Stat 505 Assignment 2

Excellent!
one typo or arithmetic error
39 of 40

Leslie Gains-Germain

Due: Sept 12, 2014

1. Construct a 4 by 4 variance-covariance matrices showing **R, D**, and **V** for each case:

   (a) so that covariances are all zeroes, variances are 1, 4, 9, 16.

$$\boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \quad \boldsymbol{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{V} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 16 \end{bmatrix}$$

   (b) so that each variance is .25 and all correlations are .6.

$$\boldsymbol{D} = \begin{bmatrix} .5 & 0 & 0 & 0 \\ 0 & .5 & 0 & 0 \\ 0 & 0 & .5 & 0 \\ 0 & 0 & 0 & .5 \end{bmatrix} \quad \boldsymbol{R} = \begin{bmatrix} 1 & 0.6 & 0.6 & 0.6 \\ 0.6 & 1 & 0.6 & 0.6 \\ 0.6 & 0.6 & 1 & 0.6 \\ 0.6 & 0.6 & 0.6 & 1 \end{bmatrix} \quad \boldsymbol{V} = \begin{bmatrix} .25 & 0.15 & 0.15 & 0.15 \\ 0.15 & .25 & 0.15 & 0.15 \\ 0.15 & 0.15 & .25 & 0.15 \\ 0.15 & 0.15 & 0.15 & .25 \end{bmatrix}$$

   (c) so that each variance is 9, neighboring observations have covariance 3, observations 2 steps apart have covariance 1, and the covariance between observations 1 and 4 is $\frac{1}{3}$.

$$\boldsymbol{D} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \quad \boldsymbol{R} = \begin{bmatrix} 1 & 1/3 & .111 & 0.037 \\ 1/3 & 1 & 1/3 & 0.111 \\ 0.111 & 1/3 & 1 & 1/3 \\ 0.037 & 0.111 & 1/3 & 1 \end{bmatrix} \boldsymbol{V} = \begin{bmatrix} 9 & 3 & 1 & \frac{1}{3} \\ 3 & 9 & 3 & 1 \\ 1 & 3 & 9 & 3 \\ \frac{1}{3} & 1 & 3 & 9 \end{bmatrix}$$

   (d) so that correlations are the same as in (c), but variances are $\mu_i^{1.4}$ where the vector of means is $\boldsymbol{\mu} = (2, 3, 7, 6)^T$.

1.4, not 1/4 -1

$$\boldsymbol{D} = \begin{bmatrix} \sqrt{2^{1/4}} & 0 & 0 & 0 \\ 0 & \sqrt{3^{1/4}} & 0 & 0 \\ 0 & 0 & \sqrt{7^{1/4}} & 0 \\ 0 & 0 & 0 & \sqrt{6^{1/4}} \end{bmatrix} \quad \boldsymbol{R} = \begin{bmatrix} 1 & 1/3 & .111 & 0.037 \\ 1/3 & 1 & 1/3 & 0.111 \\ 0.111 & 1/3 & 1 & 1/3 \\ 0.037 & 0.111 & 1/3 & 1 \end{bmatrix}$$

$$\boldsymbol{V} = \begin{bmatrix} 2^{1.4} & 1.17 & .705 & 0.211 \\ 1.17 & 3^{1.4} & 2.81 & 0.840 \\ .705 & 2.81 & 7^{1.4} & 4.56 \\ .211 & .840 & 4.56 & 6^{1.4} \end{bmatrix}$$

2. Let $\boldsymbol{z} \sim N_8(\mathbf{0}, 4\boldsymbol{I}_8)$ be a random vector and let $\boldsymbol{\Sigma}$ be a 4 by 8 matrix with these entries:

```
1 1 0 0 1 0 0 0
0 1 1 0 1 1 0 0
0 0 1 1 0 1 1 0
1 0 0 1 0 0 1 1
```

Describe the distribution of

$$\mathbf{x} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 7 \end{bmatrix} + \boldsymbol{\Sigma}\boldsymbol{z}$$

$\mathbf{x}$ *follows a normal distribution with mean*

$$\begin{bmatrix} 1 \\ 3 \\ 2 \\ 7 \end{bmatrix}$$

with the following variance-covariance matrix $\Sigma * 4I_8 * \Sigma^T =$

```
12 8 0 4
8 16 8 0
0 8 16 8
4 0 8 16
```

The variances, $\sigma^2_{x_1}, \sigma^2_{x_2}, \sigma^2_{x_3},$ *and* $\sigma^2_{x_4}$ *are on the diagonal. Neighboring observations in* $\mathbf{x}$ *have covariance* 8*, observations two steps apart have covariance* 0*, and the covariance between observations* 1 *and* 4 *is* 4*.*

Yes, covariance just depends on the number of 'shared' z's.

3. Fill in the blanks. You can use the verbatim environment

```
Source            df       SS       MS       F    p-value
---------------------------------------------------------


Between groups   3    20.42      6.8067    2.409    0.068


Within groups   217   613.14     2.8255
---------------------------------------------------------
Total           220   633.56
```

4. Set up the ANOVA model $y_{ij} = \mu + \tau_i + \epsilon ij,\ i = 1, ...4, j = 1, 2$ in R with these data:

```
y <- c(3, 4, 1, 0, 7, 9, 6, 5)
#f <- factor(rep(LETTERS[1:4], each = 2))
  ## or use the gl function:
f <- gl(4,2, labels= LETTERS[1:4])
Xf <- model.matrix( y ~ f)
X <-  cbind(1,model.matrix(y ~ f + 0))
```

(a) Fit using `lm(y ~ f)`. Give estimates of all cell means, explaining how they relate to the output.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.5000	0.6614	5.29	0.0061
fB	-3.0000	0.9354	-3.21	0.0327
fC	4.5000	0.9354	4.81	0.0086
fD	2.0000	0.9354	2.14	0.0993

The mean of y in group A is estimated to be 3.5*. This is given as "intercept" in the output above. The mean of y in group B is estimated to be* $3.5 - 3.0 = 0.5$*. The mean of y in group C is estimated to be* $3.5 + 4.5 = 8$*. The mean of y in group D is estimated to be* $3.5 + 2 = 5.5$*.*

(b) Do the same using `lm( y ~ f + 0)`.

The output below gives the estimates of the cell means for each group.

and mu.hat is set to 0

	Estimate	Std. Error	t value	Pr(>|t|)
fA	3.5000	0.6614	5.29	0.0061
fB	0.5000	0.6614	0.76	0.4918
fC	8.0000	0.6614	12.09	0.0003
fD	5.5000	0.6614	8.32	0.0011

(c) Is $X_f^T X_f$ nonsingular (invertible)? If so, solve the normal equations based on X_f using `crossprod` and `solve` functions. If not, find a solution using the Moore – Penrose generalized inverse.

Yes, $X_f^T X_f$ is nonsingular because X_f is built under the constraint that $\tau_1 = 0$. I found the least squares solution using crossprod and solve and I got the same output as the reference coded model in (a).

```
##              [,1]
## (Intercept)  3.5
## fB          -3.0
## fC           4.5
## fD           2.0
```

(d) Repeat (a) using `lm(y ~ X + 0, singular.ok=T)`.

We can see from the output below that group D is the reference level. The estimate for the mean of group A is $5.5 - 2 = 3.5$. The estimate for the mean of group B is $5.5 - 5 = 0.5$, and the estimate for the mean of group C is $5.5 + 2.5 = 8$.

	Estimate	Std. Error	t value	Pr(>|t|)
x	5.5000	0.6614	8.32	0.0011
xfA	-2.0000	0.9354	-2.14	0.0993
xfB	-5.0000	0.9354	-5.35	0.0059
xfC	2.5000	0.9354	2.67	0.0557

(e) Use the Moore Penrose inverse to find another solution to the normal equations for the full $\boldsymbol{X}$ matrix. Show that $\hat{\mu}$ and $\hat{\tau}_i$ differ from those above, but estimates of the cell means are the same.

```
##      [,1]
## [1,]  3.5
## [2,]  0.0
## [3,] -3.0
## [4,]  4.5
## [5,]  2.0
```

We see that $\hat{\mu} = 3.5$, $\widehat{\tau_A} = 0$, $\widehat{\tau_B} = -3.0$, $\widehat{\tau_C} = 4.5$, and $\widehat{\tau_D} = 2.0$. τ_D was set to 0 in part (d), but here τ_A is set to zero. As a result, we get different estimates *for μ, τ_A, τ_B, τ_C, and τ_D, but the estimated cell means $(\widehat{\mu + \tau_i})$ are the same $(3.5, 0.5, 8.0,$ and 5.5 for groups A, B, C, and D).*

(f) Explain how `Xf` and `X` differ and how they are used to create the estimates in parts a, b, d, and e.

The first column of X is a linear combination of the other four columns. The columns of the model matrix, X_f, are linearly independent because X_f is simply the X matrix with the second column removed. By removing the second column of X, the computer essentially imposes the constraint $\tau_A = 0$. This is what we see in parts (a) and (c). We see in part (e) that the Moore-Penrose generalized inverse also uses the constraint $\tau_A = 0$ to solve the normal equations for the full X matrix. In part (d), the computer uses the constraint $\tau_D = 0$. Although the parameter estimates differ depending on the constraint, all methods give the same estimates for the cell means.

R Code

```
a <- c(sqrt(2^1.4),0,0,0)
b <- c(0,sqrt(3^1.4),0,0)
c <- c(0,0,sqrt(7^1.4),0)
d <- c(0,0,0,sqrt(6^1.4))

di <- rbind(a,b,c,d)

e <- c(1,1/3,1/9,1/27)
f <- c(1/3,1,1/3,1/9)
g <- c(1/9, 1/3,1,1/3)
h <- c(1/27, 1/9, 1/3, 1)

r <- rbind(e,f,g,h)

v <- di%*%r%*%di
```

```
y <- c(3, 4, 1, 0, 7, 9, 6, 5)
#f <- factor(rep(LETTERS[1:4], each = 2))
  ## or use the gl function:
f <- gl(4,2, labels= LETTERS[1:4])
Xf <- model.matrix( y ~ f)
X <-  cbind(1,model.matrix(y ~ f + 0))
```

```
y <- c(3, 4, 1, 0, 7, 9, 6, 5)
#f <- factor(rep(LETTERS[1:4], each = 2))
  ## or use the gl function:
f <- gl(4,2, labels= LETTERS[1:4])
```

```r
require(xtable)
lm.fit <- lm(y~f)
xtable(summary(lm.fit))
```

```r
y <- c(3, 4, 1, 0, 7, 9, 6, 5)
f <- factor(rep(LETTERS[1:4], each = 2))
## or use the gl function:
lm.fit5 <- lm(y~f + 0)
require(xtable)
print(xtable(summary(lm.fit5)))
#anova(lm.fit1)
```

```r
xf <- model.matrix( y ~ f)
det <- crossprod(xf)
solve(det)%*%t(xf)%*%y
y <- c(3, 4, 1, 0, 7, 9, 6, 5)
```

```r
x <-  cbind(1,model.matrix(y ~ f + 0))
lm.sing <- lm(y~x+0, singular.ok=T)
require(xtable)
xtable(summary(lm.sing))
```

```r
det1 <- crossprod(x)
require(MASS)
inverse <- ginv(det1)
zapsmall(inverse%*%t(x)%*%y)
```