Analysis of Variance

JEFF GILL
Distinguished Professor
Departments of Government and Mathematics & Statistics
American University

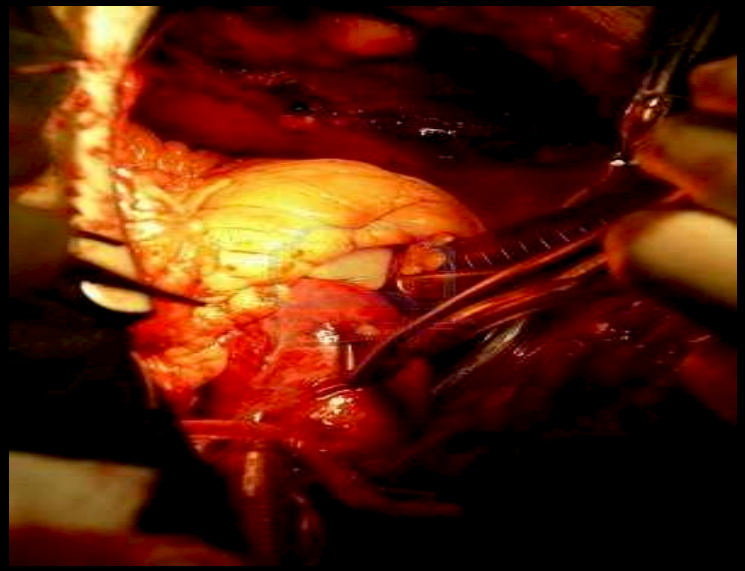
Two Purposes

- Classical ANOVA As Additive Data Decomposition. A linear model with indicator variables tool to understand relative sources of variation in a dataset, eg. does some categorization make sense statistically? Use the `aov` command in `R`.
- Classical ANOVA For Model Comparison. If a multilevel model has already been fit, which sources of variation dominate, eg. does the between-group variance exceed the within-group variance. Use the `anova` command in `R`.
- Both purposes are important.
- Note: a better name would be *analysis of variance of means*.

Cardiac Bypass Surgery Example

- 22 patients undergoing cardiac bypass surgery are randomized into three ventilation groups:
  - Group 1: a 50% nitrous oxide, 50% oxygen mixture continuously for 24 hours after procedure.
  - Group 2: a 50% nitrous oxide, 50% oxygen mixture only during the procedure.
  - Group 3: 35-50% oxygen for 24 hours after procedure.
- Red cell folate levels (RBC Folate) measured in 400 ng/mL (nanograms per milliliter):

Group 1	Group 2	Group 3
243	206	241
251	210	258
275	226	270
291	249	293
347	255	328
354	273	
380	285	
392	295	
	309	



Cardiac Bypass Surgery, Assumptions and Model

- y_{ij} is the ith observation from the j group, modeled as:

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

with $\epsilon_{ij} \sim N(0, \sigma^2)$.

- Since this model is overparameterized relative to the data, we need to assume that the group means sum to zero: $\sum_{j=1}^{J} \alpha_j = 0$.

- The standard null hypothesis, using an F-test with with $J-1$ and $N-J$ DF, is equality of groups:

$$H_0{:}\alpha_1 = \alpha_2 = \cdots = \alpha_J$$

meaning that the between-group and the within-group variances equal σ^2.

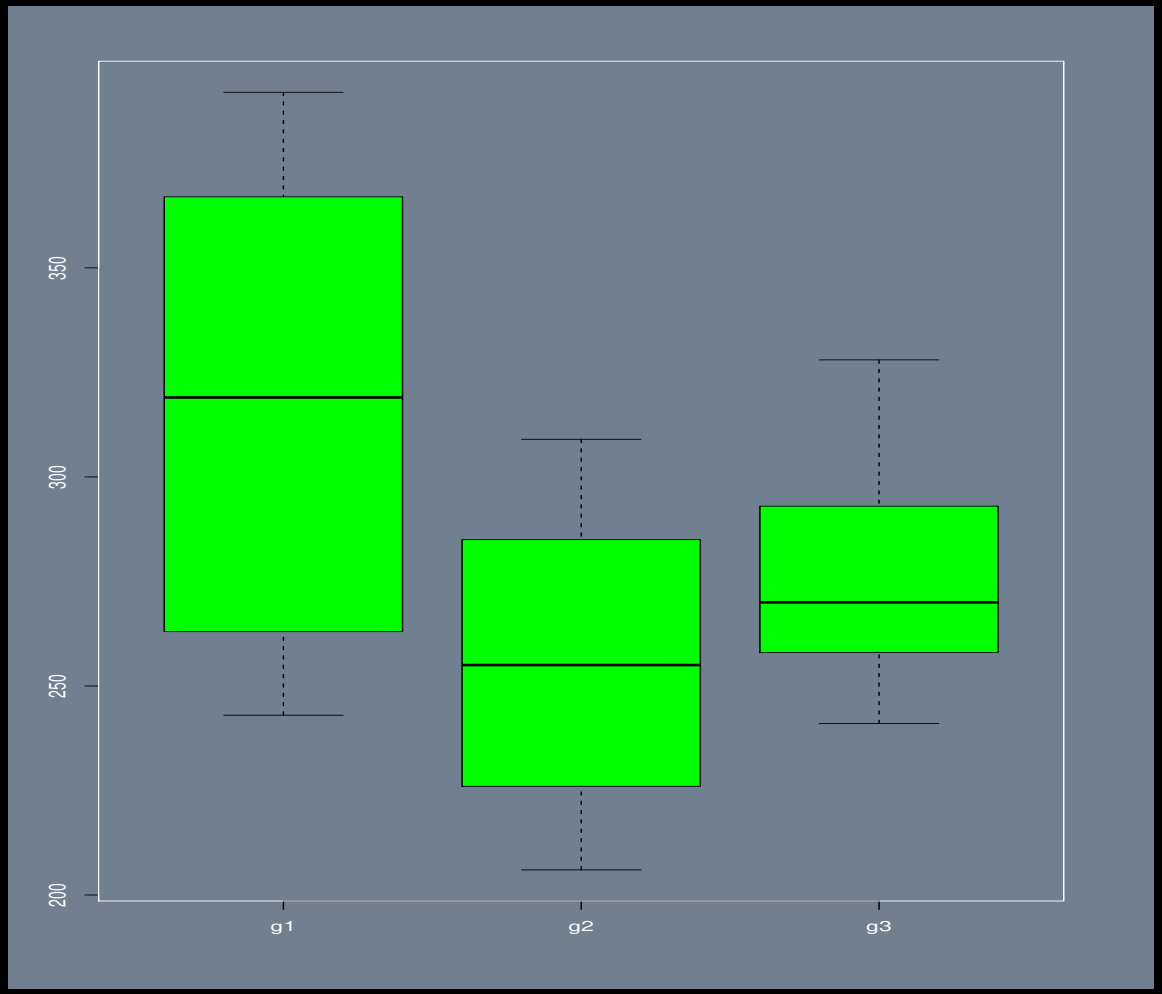
- The F-test is built according to:

Source	DF	SS	MS	MSR
Between Groups	$J-1$	$\sum_{j=1}^{J} n_j(\bar{y}_j - \bar{\bar{y}})^2$	SS/DF (1)	(1)/(2)
Within Groups	$N-J$	$\sum_{j=1}^{J}\sum_{i=1}^{n_j}(y_{ij} - \bar{y}_j)^2$	SS/DF (2)	
Total	$N-1$	$\sum_{j=1}^{J}\sum_{i=1}^{n_j}(y_{ij} - \bar{\bar{y}})^2$		

Cardiac Bypass Surgery, Code

```
redcell <- read.table("http://jgill.wustl.edu/data/redf.dat",header=TRUE)
redcell.aov <- aov(Folate~Group, contrasts = list(Group = contr.treatment),
                   data=redcell);        summary(redcell.aov)

            Df Sum Sq Mean Sq F value  Pr(>F)
Group        2  15516  7757.9  3.7113 0.04359
Residuals   19  39716  2090.3

print(model.tables(redcell.aov,"means"),digits=3)

Tables of means
Grand mean              Group
283.2273              g1  g2  g3
                     317 256 278
              rep     8   9   5

par(bg="slategray",mar=c(4,2,2,2),col.main="white",col.axis="white",col="white")
boxplot(Folate ~ Group, col="green", data=redcell)
```

Cardiac Bypass Surgery, Boxplots



Data Decomposition Example

- A long-term study of patients 65 or older at three locations with two conditions (Wentworth Medical Center in New York study).
- Consider the two-way balanced setup:
  - 20 individuals each in Florida, New York, and North Carolina in "reasonably good health" are given a standardized test for depression where higher scores indicate higher levels of depression,
  - 20 individuals each in Florida, New York, and North Carolina with chronic conditions (arthritis, hypertension, and/or heart ailment) are given the same test.

- Data:

Good Health			Chronic Conditions		
Florida	New York	North Carolina	Florida	New York	North Carolina
3	8	10	13	14	10
7	11	7	12	9	12
7	9	3	17	15	15
3	7	5	17	12	18
8	8	11	20	16	12
8	7	8	21	24	14
8	8	4	16	18	17
5	4	3	14	14	8
5	13	7	13	15	14
2	10	8	17	17	16
6	6	8	12	20	18
2	8	7	9	11	17
6	12	3	12	23	19
6	8	9	15	19	15
9	6	8	16	17	13
7	8	12	15	14	14
5	5	6	13	9	11
4	7	3	10	14	12
7	7	8	11	13	13
3	8	11	17	11	11

ANOVA for the Depression Data

- For the depression study, a two-way ANOVA with levels starts with:

$$
\begin{aligned}
y_i &= \mu + \gamma_{j[i]} + \delta_{k[i]} + \epsilon_i, && \text{for } i = 1, \ldots, n \\
\gamma_j &\sim N(0, \sigma_\gamma^2), && \text{for } j = 1, \ldots, J \quad \text{given three states} \\
\delta_k &\sim N(0, \sigma_\delta^2), && \text{for } k = 1, \ldots, K \quad \text{given two health conditions} \\
\epsilon_i &\sim N(0, \sigma_\epsilon^2), && \text{for } i = 1, \ldots, n.
\end{aligned}
$$

- The degrees of freedom are the number of coefficients minus the number of constraints for each group:

$$
\begin{aligned}
DF_{\text{data}} &= C - 1 = 5, \quad \text{number columns (residuals) minus overall mean} \\
DF_\gamma &= J - 1 = 2, \quad \text{number states minus state mean} \\
DF_\delta &= K - 1 = 1, \quad \text{number health conditions minus health conditions mean.}
\end{aligned}
$$

- So the degrees of freedom for the residuals are:

$$DF_\epsilon = DF_{\text{data}} - DF_\gamma - DF_\delta = 2.$$

Data Decomposition Example, Model

- Assumptions: (1) data are iid, (2) residuals are normally distributed, and (3) variance of the data in a group is the same.
- An additive decomposition model breaks these factors up according to:

$$y_i = \mu + \gamma_{j[i]} + \delta_{k[i]} + \epsilon_i$$

- Since we have one "observation" per cell, the residuals (ϵ_i) are equivalent to interactions between the two factors.
- Sometimes the constraint $\mu = 0$ is imposed and the interest is on variances rather than coefficients (centering), which makes the residual sum of squares equal to one.

Data Decomposition Example, Estimation

- The coefficients are estimated with OLS.
- The variance components are analyzed with sums of squares for each quantity:

$$SS_{\text{state}} = \sum_{i=1}^{6} \hat{\gamma}_{j[i]}^2 \qquad SS_{\text{health}} = \sum_{i=1}^{6} \hat{\delta}_{j[i]}^2 \qquad SS_{\text{residuals}} = \sum_{i=1}^{6} \hat{\epsilon}_{j[i]}^2.$$

- If the data are balanced (as they are here) then the three sums of squares add up to the total sum of squares:

$$SS_{\text{total}} = \sum_{i=1}^{6} (y_i - \hat{\mu})^2.$$

Data Decomposition Example, Estimation

- For each source of variation, we further create the *mean square error* by dividing the sum of squares by its DF.
- If this quantity is close to the MSE of the residuals, then we surmise that there is no important variation for this factor.
- If this quantity is noticeably bigger than MSE of the residuals, then we surmise that the variation is important.
- The formal test uses an F-distriibution, F_{ν_1,ν_2}, with ν_1 equal to the DF of the numerator and ν_2 equal to the DF of the denominator. Tail values show a significant difference.

Data Decomposition Example, Code

```
depression.vec <- scan("http://jgill.wustl.edu/data/depression.data")
depression.mat <- matrix(depression.vec,ncol=6,byrow=TRUE)
t(depression.mat)
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
[1,]    3    7    7    3    8    8    8    5    5     2     6     2     6     6     9
[2,]    8   11    9    7    8    7    8    4   13    10     6     8    12     8     6
[3,]   10    7    3    5   11    8    4    3    7     8     8     7     3     9     8
[4,]   13   12   17   17   20   21   16   14   13    17    12     9    12    15    16
[5,]   14    9   15   12   16   24   18   14   15    17    20    11    23    19    17
[6,]   10   12   15   18   12   14   17    8   14    16    18    17    19    15    13
     [,16] [,17] [,18] [,19] [,20]
[1,]     7     5     4     7     3
[2,]     8     5     7     7     8
[3,]    12     6     3     8    11
[4,]    15    13    10    11    17
[5,]    14     9    14    13    11
[6,]    14    11    12    13    11
```

Data Decomposition Example, Code

```
treatment.state <- rep(1:3,each=2)
treatment.condition <- rep(1:2,each=3)
y <- apply(depression.mat,2,mean)[c(1,4,2,5,3,6)]
( depression.table <- matrix(y,ncol=3,byrow=FALSE) )
      [,1]  [,2]  [,3]
[1,]  5.55  8.00  7.05
[2,] 14.50 15.25 13.95

depression.out <- aov(y ~ factor(treatment.state) + factor(treatment.condition))
matrix(depression.out$fitted.values,ncol=3)
       [,1]   [,2]  [,3]
[1,]  6.175  7.775  6.65
[2,] 13.875 15.475 14.35
```

Data Decomposition Example, Code

```
summary(depression.out)
                           Df Sum Sq Mean Sq F value Pr(>F)
factor(treatment.state)     2    2.7     1.4    2.25 0.3081
factor(treatment.condition) 1   88.9    88.9  147.92 0.0067
Residuals                   2    1.2     0.6

depression.out$coefficients
                (Intercept)     factor(treatment.state)2
                       6.18                        1.60
    factor(treatment.state)3 factor(treatment.condition)2
                       0.47                        7.70
```

ANOVA For Model Comparison

- This is a test between *nested* models that have already been fit.
- The hypothesis is that the smaller model is true, meaning that the coefficients that are estimated in the larger model, but not the smaller model, are actually zero.
- Suppose that the smaller model has k_1 covariates and the larger model has $k_1 + k_2$ covariates (nesting). The linear predictor from the smaller model is expressed as $\mathbf{X}_1\hat{\boldsymbol{\beta}}_1$ and the linear predictor from the larger model is expressed as $\mathbf{X}_2\hat{\boldsymbol{\beta}}_2$.
- The model is summarized by:

	DF	SS	MSE
Model Difference	k_2	$\sum_i (y_i - \mathbf{X}_1\hat{\boldsymbol{\beta}}_1)^2 - \sum_i (y_i - \mathbf{X}_2\hat{\mathbf{X}}_2)^2$	SS/k_2
Residuals	$n - k_1 - k_2$	$\sum_i (y_i - \mathbf{X}_2\hat{\boldsymbol{\beta}}_2)^2$	$SS/(n - k_1 - k_2)$

ANOVA For Model Comparison, Load and Condition Data

```
############## CAESARIAN BIRTH EXAMPLE, FAHRMEIER & TUTZ, PAGE 75 #############

caes.df <- read.table("http://jgill.wustl.edu/data/caesarian.dat",header=TRUE)
caes.df$NOPLAN <- factor(caes.df$NOPLAN)
   levels(caes.df$NOPLAN) <- c("FALSE","TRUE")
caes.df$ANTIB<- factor(caes.df$ANTIB)
   levels(caes.df$ANTIB) <- c("NOT GIVEN","GIVEN")
caes.df$FACTOR<- factor(caes.df$FACTOR)
   levels(caes.df$FACTOR) <- c("NONE","PRESENT")
caes.df$INFECT <- factor(caes.df$INFECT);
   levels(caes.df$INFECT) <- c("None","TYPE I","TYPE II")

attach(caes.df); caes.df
```

ANOVA For Model Comparison, Example Data

	FREQ	INFECT	NOPLAN	ANTIB	FACTOR
1	32	None	FALSE	NOT GIVEN	NONE
2	4	TYPE I	FALSE	NOT GIVEN	NONE
3	4	TYPE II	FALSE	NOT GIVEN	NONE
4	30	None	FALSE	NOT GIVEN	PRESENT
5	11	TYPE I	FALSE	NOT GIVEN	PRESENT
6	17	TYPE II	FALSE	NOT GIVEN	PRESENT
7	2	None	FALSE	GIVEN	NONE
8	0	TYPE I	FALSE	GIVEN	NONE
9	0	TYPE II	FALSE	GIVEN	NONE
10	17	None	FALSE	GIVEN	PRESENT
11	0	TYPE I	FALSE	GIVEN	PRESENT
12	1	TYPE II	FALSE	GIVEN	PRESENT
13	9	None	TRUE	NOT GIVEN	NONE
14	0	TYPE I	TRUE	NOT GIVEN	NONE
15	0	TYPE II	TRUE	NOT GIVEN	NONE
16	3	None	TRUE	NOT GIVEN	PRESENT
17	10	TYPE I	TRUE	NOT GIVEN	PRESENT
18	13	TYPE II	TRUE	NOT GIVEN	PRESENT
19	77	None	TRUE	GIVEN	PRESENT
20	4	TYPE I	TRUE	GIVEN	PRESENT
21	7	TYPE II	TRUE	GIVEN	PRESENT

ANOVA For Model Comparison, Model 1

```
options()$contrasts
        unordered           ordered
"contr.treatment"      "contr.poly"
options(digits =  5)

library(nnet)
caes.out <- multinom(INFECT ~ NOPLAN+ANTIB+FACTOR,data=caes.df,weights=FREQ)
summary(caes.out,correlation=FALSE)

Coefficients:
        (Intercept) NOPLANTRUE ANTIBGIVEN FACTORPRESENT
TYPE I      -2.6255     1.1885    -3.4163         1.8305
TYPE II     -2.5651     1.0134    -2.9861         2.1964

Std. Errors:
        (Intercept) NOPLANTRUE ANTIBGIVEN FACTORPRESENT
TYPE I      0.55699    0.51988    0.67068       0.60254
TYPE II     0.54654    0.47923    0.54825       0.58717

Residual Deviance: 319.31
AIC: 335.31
```

ANOVA For Model Comparison, Model 2 (interactions added)

```
caes2.out <- multinom(INFECT ~ NOPLAN*ANTIB*FACTOR,data=caes.df,weights=FREQ)
anova(caes.out,caes2.out,test="none")

Likelihood ratio tests of Multinomial Models

Response: INFECT
                    Model Resid. df Resid. Dev   Test    Df LR stat.
1 NOPLAN + ANTIB + FACTOR        34     319.31
2 NOPLAN * ANTIB * FACTOR        28     307.52 1 vs 2     6   11.787

pchisq(11.787,df=6,lower.tail=FALSE)
[1] 0.066893
```

One-Way ANOVA: Back to Radon

- We can conceptualize of a random intercept model *with non-zero group means* as an ANOVA.
- Model:

$$y_i = N(\alpha_{j[i]}, \sigma_y^2), \qquad \text{for } i = 1, \ldots, n$$
$$\alpha_j = N(\mu_\alpha, \sigma_\alpha^2), \qquad \text{for } j = 1, \ldots, J.$$

- The following code highlights the difference between the finite population standard deviations and superpopulation standard deviations.

One-Way ANOVA, JAGS Code

```
model {
    for (i in 1:n){
        y[i]     ~ dnorm (y.hat[i], tau.y)
        y.hat[i] <- a[county[i]]
        e.y[i]   <- y[i] - y.hat[i]        # DATA-LEVEL ERRORS
    }

    tau.y   <- pow(sigma.y, -2)
    sigma.y ~ dunif (0, 100)
    s.y     <- sd(e.y[])
    for (j in 1:J){
        a[j] ~ dnorm (mu.a, tau.a)
    }
    mu.a    ~ dnorm (0, .0001)
    tau.a   <- pow(sigma.a, -2)
    sigma.a ~ dunif (0, 100)
    a.y     <- sd(a[])
}
```

One-Way ANOVA, Running JAGS

```
. model in "radon.nopreds.anova.bug"
. data in "radon.no.preds.jags.dat"

 Reading data file radon.no.preds.jags.dat

. compile

Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 2668

. inits in "radon.jags.init"
. initialize

Reading initial values file radon.jags.init
```

One-Way ANOVA, Running JAGS

```
. update 5000

Updating 5000
----------------------------------------| 5000
**************************************** 100%

. monitor set e.y
. monitor set sigma.y
. monitor set s.y
. monitor set a
. monitor set mu.a
. monitor set sigma.a
. update 10000

Updating 10000
----------------------------------------| 10000
**************************************** 100%
. coda *
```

One-Way ANOVA: Running CODA

```
> library(coda)
> codamenu()
CODA startup menu

1: Read BUGS output files
2: Use an mcmc object
3: Quit

Selection: 1
Enter CODA index file name
(or a blank line to exit)
1: /Users/jgill/Class.Multilevel/examples/radon/CODAindex.txt
```

One-Way ANOVA: Running CODA

```
Enter CODA output file names, separated by return key
(leave a blank line when you have finished)
1: /Users/jgill/Class.Multilevel/examples/radon/CODAchain1.txt
2:
Abstracting e.y[1] ... 10000 valid values
Abstracting e.y[2] ... 10000 valid values
Abstracting e.y[3] ... 10000 valid values
:
:
Abstracting a[83] ... 10000 valid values
Abstracting a[84] ... 10000 valid values
Abstracting a[85] ... 10000 valid values
Abstracting mu.a ... 10000 valid values
Abstracting sigma.a ... 10000 valid values
Abstracting s.y ... 10000 valid values
Checking effective sample size ...OK
```

One-Way ANOVA: Running CODA

```
CODA Main Menu

1: Output Analysis
2: Diagnostics
3: List/Change Options
4: Quit

Selection: 1
```

One-Way ANOVA: Running CODA

```
CODA Main Menu

1: Output Analysis
2: Diagnostics
3: List/Change Options
4: Quit

Selection: 2
CODA Diagnostics Menu

1: Geweke
2: Gelman and Rubin
3: Raftery and Lewis
4: Heidelberger and Welch
5: Autocorrelations
6: Cross-Correlations
7: List/Change Options
8: Return to Main Menu
```

One-Way ANOVA: Running CODA

```
Selection: 1

GEWEKE CONVERGENCE DIAGNOSTIC (Z-score)
=======================================

Iterations used = 0: 9999
Thinning interval = 1
Sample size per chain = 10000

$`/Users/jgill/Class.Multilevel/examples/radon/CODAchain1.txt`

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5
```

e.y[913]	e.y[914]	e.y[915]	e.y[916]	e.y[917]	e.y[918]	e.y[919]	sigma.y	s.y	a[1]	a[2]	a[3]
2.2497	2.2497	2.2497	2.2497	2.2497	0.0995	0.0995	-2.1408	-0.0644	-0.1587	2.2533	-1.6188
a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a[11]	a[12]	a[13]	a[14]	a[15]
-1.5540	0.5727	0.5349	-0.3201	0.3200	-2.4397	0.0941	1.3714	-0.0658	0.3195	-0.9510	1.3691

One-Way ANOVA: Running CODA

```
CODA Output Analysis menu

1: Plots
2: Statistics
3: List/Change Options
4: Return to Main Menu

Selection: 2
/Users/jgill/Class.Multilevel/examples/radon/CODAchain1.txt

Iterations = 0: 9999
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000
```

One-Way ANOVA: Running CODA

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

	Mean	SD	Naive SE	Time-series SE
e.y[1]	-0.273000	0.25365	2.54e-03	0.003318
e.y[2]	-0.273000	0.25365	2.54e-03	0.003318
e.y[3]	0.003253	0.25365	2.54e-03	0.003318
:				
a[83]	1.058199	1.289322	1.412560	1.53447	1.763863
a[84]	1.141359	1.374208	1.495045	1.61658	1.848449
a[85]	0.740655	1.098900	1.282850	1.47047	1.825231
mu.a	1.213629	1.279490	1.311740	1.34496	1.412434
sigma.a	0.226575	0.284225	0.315733	0.35021	0.418392
s.y	0.230221	0.283173	0.310787	0.33851	0.391804

Multilevel Linear Model R^2

- Classic linear model R^2:

$$R^2 = 1 - \frac{\overset{n}{\underset{i=1}{V}} \epsilon_i}{\overset{n}{\underset{i=1}{V}} y_i}$$

where:

$$\overset{n}{\underset{i=1}{V}} x_i = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2.$$

- But this does not account for what a "null model" would be since different levels could be reduced.
- G&H suggest using an R^2 at each level.
- For each level, $m = 1, \ldots, M$, for each group, $k = 1, \ldots, K^{(m)}$, consider:

$$\theta_k^{(m)} = \hat{\theta}_k^{(m)} + \epsilon_k^{(m)}.$$

Multilevel Linear Model R^2

- For each level, $m = 1, \ldots, M$, we can calculate its own value:

$$R^2_m = 1 - \frac{\overset{K}{\underset{i=1}{V}} \epsilon_i}{\overset{K}{\underset{i=1}{V}} \theta_i}$$

- These are calculated from MCMC output.
- If the *average residual error variance* is close to the *average variance of the systematic component*, then this value is small.
- If the *residual errors* is close to zero then this value is large.
- This value is large for the hierarchy in question when $\hat{\theta}_k^{(m)}$values are close to $\theta_k^{(m)}$.

One-Way ANOVA, JAGS Code, Additions To Get R^2, With Basement and Uranium

```
model {
    for (i in 1:n){
        y[i]     ~ dnorm (y.hat[i], tau.y)
        y.hat[i] <- a[county[i]] + b[county[i]]*x[i]
        e.y[i]   <- y[i] - y.hat[i]       # DATA-LEVEL ERRORS
    }
    tau.y   <- pow(sigma.y, -2)
    sigma.y ~ dunif (0, 100)
    for (j in 1:J){
        a[j] <- B[j,1]                          # PUT BOTH GROUP-LEVEL...
        b[j] <- B[j,2]                        # ... COEFFICIENTS IN A VECTOR
        B[j,1:2] ~ dnorm(B.hat[j,], Tau.B[,]) # BIVARIATE NORMAL DRAW
        B.hat[j,1] <- g.a.0 + g.a.1*u[j]      # PUT URANIUM IN BOTH HIER. PARAMS
        B.hat[j,2] <- g.b.0 + g.b.1*u[j]
        for (k in 1:2)  {
            E.B[j,k] <- B[j,k] - B.hat[j,k]   # GROUP-LEVEL ERRORS
        }
    }
}
```

One-Way ANOVA, R Code For R^2 At Levels

- These steps can be added to the JAGS code to get distributional summaries of R-Square, for a single value summary, the posterior mean is appropriate.
- Get the full model R-Square from the finite-population SEs:

```
rsquared.y <- 1 - mean(apply(e.y,1,var))/var(y)
```

- Pull out the two group-level error vectors:

```
e.a <- E.B[,1]
e.b <- E.B[,2]
```

- Now do the same R-Square for each level:

```
rsquared.a <- 1 - mean(apply(e.a,1,var))/mean(apply(a,1,var))
rsquared.b <- 1 - mean(apply(e.b,1,var))/mean(apply(b,1,var))
```

Two-Way ANOVA: Flight Simulator Data

- Model:

$$
\begin{aligned}
y_i &= \mu + \gamma_{j[i]} + \delta_{k[i]} + \epsilon_i && \text{for } i = 1, \ldots, n \\
\gamma_j &= N(0, \sigma^2_\gamma), && \text{for } j = 1, \ldots, J. \\
\delta_k &= N(0, \sigma^2_\delta), && \text{for } k = 1, \ldots, K. \\
\epsilon_i &= N(0, \sigma^2_\epsilon), && \text{for } i = 1, \ldots, n.
\end{aligned}
$$

- Degrees of freedom:

$$
\begin{aligned}
\text{Model}: \; & n - 1 = 39 \\
\gamma: \; & J - 1 = 4 \\
\delta: \; & K - 1 = 7 \\
\epsilon: \; & 39 - 4 - 7 = 28
\end{aligned}
$$

Two-Way ANOVA: Flight Simulator Data

```
model {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- mu + b.treatment[treatment[i]] + b.airport[airport[i]]
    e.y[i] <- y[i] - y.hat[i]
  }
  mu ~ dnorm (0, .0001)
  g.mu <- mu + mean(b.treatment[]) + mean(b.airport[])
  gg.mu <- mu + mu.treatment + mu.airport
  tau.y <- pow(sigma.y,-2)
  sigma.y ~ dunif (0, 100)
  s.error <- sd(e.y[])
  # LOOP THROUGH THE TREATMENTS
  for (j in 1:n.treatment){
    b.treatment[j] ~ dnorm (mu.treatment, tau.treatment)
    g.treatment[j] <- b.treatment[j] - mean(b.treatment[])
    gg.treatment[j] <- b.treatment[j] - mu.treatment
  }
```

Two-Way ANOVA: Flight Simulator Data

```
  mu.treatment ~ dnorm (0, .0001)
  tau.treatment <- pow(sigma.treatment,-2)
  sigma.treatment ~ dunif (0, 100)
  s.treatment <- sd(b.treatment[])
  # LOOP THROUGH THE AIRPORTS
  for (j in 1:n.airport){
    b.airport[j] ~ dnorm (mu.airport, tau.airport)
    g.airport[j] <- b.airport[j] - mean(b.airport[])
    gg.airport[j] <- b.airport[j] - mu.airport
  }
  mu.airport ~ dnorm (0, .0001)
  tau.airport <- pow(sigma.airport,-2)
  sigma.airport ~ dunif (0, 100)
  s.airport <- sd(b.airport[])
}
```

Two-Way ANOVA with Replication

- When replications are made in the experiments, we can estimate separately the two-way interactions and the measurement error.
- This expands the two-way ANOVA model to:

$$y_i = \mu + \gamma_{j[i]} + \delta_{k[i]} + +\eta_{j[i],k[i]} + \epsilon_i$$

  in G&H notation.
- Thus the degrees of freedom are the same except that for m the error DF becomes $n - m \times r \times c$, where r is the number of rows and c is the number of columns. The means consume extra DF since they are an estimate.

Adding Predictors: Analysis of Covariance and Contrast Analysis

- *Latin Square designs* (Fisher) are good approaches when we cannot use each treatment level for the same combination of blocking levels.
- Consider an experiment with four diets (different types of political advertising), each to be given to four patients (prospective voters) in succession.
- If each patient was given the diets in the same order, the treatment effect would be confounded with the effect due to the order in which the diets were given.
- But each patient can only be given a single diet during a single time period.
- This is an example of an *incomplete block design* where there is a single treatment and two blocking variables (diet and time), each with the same number of levels.
- Just a single treatment is applied within each combination of blocking variables.

Adding Predictors: Analysis of Covariance and Contrast Analysis

- Continuing the example, we have four diet treatments labelled A, B, C, and D.
- There are four patients which we will assign to rows $1:4$.
- There are four time periods which we will assign to columns $1:4$.
- So each patient gets each treatment exactly once in a random order.
- Note that Latin Squares are *not unique*.
- Latin squares are a method of treatment randomization.

Adding Predictors: Analysis of Covariance and Contrast Analysis

```
latin.sq <- function (n, rand = 25)  {
    X = matrix(LETTERS[1:n], n, n)
    X = t(X)                                        # NOW CONSTANT DOWN COLUMNS
    for (i in 2:n) X[i, ] = X[i, c(i:n, 1:(i - 1))]     # CREATES PERMUTATIONS
    if (rand > 0) {                             # SHUFFLES PERMUTATIONS rand TIMES
        for (i in 1:rand) {
            X = X[sample(n),]      # sample DEFAULT IS UNIFORM WITHOUT REPLACE
            X = X[,sample(n)]
        }
    }
    dimnames(X) <- list(paste("P",1:n,sep=""),paste("D",1:n,sep=""))
    return(X)
}
latin.sq(4)
   D1  D2  D3  D4
P1 "A" "D" "C" "B"
P2 "C" "B" "A" "D"
P3 "D" "C" "B" "A"
P4 "B" "A" "D" "C"
```

Adding Predictors: Analysis of Covariance and Contrast Analysis

```
library(SMPracticals)
data(millet)

str(millet)        # DISPLAYS STRUCTURE OF AN OBJECT
'data.frame':        25 obs. of  4 variables:
 $ row : Factor w/ 5 levels "1","2","3","4",..: 1 1 1 1 1 2 2 2 2 2 ...
 $ col : Factor w/ 5 levels "1","2","3","4",..: 1 2 3 4 5 1 2 3 4 5 ...
 $ dist: Factor w/ 5 levels "2","4","6","8",..: 4 1 2 5 3 3 2 1 4 5 ...
 $ y   : num  277 230 279 307 262 305 283 245 300 280 ...
names(millet) <- c("row", "column", "treat", "yield")
```

Adding Predictors: Analysis of Covariance and Contrast Analysis

```
millet
```

row	col	dist	y	row	col	dist	y	row	col	dist	y
1	1	8	277	2	4	8	300	4	2	10	224
1	2	2	230	2	5	10	280	4	3	6	287
1	3	4	279	3	1	2	182	4	4	2	193
1	4	10	307	3	2	8	272	4	5	8	279
1	5	6	262	3	3	10	306	5	1	10	251
2	1	6	305	3	4	6	306	5	2	6	331
2	2	4	283	3	5	4	250	5	3	8	286
2	3	2	245	4	1	4	203	5	4	4	334
								5	5	2	338

Adding Predictors: Analysis of Covariance and Contrast Analysis

- Adding covariates at the group level to an ANOVA model is equivalent to having contrasts.
- Multilevel ANOVA with no contrasts to estimate three effects:

$$
\begin{aligned}
y_i &\sim N(\mu + \beta_{j[i]}^{\text{row}} + \beta_{j[i]}^{\text{column}} + \beta_{j[i]}^{\text{treat}}, \sigma_y^2), && \text{for } i = 1, \ldots, 5 \\
\beta_j^{\text{row}} &\sim N(0, \sigma_{\beta,\text{row}}), && \text{for } j = 1, \ldots, 5 \\
\beta_k^{\text{column}} &\sim N(0, \sigma_{\beta,\text{column}}), && \text{for } k = 1, \ldots, 5 \\
\beta_\ell^{\text{treat}} &\sim N(0, \sigma_{\beta,\text{treat}}), && \text{for } \ell = 1, \ldots, 5
\end{aligned}
$$

- Estimate with:

```
millet1.out <- lm(yield ~ row + column + treat, data=millet)
anova(millet1.out)

Response: yield
          Df Sum Sq Mean Sq F value Pr(>F)
row        4  13485    3371    3.10  0.057
column     4   6239    1560    1.43  0.282
treat      4   9965    2491    2.29  0.120
Residuals 12  13051    1088
```

Adding Predictors: Analysis of Covariance and Contrast Analysis

```
summary(millet1.out)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    207.4       23.8    8.72  1.5e-06
row2            11.6       20.9    0.56    0.588
row3            -7.8       20.9   -0.37    0.715
row4           -33.8       20.9   -1.62    0.131
row5            37.0       20.9    1.77    0.101
column2         24.4       20.9    1.17    0.265
column3         37.0       20.9    1.77    0.101
column4         44.4       20.9    2.13    0.055
column5         38.2       20.9    1.83    0.092
treat4          32.2       20.9    1.54    0.149
treat6          60.6       20.9    2.91    0.013
treat8          45.2       20.9    2.17    0.051
treat10         36.0       20.9    1.73    0.110
Residual standard error: 33 on 12 degrees of freedom
Multiple R-squared: 0.695,        Adjusted R-squared: 0.389
F-statistic: 2.27 on 12 and 12 DF,  p-value: 0.0844
```

Adding Predictors: Analysis of Covariance and Contrast Analysis

- Now add linear contrasts for the rows, columns, and treatments.

```
millet$row <- as.integer(millet$row)
millet$column <- as.integer(millet$column)
millet$treat <- as.integer(millet$treat)
millet$yield <- scale(millet$yield,scale=FALSE)

str(millet)
'data.frame':        25 obs. of  4 variables:
 $ row   : int  1 1 1 1 1 2 2 2 2 2 ...
 $ column: int  1 2 3 4 5 1 2 3 4 5 ...
 $ treat : int  4 1 2 5 3 3 2 1 4 5 ...
 $ yield : num [1:25, 1] 4.6 -42.4 6.6 34.6 -10.4 ...
  ..- attr(*, "scaled:center")= num 272
```

Adding Predictors: Analysis of Covariance and Contrast Analysis

```
millet2.out <- lm(yield ~ row + column + treat, data=millet)
anova(millet2.out)
Response: yield
          Df Sum Sq Mean Sq F value Pr(>F)
row        1    409     409    0.25   0.62
column     1   4646    4646    2.86   0.11
treat      1   3613    3613    2.23   0.15
Residuals 21  34072    1622
```

Adding Predictors: Analysis of Covariance and Contrast Analysis

```
summary(millet2.out)
Residuals:
   Min     1Q Median     3Q    Max
-74.90 -24.84   0.74  21.10  62.52

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -63.00      30.68   -2.05    0.053
row             2.86       5.70    0.50    0.621
column          9.64       5.70    1.69    0.105
treat           8.50       5.70    1.49    0.151

Residual standard error: 40.3 on 21 degrees of freedom
Multiple R-squared: 0.203,        Adjusted R-squared: 0.0889
F-statistic: 1.78 on 3 and 21 DF,  p-value: 0.182
```