  
##  
##      Balanced one-way analysis of variance power calculation  
##  
##           groups = 4  
##           n = 3.520243  
##      between.var = 2083.333  
##      within.var = 625  
##      sig.level = 0.01  
##           power = 0.9  
##  
## NOTE: n is number in each group
```

Simulation: DAE example bottom p. 107

- ▶ Want to reject the null if *any two levels* differ by at least 75
- ▶ Move creation of mus inside the *apply() function so that they are generated randomly each time.

```
n <- 5
typeII <- sapply(1:2000, function(i){
  mus <- c(mu1 = 525, mu2 = 600, mu3 = runif(1, 525, 600), mu4 = runif(1, 525, 600))
  sim_data <- data.frame(power = as.factor(rep(mus, rep(n, 4))),
                        rate = c(sapply(mus, function(mu) rnorm(n, mu, sd = sigma))))
  sim_aov <- aov(rate ~ power, data = sim_data)
  summary(sim_aov)[[1]]$`Pr(>F)`[1] < alpha
})
mean(typeII)
## [1] 0.8935
```

Simulation: Maximum difference of 2

```
typeII <- sapply(1:2000, function(i){  
  n <- 6  
  mus <- c(mu1 = 525, mu2 = 600, mu3 = runif(1, 525, 600), mu4 = runif(1, 525, 600))  
  alpha <- 0.01  
  sigma <- 25  
  sim_data <- data.frame(power = as.factor(rep(mus, rep(n, 4))),  
                        rate = c(sapply(mus, function(mu) rnorm(n, mu, sd = sigma))))  
  sim_aov <- aov(rate ~ power, data = sim_data)  
  summary(sim_aov)[[1]]$`Pr(>F)`[1] < alpha  
})  
mean(typeII)  
## [1] 0.9665
```

- ▶ Conclusion: at least $n = 6$ required (agrees with DAE).
- ▶ Simulation approach admits lots of flexibility in hypothetical scenarios.