Stat 430 Homework 4

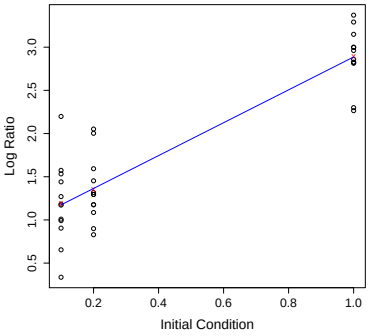
due: November 23, 2009 at 5pm

Problems marked [optional] are to help you learn the material and prepare for exams, but they are not to be turned in.

1. (a) Take your ANOVA analysis of the PND5 virus data from Homework 3 and compare it to a simple linear regression model for the same data, where the initial conditions are treated as a continuous parameter. Plot the data (x-axis initial condition, y-axis log ratio) and the fitted models (for ANOVA, plot the sample means; for linear regression, plot the fitted line). Is there much difference in the fit?

Solution:

```
# read data
d <- read.table("virus.txt")
# extract PND5 data
d.5 <- d[d$V1=="PND5",]
# fit both models
m.anova <- lm(V3 ~ as.factor(V2), data=d.5)
m.lm <- lm(V3 ~ V2, data=d.5)
# plot data
plot(x=d.5$V2, y=d.5$V3, cex.axis=1.3, cex.lab=1.5, xlab="Initial Condition", ylab="Log Ratio")
# plot anova means (also available as coefficients(m.anova))
points(x=0.1, y=mean(d.5$V3[d.5$V2==0.1]), pch="x", col="red")
points(x=0.2, y=mean(d.5$V3[d.5$V2==0.2]), pch="x", col="red")
points(x=1, y=mean(d.5$V3[d.5$V2==1]), pch="x", col="red")
# plot fitted regression line
lines(x=d.5$V2, y=predict(m.lm), col="blue")
# equivalent: lines(x=x, y=coefficients(m.lm)[1] + coefficients(m.lm)[2]*x, col="blue")
```



The plot shows the data, with the ANOVA means as red crosses and the estimated regression as blue line.
There is very little difference in the fit of the two models. The blue line virtually passes through the red crosses.

(b) The ANOVA model has one more parameter than the linear regression model. What F-like statistic could you use to test for a significantly better fit with the ANOVA model? Does the proposed statistic have an F distribution (I'm not asking for a proof, just an argument by analogy to previous F test derivations)? Assuming it does have an F distribution, perform the test.

Solution: The ANOVA model has three parameters: four parameters $\mu, \alpha_1, \alpha_2, \alpha_3$ with one constraint $\alpha_1 + \alpha_2 + \alpha_3 = 0$. The linear model has two parameters: β_0 and β_1.

If the linear model is correct, then the residual sum-of-squares from the linear model can provide an estimate of the population variance.

$$\frac{SS_E^{(lm)}}{36-2}$$

Similarly, the residual sum-of-squares of the anova model can also provide an estimate of the population variance.

$$\frac{SS_E^{(anova)}}{36-3}$$

Further, we expect the ratio of these estimates to be near 1. If the ratio is exceptionally large, there is evidence that the linear model is not a good fit. The code below shows the ratio is not very large. It also, erroneously, compares the ratio to an F distribution, finding the ratio is not unusually large.

```
# comput residual sums-of-squares for both models
> rss.lm <- sum(residuals(m.lm)^2)
> rss.anova <- sum(residuals(m.anova)^2)
> f <- rss.lm/(36-2)/rss.anova*(36-3)
> print(f)
[1] 0.9716719
> 1-pf(f, df1=36-2, df2=36-3)
[1] 0.5335782
```

The above argument is logical and correct up to a point. If you made it, you are clearly understanding a great deal. The problem is that the asymptotic F distribution is not correct for the proposed ratio because the two population variance estimates are not independent. The correct test is described below.

Based on the latest material covered in lecture, we can also look at the partial regression sums-of-squares. So, if $R(\beta_1 \mid \beta_0)$ is the increase in regression sum-of-squares achieved by adding the slope β_1 to the model, then $R(\alpha_1, \alpha_2, \alpha_3, \mu \mid \beta_0, \beta_1)$ (forgive the abuse of notation) is the increase in regression sum-of-squares achieved by adding the third anova parameter to the model. We argued then that

$$\frac{R(\alpha_1, \alpha_2, \alpha_3, \mu \mid \beta_0, \beta_1)}{SS_E^{(anova)}} \sim F(1, 36-3)$$

if the null hypothesis (linear model) is correct.

```
> regss.anova <- sum((predict(m.anova)-mean(d.5$V3))^2)
> regss.lm <- sum((predict(m.lm)-mean(d.5$V3))^2)
> f <- (regss.anova - regss.lm)/rss.anova*(36-3)
> 1-pf(f, df1=1, df2=33)
[1] 0.8489608
```

A short-cut that takes advantage of R's `anova()` function for comparing nested models is:

```
> anova(m.lm, m.anova)
Analysis of Variance Table

Model 1: V3 ~ V2
Model 2: V3 ~ as.factor(V2)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     34 5.3698
2     33 5.3638  1    0.0060 0.0368  0.849
```

2. The **random effects model** for the one-way layout considers the I treatments as sampled from a larger population of treatments. For example, in the PND5 virus, the three initial conditions, 0.1, 0.2, and 1, are the ratio of PND5 and standard virus at the start of the experiment. Thus, the three numbers in the experiment are representative of infinitely many possible initial conditions. The random effects model is

$$Y_{ij} = \mu + A_i + \epsilon_{ij}$$

where A_i is now a random variable (rather than a fixed effect) such that $E[A_i] = 0$ and $\text{Var}(A_i) = E[A_i^2] = \sigma_A^2$, and all $A_1, \ldots, A_I$ are independent of each other. The measurement errors ϵ_{ij} continue to have $E[\epsilon_{ij}] = 0$ and $\text{Var}(\epsilon_{ij}) = \sigma_\epsilon^2$, and they are independent of the random A effects. Given the assumptions, we know

$$\text{Var}(Y_{ij}) = \sigma_A^2 + \sigma_\epsilon^2$$

(a) Show that

$$E[MS_W] = \sigma_\epsilon^2 \qquad\qquad E[MS_B] = \sigma_\epsilon^2 + J\sigma_A^2$$

Solution: The following derivation uses matrix algebra. One can also show the result by working with the individual terms in the sums-of-squares. The key is to correctly identify the form of the covariance matrix Σ_{YY} which is no longer $\sigma^2 I$ as for the fixed effect model.

First, notice that for $j \neq j'$,

$$\begin{aligned}
\text{Cov}(Y_{ij}, Y_{ij'}) &= \text{Cov}(\mu + A_i + \epsilon_{ij}, \mu + A_i + \epsilon_{ij'}) \\
&= \text{Cov}(A_i, A_i) + \text{Cov}(A_i, \epsilon_{ij'}) + \text{Cov}(\epsilon_{ij}, A_i) + \text{Cov}(\epsilon_{ij}, \epsilon_{ij'}) \\
&= \sigma_A^2
\end{aligned}$$

because of independence of everything but A_i with itself, so

$$\Sigma_{Y_{i\cdot}Y_{i\cdot}} = \begin{pmatrix} \sigma_A^2 + \sigma_\epsilon^2 & \sigma_A^2 & \cdots & \sigma_A^2 \\ \sigma_A^2 & \sigma_A^2 + \sigma_\epsilon^2 & & \vdots \\ \vdots & & \ddots & \sigma_A^2 \\ \sigma_A^2 & \cdots & \sigma_A^2 & \sigma_A^2 + \sigma_\epsilon^2 \end{pmatrix}$$

and

$$\Sigma_{YY} = \begin{pmatrix} \Sigma_{Y_{1\cdot}Y_{1\cdot}} & 0 & \cdots & 0 \\ 0 & \Sigma_{Y_{2\cdot}Y_{2\cdot}} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \Sigma_{Y_{I\cdot}Y_{I\cdot}} \end{pmatrix}.$$

Now, we carry out the derivation.

$$\begin{aligned}
E[SS_W] &= \sum_{i=1}^{I}\sum_{j=1}^{J} E\left[(Y_{ij} - \bar{Y}_{i\cdot})^2\right] \\
&= \sum_{i=1}^{I} E\left[Y_{i\cdot}^T \left(I - \frac{1}{J}\mathbf{1}'\mathbf{1}\right) Y_{i\cdot}\right] \\
&= \sum_{i=1}^{I} \left\{\text{trace}\left[\left(I - \frac{1}{J}\mathbf{1}'\mathbf{1}\right)\Sigma_{Y_{i\cdot}Y_{i\cdot}}\right] + (E[Y_{i\cdot}])^T \left(I - \frac{1}{J}\mathbf{1}'\mathbf{1}\right) E[Y_{i\cdot}]\right\} \\
&= \sum_{i=1}^{I} \left[\text{trace}(I\Sigma_{Y_{i\cdot}Y_{i\cdot}}) - \text{trace}\left(\frac{1}{J}\mathbf{1}'\mathbf{1}\Sigma_{Y_{i\cdot}Y_{i\cdot}}\right)\right] \\
&= \sum_{i=1}^{I} \left[J(\sigma_A^2 + \sigma_\epsilon^2) - (\sigma_A^2 + \sigma_\epsilon^2) - (J-1)\sigma_A^2\right] \\
&= \sum_{i=1}^{I} (J-1)\sigma_\epsilon^2 \\
&= I(J-1)\sigma_\epsilon^2
\end{aligned}$$

For the between sum-of-squares, let $Y^T = (Y_{11}, \ldots, Y_{1J}, Y_{21}, \ldots, Y_{I-1,J}, Y_{I1}, \ldots, Y_{IJ})$ and $\text{diag}(\mathbf{1}'\mathbf{1})$ be a $IJ \times IJ$ matrix consisting of I $J \times J$ matrices of 1's arranged along the diagonal with 0 everywhere else.

$$\begin{aligned}
E[SS_B] &= \sum_{i=1}^{I}\sum_{j=1}^{J} E\left[(\bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot})^2\right] \\
&= \sum_{i=1}^{I} JE\left[(\bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot})^2\right] \\
&= E\left[Y^T \left(\frac{1}{J}\text{diag}(\mathbf{1}'\mathbf{1}) - \frac{1}{IJ}\mathbf{1}'\mathbf{1}\right) Y\right] \\
&= \text{trace}\left(\frac{1}{J}\text{diag}(\mathbf{1}'\mathbf{1})\Sigma_{YY}\right) - \text{trace}\left(\frac{1}{IJ}\mathbf{1}'\mathbf{1}\Sigma_{YY}\right) \\
&= \left(\frac{1}{J} - \frac{1}{IJ}\right)\text{trace}\left(\mathbf{1}'\mathbf{1}\Sigma_{YY}\right) \\
&= \frac{I-1}{IJ}\left[\sigma_A^2 + \sigma_\epsilon^2 + (J-1)\sigma_A^2\right] IJ = (I-1)(\sigma_\epsilon^2 + J\sigma_A^2)
\end{aligned}$$

(b) Use the above to produce estimators of both variance components (σ^2_ϵ and σ^2_A).

Solution:

$$\begin{aligned} \hat{\sigma}^2_\epsilon &= \frac{SS_W}{I(J-1)} = MS_W \\ \hat{\sigma}^2_A &= \frac{MS_B - MS_W}{J} \end{aligned}$$

(c) The virus data does not satisfy the random effects model, because in question 1, you find preliminary evidence that $E[A_i]$ depends on the level i. To demonstrate the random effects model, access the dye data. This dataset tests the strength of dye across batches. To measure strength, the product was used to dye a square of cloth. The resulting fabric was visually assessed and given a numeric score by experts. Large samples of dye were taken from six batches. The samples were well-mixed, and six random subsamples were taken from each large sample. The 36 subsamples were submitted to a laboratory for testing in random order. Estimate μ, σ^2_ϵ, and σ^2_A for this model.

Solution:

```
# \mu
> mean(d$Strength)
[1] 93.61111
> m <- lm(Strength ~ as.factor(Batch), data=d)
> MSW <- sum(residuals(m)^2)/30
# \sigma_\epsilon^2
> MSW
[1] 4.469444
> MSB <- sum((predict(m) - mean(d$Strength))^2)/5
> MSB
[1] 83.09444
# \sigma_A^2
> (MSB-MSW)/6
[1] 13.10417
```

Alternatively, one can use the `lme4` package to get the same results via `lmer`. However, learning how to use `lme4` and related package `nlme` is no walk in the park.

```
> m.mer <- lmer(Strength ~ 1 | Batch, data=d)
> summary(m.mer)
Linear mixed model fit by REML
Formula: Strength ~ 1 | Batch
   Data: d
```

```
   AIC   BIC logLik deviance REMLdev
 175.9 180.7 -84.96    172.6   169.9
Random effects:
 Groups   Name        Variance Std.Dev.
 Batch    (Intercept) 13.1033  3.6198
 Residual              4.4695  2.1141
Number of obs: 36, groups: Batch, 6

Fixed effects:
            Estimate Std. Error t value
(Intercept)   93.611      1.519   61.62
```

(d) Suggest and perform a test for $H_0 : \sigma_A^2 = 0$.

Solution: If $\sigma_A^2 = 0$, then both MS_W and MS_B estimate σ_ϵ^2, and specifically

$$f = \frac{MS_B}{MS_W} \sim F(I-1, I(J-1)) = F(5, 30).$$

In this case, $f = 18.59167$, which on a one-sided test, yields p-value 2.16×10^{-08}.

(e) [optional] Propose and implement a computer-based resampling test to test the same null $H_0 : \sigma_A^2 = 0$.

Solution: If $\sigma_A^2 = 0$, then there is no batch effect and datasets with Strength values permuted arbitrarily should have f values similar to the one observed. It is not surprising that a call to the following code shows all 100 resampled datasets have f statistics below the one observed above.

```
d <- read.table("dye.txt", header=T)
mu <- mean(d$Strength)
f.obs <- MSB/MSW   # using calculations from before

B <- 100
f.p <- NULL
# permute Strengths B times
for(i in 1:B) {
   s <- sample(d$Strength)
   d$Permuted.Strength <- s
   m.p <- lm(Permuted.Strength ~ as.factor(Batch), data=d)
   MSW.p <- sum(residuals(m.p)^2)/30
   MSB.p <- sum((predict(m.p) - mu)^2)/5
   f.p[i] <- MSB.p/MSW.p
```

```
}
print(f.p)
# output times permuted data more extreme batch differences
print(sum(f.obs>f.p))
```

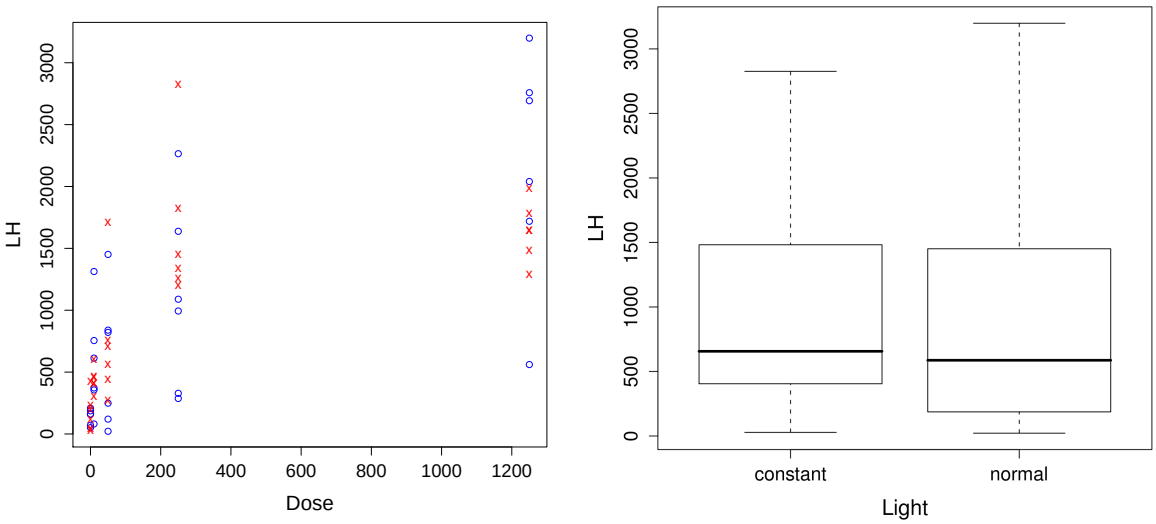
Notice, the justification given for not using the virus data was flawed. That data could have been used to yield:

```
> lmer(V3 ~ 1 | V2, data=d.5)
Linear mixed model fit by REML
Formula: V3 ~ 1 | V2
Data: d.5
    AIC   BIC logLik deviance REMLdev
  53.67 58.42 -23.84    48.27   47.67
Random effects:
 Groups   Name        Variance Std.Dev.
 V2       (Intercept) 0.86750  0.93140
 Residual             0.16254  0.40316
Number of obs: 36, groups: V2, 3

Fixed effects:
            Estimate Std. Error t value
(Intercept)   1.8083     0.5419   3.337
> MSW <- 0.16254
> MSB <- 0.86750*9+MSW
> f <- MSB/MSW
> 1-pf(f, df1=2, df2=3*8)
[1] 3.336451e-09
```

3. Female rats exposed to different lighting cycles were dosed with 0 (control saline solution), 10, 50, 250, or 1250 ng luteinizing releasing factor (LRF). The amount of luteinizing hormone (LH) was measured in the blood at a later time. Use ANOVA to analyze the data to determine the effect of the light regimen and LRF dose on LH release. Perform a non-parametric test as well, and assess your findings.

Solution: First, we plot the data. We check for an interaction by coloring the data points in the Dose by LH plot by their light condition (red 'x' indicates constant light). There is no obvious interaction effect, except a small possibility that the dose effect saturates faster (and lower?) in constant light conditions.



In the code below, we fit the model with and without interaction and use `anova()` to perform an F test for the interaction effect. Since the result is not significant, we display the additive model fit. The light effect is clearly not important, but dose has a substantial impact on LH levels. The Kruskal-Wallis tests confirm these results on additive effects without making assumptions of normality.

```
# fit model with interaction
> m.int <- aov(LH ~ as.factor(Dose) * Light, data=d)
# fit model with additive effects only
> m.add <- aov(LH ~ as.factor(Dose) + Light, data=d)
# compare models
> anova(m.add, m.int)
Analysis of Variance Table

Model 1: LH ~ as.factor(Dose) + Light
Model 2: LH ~ as.factor(Dose) * Light
  Res.Df      RSS Df Sum of Sq      F Pr(>F)
1     54 15343835
2     50 13480497  4   1863338 1.7278 0.1586
> summary(m.add)
                 Df   Sum Sq  Mean Sq F value    Pr(>F)
as.factor(Dose)  4 23790247  5947562 20.9314 1.829e-10 ***
Light            1     3450     3450  0.0121    0.9127
Residuals       54 15343835   284145
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
# manual Kruskal using asymptotic chi-square
> d$R <- rank(d$LH)
> m.2 <- aov(R ~ as.factor(Dose)+Light, data=d)
> 1-pchisq(12*11674.3/60/61, df=4)
[1] 9.826692e-08
> 1-pchisq(12*153.6/60/61, df=1)
[1] 0.4779197

# same, using kruskal.test() function
> kruskal.test(LH ~ Light, data=d)

Kruskal-Wallis rank sum test

data:  LH by Light
Kruskal-Wallis chi-squared = 0.5036, df = 1, p-value = 0.4779

> kruskal.test(LH ~ as.factor(Dose), data=d)

Kruskal-Wallis rank sum test

data:  LH by as.factor(Dose)
Kruskal-Wallis chi-squared = 38.2786, df = 4, p-value = 9.816e-08
```