HW12 ANOVA decomposition

Jared Andreatta

2025-04-28

The PVC dataset

Loading the small example data set **pvc** from the textbook *Linear Models with R*, Julian Faraway.

```
load("pvc.rda")
head( pvc)
```

```
##   psize resin operator
## 1  36.2     1        1
## 2  36.3     1        1
## 3  35.3     2        1
## 4  35.0     2        1
## 5  30.8     3        1
## 6  30.6     3        1
```

```
# each combination of resin/operator has two replications
table( pvc$resin, pvc$operator)
```

```
##
##     1 2 3
##   1 2 2 2
##   2 2 2 2
##   3 2 2 2
##   4 2 2 2
##   5 2 2 2
##   6 2 2 2
##   7 2 2 2
##   8 2 2 2
```

About the dataset (from Faraway's R package help file)

Production of PVC by operator and resin railcar

Description

Data from an experiment to study factors affecting the production of the plastic PVC, 3 operators used 8 different devices called resin railcars to produce PVC. For each of the 24 combinations, two samples were produced.

Dataset contains the following variables

- psize Particle size
- operator Operator number 1, 2 or 3
- resin Resin railcar 1-8

Source

R. Morris and E. Watson (1998) "A comparison of the techniques used to evaluate the measurement process" *Quality Engineering*, 11, 213-219

For reference the complete (and default) 2-way analysis. Note the use of the * in the formula equation.

```
fullFit<- lm(psize ~ operator*resin, pvc )

ANOVATable<- anova( fullFit)
ANOVATable
```

```
## Analysis of Variance Table
##
## Response: psize
##                Df  Sum Sq Mean Sq F value    Pr(>F)
## operator        2  20.718  10.359  7.0072   0.00401 **
## resin           7 283.946  40.564 27.4388 5.661e-10 ***
## operator:resin 14  14.335   1.024  0.6926   0.75987
## Residuals      24  35.480   1.478
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# the complete sums of squares subtracting off the grand mean is
SSTotal<- sum( (pvc$psize - mean(pvc$psize))^2)
SSTotal
```

```
## [1] 354.4792
```

```
# compare to
sum( ANOVATable[,2])
```

```
## [1] 354.4792
```

Problem 1

- The fields function **stats** is a handy function that will find summary statistics of the columns of a matrix or a data frame E.g. `stats(pvc)` in this case the summaries for operator and resin are listed as NA. Why?

These aren't numerical vectors. Instead, they are factors, or categorical variables.

- For these data compute the 1-way analysis "by hand" (use R basic arithmetic and functions but not **lm**) for the **resin** factor and the response **psize**. Specifically under the model:

$$Y_{ik} = \mu + \alpha_i + e_{ik}$$

  Determine the OLS estimates for the grand mean, μ, and main effects, $\{\alpha_i\}$ $i = 1, ..., 8$, where the main effects are constrained to sum to 0.

```
# Total mean
mu_hat <- mean(pvc$psize)
# resin means
ybar_resin <- tapply(pvc$psize, pvc$resin, mean)
# Effects
alpha_hat <- ybar_resin - mu_hat

# Approximately 0
sum(alpha_hat)
```

```
## [1] 2.4869e-14
```

```
data.frame(
  resin = names(alpha_hat),
  alpha = alpha_hat
)
```

```
##   resin      alpha
## 1     1  3.2958333
## 2     2  2.2625000
## 3     3 -2.5041667
## 4     4 -2.8875000
## 5     5 -1.5041667
## 6     6 -2.1541667
## 7     7  0.3791667
## 8     8  3.1125000
```

- Fit this model using **lm**. Do you get the same results? Which parts are the same and which are different?

The fitted values, coefficients, and residuals are identical. The difference to note is that resin8 is not accounted for in these coefficients.

```
# sum(alpha)=0 contrast
options(contrasts = c("contr.sum", "contr.poly"))
fit <- lm(psize ~ resin, data = pvc)
summary(fit)$coef
```

```
##               Estimate Std. Error     t value     Pr(>|t|)
## (Intercept) 32.3541667  0.1916667 168.8043478 1.076853e-58
## resin1       3.2958333  0.5071023   6.4993456 9.395846e-08
## resin2       2.2625000  0.5071023   4.4616241 6.462478e-05
## resin3      -2.5041667  0.5071023  -4.9381880 1.441509e-05
## resin4      -2.8875000  0.5071023  -5.6941170 1.271778e-06
## resin5      -1.5041667  0.5071023  -2.9661995 5.066524e-03
## resin6      -2.1541667  0.5071023  -4.2479920 1.250717e-04
## resin7       0.3791667  0.5071023   0.7477123 4.590072e-01
```

```
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: psize
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## resin      7 283.946  40.564  23.004 3.712e-12 ***
## Residuals 40  70.533   1.763
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Without using **lm** calculate the F statistic for the null hypothesis that all the $\{\alpha_i\}$ are zero. Check your answer against the results of using **lm** and the **anova** function.

```
# mu
ybar_all <- mean(pvc$psize)

# resin sample and means
n_i <- as.numeric(table(pvc$resin))
ybar_i <- tapply(pvc$psize, pvc$resin, mean)
```

```r
# Sum of squares between groups
SSB <- sum( n_i * (ybar_i - ybar_all)^2 )
# Sum of squared within group
SSW <- sum( tapply(pvc$psize, pvc$resin, function(x) sum((x - mean(x))^2)) )

# Df's
dfB <- length(ybar_i) - 1
dfW <- length(pvc$psize) - 8

# Mean squared between/within groups
MSB <- SSB / dfB
MSW <- SSW / dfW

# F-stat
F <- MSB / MSW

# Statistics
c(SSB=SSB, SSW=SSW, MSB=MSB, MSW=MSW, F=F)
```

```
##        SSB        SSW        MSB        MSW          F
## 283.945833  70.533333  40.563690   1.763333  23.003983
```

```r
# Sanity check
fit <- lm(psize ~ resin, data = pvc)
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: psize
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## resin      7 283.946  40.564  23.004 3.712e-12 ***
## Residuals 40  70.533   1.763
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Problem 2

- Now consider both factors, **resin** and **operator** and estimate "by hand" the parameters in the model

$$Y_{ijk} = \mu + \alpha_i + \beta_j + e_{ijk}$$

  Here i indexes the 8 resin levels and j indexes the 3 operator levels. α_i are the 8 main effects for resin and β_j are the 3 main effects for operator. Again assume that both estimates of the main effects are constrained to sum to 0.

```r
# Grand mean
mu_hat <- mean(pvc$psize)

# resin effects (alpha)
ybar_resin <- tapply(pvc$psize, pvc$resin,    mean)
alpha_hat <- ybar_resin - mu_hat
stopifnot(abs(sum(alpha_hat))  < 1e-12) # sum to 0 constraint

# operator effects (beta)
ybar_operator <- tapply(pvc$psize, pvc$operator, mean)
beta_hat <- ybar_operator  - mu_hat
```

```r
stopifnot(abs(sum(beta_hat)) < 1e-12)                # sum to 0 constraint

mu_df <- data.frame(parameter = "mu", level = NA, value = mu_hat)
alpha_df <- data.frame(parameter = "alpha", level = names(alpha_hat), value = alpha_hat)
beta_df <- data.frame(parameter = "beta", level = names(beta_hat),  value = beta_hat)

mu_df
```

```
##   parameter level    value
## 1        mu    NA 32.35417
```

```r
alpha_df
```

```
##   parameter level      value
## 1     alpha     1  3.2958333
## 2     alpha     2  2.2625000
## 3     alpha     3 -2.5041667
## 4     alpha     4 -2.8875000
## 5     alpha     5 -1.5041667
## 6     alpha     6 -2.1541667
## 7     alpha     7  0.3791667
## 8     alpha     8  3.1125000
```

```r
beta_df
```

```
##   parameter level      value
## 1      beta     1  0.5895833
## 2      beta     2  0.3270833
## 3      beta     3 -0.9166667
```

- Find the F statistics for testing both for the resin main effects being zero and separately for the operator effects being zero. Compare your computation to the full ANOVA given above in the introduction – why are they slightly different?

```r
y <- pvc$psize
N <- length(y)

# Grand mean
mu <- mean(y)

# resin
n_i <- table(pvc$resin)
mean_i <- tapply(y, pvc$resin, mean)
SS_resin <- sum( n_i * (mean_i - mu)^2 )
df_resin <- length(mean_i) - 1
MS_resin <- SS_resin / df_resin

# operators
n_j <- table(pvc$operator)
mean_j <- tapply(y, pvc$operator, mean)
SS_oper <- sum( n_j * (mean_j - mu)^2 )
df_oper <- length(mean_j) - 1
MS_oper <- SS_oper / df_oper

# residuals
SST <- sum( (y - mu)^2 )
```

```r
SS_err <- SST - SS_resin - SS_oper
df_err <- N - 1 - df_resin - df_oper
MS_err <- SS_err / df_err

# F-stats
F_resin <- MS_resin / MS_err
F_oper <- MS_oper / MS_err

# Table of stats
cbind(Df = c(df_resin, df_oper, df_err),
      "Sum sq" = c(SS_resin, SS_oper, SS_err),
      "Mean sq" = c(MS_resin, MS_oper, MS_err),
      "F value"  = c(F_resin,  F_oper,  NA))
```

```
##      Df    Sum sq   Mean sq  F value
## [1,]  7 283.94583 40.563690 30.94263
## [2,]  2  20.71792 10.358958  7.90198
## [3,] 38  49.81542  1.310932       NA
```

```r
# Sanity check
add_fit <- lm(psize ~ resin + operator, data = pvc,
              contrasts = list(resin = contr.sum, operator = contr.sum))
anova(add_fit)
```

```
## Analysis of Variance Table
##
## Response: psize
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## resin      7 283.946  40.564  30.943 8.111e-14 ***
## operator   2  20.718  10.359   7.902   0.00135 **
## Residuals 38  49.815   1.311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- An engineer knows that the resin railcars have an effect but then asks: 'What is the relative impact of different operators on **psize** compared to the resin effect?' What would you tell him?

According to the ANOVA table, it is clear that operators have statistically significant effects on psize. However, it has considerably less impact than resin railcars. Resin railcars account for around 84% of the variability, while the operator choice accounts for about 6% of the variability.

Problem 3

Now consider the full model.

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + e_{ijk}$$

where $\{\gamma_{ij}\}$ are the 24 interaction terms.

- List the number of constraints on the main effects and interactions needed to give an X matrix that has full rank

For the full model, we have a total count of 36 parameters: 1 for μ, 8 for α, 3 for β, and 24 for γ. For an ANOVA with 2 factors, we would only have 24 parameters. Thus, we must impose 12 constraints so that X has full rank.

- Examine the **lm** fit of this full model given above. Which interactions does the software choose to leave out to create a full rank X matrix.

As we can see, there are 24 terms accounted for in the lm() fit. The interactions between operator1 and resin1:8 are all dropped, since they are the main-effect dummies created for each resin. Additionally, the operator2 and operator3 interaction terms with resin1 are dropped as well, since they are redundant with the main-effect dummies for the operator variable. Hence, we have 10 omitted interaction terms, so when we combine these with the two constraints where $\sum \alpha = 0$ and $\sum \beta = 0$, that gives us 12 constraints, which we know will give an X with full rank.

```
options(contrasts = c("contr.treatment", "contr.poly"))
fullFit <- lm(psize ~ operator * resin, data = pvc)

coef(fullFit)
```

```
##      (Intercept)        operator2        operator3           resin2
##            36.25            -0.85            -0.95            -1.10
##           resin3           resin4           resin5           resin6
##            -5.55            -6.55            -4.40            -6.05
##           resin7           resin8 operator2:resin2 operator3:resin2
##            -3.35             0.55             1.05            -0.85
## operator2:resin3 operator3:resin3 operator2:resin4 operator3:resin4
##            -0.20            -0.55             1.20            -0.10
## operator2:resin5 operator3:resin5 operator2:resin6 operator3:resin6
##             0.40            -1.60             1.30             0.50
## operator2:resin7 operator3:resin7 operator2:resin8 operator3:resin8
##             0.45             0.85             0.50            -2.70
```

- Following the definition of an interaction, and with this balanced design for the data, we have the estimates:

$$\hat{\gamma_{ij}} = \overline{Y}_{ij} - \hat{\mu} - \hat{\alpha}_i - \hat{\beta}_j$$

where $\overline{Y}_{ij}$ is the group mean for resin level i and operator level j.

Create a table of 48 rows and with 6 columns: the data, the grand mean, the resin main effects,the operator main effects, the interaction effects, and the residuals. Call this `tablePVC` and list out the table. (use `\newpage` to create a separate page for this in your pdf.)

If done correctly, columns 2 through 6 should sum to the first. E.g. `rowSums(tablePVC[,2:6])` is equal to `tablePVC[,1]`. How are the sum of squares of *columns* 2 through 6 `colSums(tablePVC[,2:6])` related to the two way ANOVA table reported by `anova( fullFit)`

Here is an example of indexing that might help you to fill out the table with the estimates. Note you will need to find **alphaHat** and **betaHat** and **muijHat**, the 8×3 matrix of group means to make this code work.

```
indexResin<-as.numeric( pvc$resin)
indexOperator<-as.numeric( pvc$operator)
vecMeans<- muijHat[ cbind(indexResin, indexOperator)]
vecGrand<- rep( mean( pvc$psize), 48)
vecAlpha<- alphaHat[indexResin]
vecBeta<-  betaHat[indexOperator]
vecGamma<- vecMeans - vecAlpha - vecBeta - vecGrand
residual<- pvc$psize - vecMeans
```

```
y <- pvc$psize
N <- length(y)

# Grand mean
```

```r
mu_hat <- mean(y)

# Main effect estimates for alpha and beta
alpha_hat <- tapply(y, pvc$resin,    mean) - mu_hat
beta_hat  <- tapply(y, pvc$operator, mean) - mu_hat

# Group means
mu_ij_hat <- with(pvc, tapply(psize, list(resin, operator), mean))

# Code from above
idxRes <- as.integer(pvc$resin)
idxOper <- as.integer(pvc$operator)
vecGrand <- rep(mu_hat, N)
vecAlpha <- alpha_hat[idxRes]
vecBeta <- beta_hat[idxOper]
vecMean <- mu_ij_hat[cbind(idxRes, idxOper)]
vecGamma <- vecMean - vecGrand - vecAlpha - vecBeta     # interaction piece
vecResid <- y - vecMean

## PVC Table
tablePVC <- cbind(
  psize     = y,
  grandMean = vecGrand,
  alphaHat  = vecAlpha,
  betaHat   = vecBeta,
  gammaHat  = vecGamma,
  resid     = vecResid
)

# Sanity check (it is TRUE)
all.equal(rowSums(tablePVC[, 2:6]), tablePVC[, 1])
```

```
## [1] TRUE
```

```r
# Printed out table
print(tablePVC, digits = 4)
```

```
##   psize grandMean alphaHat betaHat gammaHat resid
## 1  36.2     32.35   3.2958  0.5896  0.01042 -0.05
## 1  36.3     32.35   3.2958  0.5896  0.01042  0.05
## 2  35.3     32.35   2.2625  0.5896 -0.05625  0.15
## 2  35.0     32.35   2.2625  0.5896 -0.05625 -0.15
## 3  30.8     32.35  -2.5042  0.5896  0.26042  0.10
## 3  30.6     32.35  -2.5042  0.5896  0.26042 -0.10
## 4  29.8     32.35  -2.8875  0.5896 -0.35625  0.10
## 4  29.6     32.35  -2.8875  0.5896 -0.35625 -0.10
## 5  32.0     32.35  -1.5042  0.5896  0.41042  0.15
## 5  31.7     32.35  -1.5042  0.5896  0.41042 -0.15
## 6  30.7     32.35  -2.1542  0.5896 -0.58958  0.50
## 6  29.7     32.35  -2.1542  0.5896 -0.58958 -0.50
## 7  33.4     32.35   0.3792  0.5896 -0.42292  0.50
## 7  32.4     32.35   0.3792  0.5896 -0.42292 -0.50
## 8  37.1     32.35   3.1125  0.5896  0.74375  0.30
## 8  36.5     32.35   3.1125  0.5896  0.74375 -0.30
```

```
## 1  35.8     32.35   3.2958  0.3271 -0.57708  0.40
## 1  35.0     32.35   3.2958  0.3271 -0.57708 -0.40
## 2  35.6     32.35   2.2625  0.3271  0.40625  0.25
## 2  35.1     32.35   2.2625  0.3271  0.40625 -0.25
## 3  30.4     32.35  -2.5042  0.3271 -0.52708  0.75
## 3  28.9     32.35  -2.5042  0.3271 -0.52708 -0.75
## 4  30.2     32.35  -2.8875  0.3271  0.25625  0.15
## 4  29.9     32.35  -2.8875  0.3271  0.25625 -0.15
## 5  31.1     32.35  -1.5042  0.3271  0.22292 -0.30
## 5  31.7     32.35  -1.5042  0.3271  0.22292  0.30
## 6  30.9     32.35  -2.1542  0.3271  0.12292  0.25
## 6  30.4     32.35  -2.1542  0.3271  0.12292 -0.25
## 7  32.9     32.35   0.3792  0.3271 -0.56042  0.40
## 7  32.1     32.35   0.3792  0.3271 -0.56042 -0.40
## 8  36.7     32.35   3.1125  0.3271  0.65625  0.25
## 8  36.2     32.35   3.1125  0.3271  0.65625 -0.25
## 1  36.0     32.35   3.2958 -0.9167  0.56667  0.70
## 1  34.6     32.35   3.2958 -0.9167  0.56667 -0.70
## 2  33.0     32.35   2.2625 -0.9167 -0.35000 -0.35
## 2  33.7     32.35   2.2625 -0.9167 -0.35000  0.35
## 3  31.3     32.35  -2.5042 -0.9167  0.26667  2.10
## 3  27.1     32.35  -2.5042 -0.9167  0.26667 -2.10
## 4  30.0     32.35  -2.8875 -0.9167  0.10000  1.35
## 4  27.3     32.35  -2.8875 -0.9167  0.10000 -1.35
## 5  28.7     32.35  -1.5042 -0.9167 -0.63333 -0.60
## 5  29.9     32.35  -1.5042 -0.9167 -0.63333  0.60
## 6  30.8     32.35  -2.1542 -0.9167  0.46667  1.05
## 6  28.7     32.35  -2.1542 -0.9167  0.46667 -1.05
## 7  35.5     32.35   0.3792 -0.9167  0.98333  2.70
## 7  30.1     32.35   0.3792 -0.9167  0.98333 -2.70
## 8  32.6     32.35   3.1125 -0.9167 -1.40000 -0.55
## 8  33.7     32.35   3.1125 -0.9167 -1.40000  0.55
```

```
# Sums of squares
ss_cols <- colSums(tablePVC[, 2:6]^2)
print(ss_cols, digits = 4)
```

```
## grandMean  alphaHat   betaHat  gammaHat     resid
##  50246.02    283.95     20.72     14.34     35.48
```

```
# ANOVA sums of squares
anova(fullFit)
```

```
## Analysis of Variance Table
##
## Response: psize
##                Df  Sum Sq Mean Sq F value    Pr(>F)
## operator        2  20.718  10.359  7.0072   0.00401 **
## resin           7 283.946  40.564 27.4388 5.661e-10 ***
## operator:resin 14  14.335   1.024  0.6926   0.75987
## Residuals      24  35.480   1.478
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Is there statistical evidence that there is an interaction effect between resin and operators?

It's clear to see that, from the ANOVA table, there is no evidence that there is a significant interaction between resin and operators. The F-stat for gammaHat is ~.69 with a p-val of ~.76. Below is a simple calculation for the F-stat of gamma.

```
# SS
SS_gamma <- ss_cols["gammaHat"]
SS_resid <- ss_cols["resid"]

# DFs
df_gamma <- (length(alpha_hat) - 1) * (length(beta_hat) - 1)
df_resid <- N - length(mu_ij_hat)

# MS
MS_gamma <- SS_gamma / df_gamma
MS_resid <- SS_resid / df_resid

# F-stat for gamma
F_gamma <- MS_gamma / MS_resid

F_gamma
```

```
##  gammaHat
## 0.6926437
```