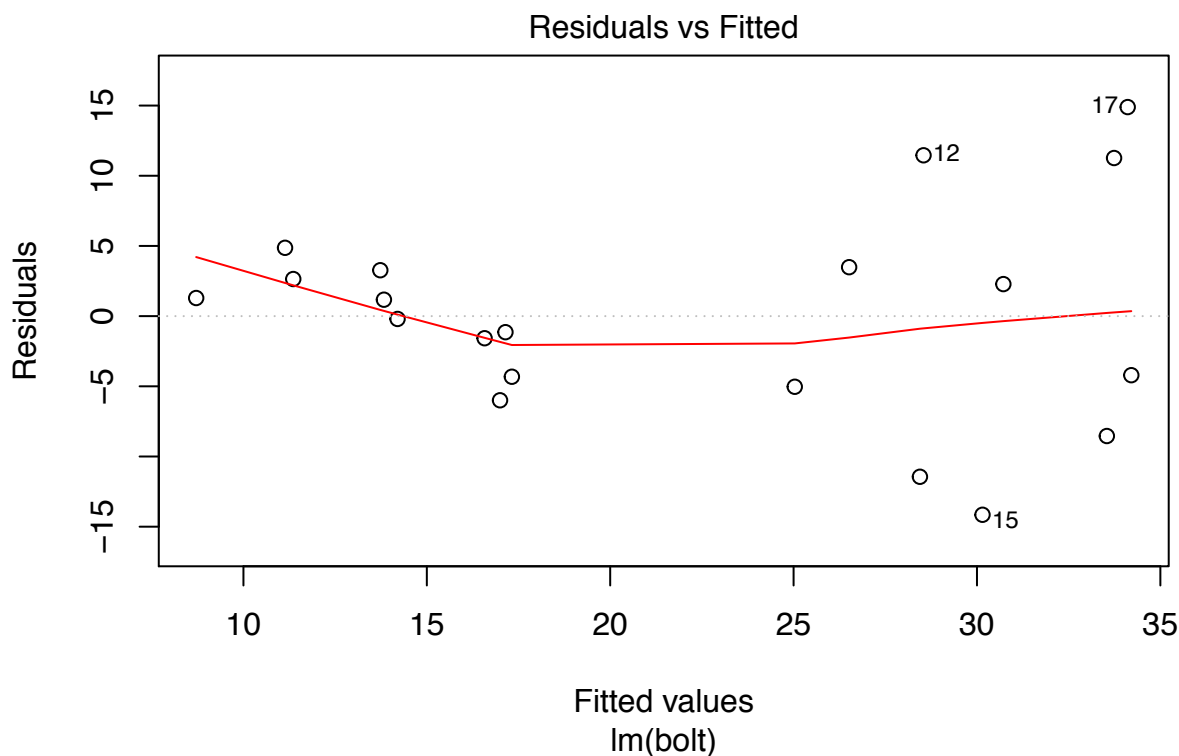


# Stat 424: Homework 5

*Evangeline Lim*

11/8/2019

## Question 1 (Textbook #14)



The residuals plot shows that homoskedasticity is violated because of its fanning out shape, meaning that the variance is uneven throughout the dataset. Linearity seems to be relatively okay because the line of best fit is pretty straight.

## Question 2 (Textbook 16)

For the two factors A and B, let  $a_i = \text{numerical value of } i\text{th level of } A$  and  $b_j = \text{numerical value of } j\text{th level of } B$  and  $i, j = 1, 2, 3$  since each are quantitative at three levels. Therefore, the set of variables for the regression model will look like  $1, a_i, a_i^2, b_j, b_j^2, a_i b_j, a_i b_j^2, a_i^2 b_j, a_i^2 b_j^2$  for their linear and quadratic effects, and also interactions between A and B. Assuming  $n = 1$ , the model matrix will be:

$$Y = X\beta = \begin{bmatrix} y_{(111)} \\ y_{(121)} \\ y_{(131)} \\ y_{(211)} \\ y_{(221)} \\ y_{(231)} \\ y_{(311)} \\ y_{(321)} \\ y_{(331)} \end{bmatrix}$$

$$Y = \begin{bmatrix} 1 & a_1 & a_1^2 & b_1 & b_1^2 & a_1b_1 & a_1b_1^2 & a_1^2b_1 & a_1^2b_1^2 \\ 1 & a_1 & a_1^2 & b_2 & b_2^2 & a_1b_2 & a_1b_2^2 & a_1^2b_2 & a_1^2b_2^2 \\ 1 & a_1 & a_1^2 & b_3 & b_3^2 & a_1b_3 & a_1b_3^2 & a_1^2b_3 & a_1^2b_3^2 \\ 1 & a_2 & a_2^2 & b_1 & b_1^2 & a_2b_1 & a_2b_1^2 & a_2^2b_1 & a_2^2b_1^2 \\ 1 & a_2 & a_2^2 & b_2 & b_2^2 & a_2b_2 & a_2b_2^2 & a_2^2b_2 & a_2^2b_2^2 \\ 1 & a_2 & a_2^2 & b_3 & b_3^2 & a_2b_3 & a_2b_3^2 & a_2^2b_3 & a_2^2b_3^2 \\ 1 & a_3 & a_3^2 & b_1 & b_1^2 & a_3b_1 & a_3b_1^2 & a_3^2b_1 & a_3^2b_1^2 \\ 1 & a_3 & a_3^2 & b_2 & b_2^2 & a_3b_2 & a_3b_2^2 & a_3^2b_2 & a_3^2b_2^2 \\ 1 & a_3 & a_3^2 & b_3 & b_3^2 & a_3b_3 & a_3b_3^2 & a_3^2b_3 & a_3^2b_3^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_a \\ \beta_a^2 \\ \beta_b \\ \beta_b^2 \\ \beta_{(ab)} \\ \beta_{(ab^2)} \\ \beta_{(a^2b)} \\ \beta_{(a^2b^2)} \end{bmatrix}$$

### Question 3 (Textbook 17)

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##   select

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##   recode

##
## Call:
## lm(formula = y ~ x1 + x2 + x3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -644.6  -279.1  -150.1   199.6  1268.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1048.0      220.3   4.757 8.54e-05 ***
## x1              660.0      115.1   5.736 7.66e-06 ***
## x2             -535.9      115.1  -4.657 0.000109 ***
## x3             -310.8      115.1  -2.701 0.012751 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 488.2 on 23 degrees of freedom
## Multiple R-squared:  0.7291, Adjusted R-squared:  0.6937
## F-statistic: 20.63 on 3 and 23 DF,  p-value: 1.029e-06
```

```
## [1] 19 20
```

```
## [1] 9 8
```

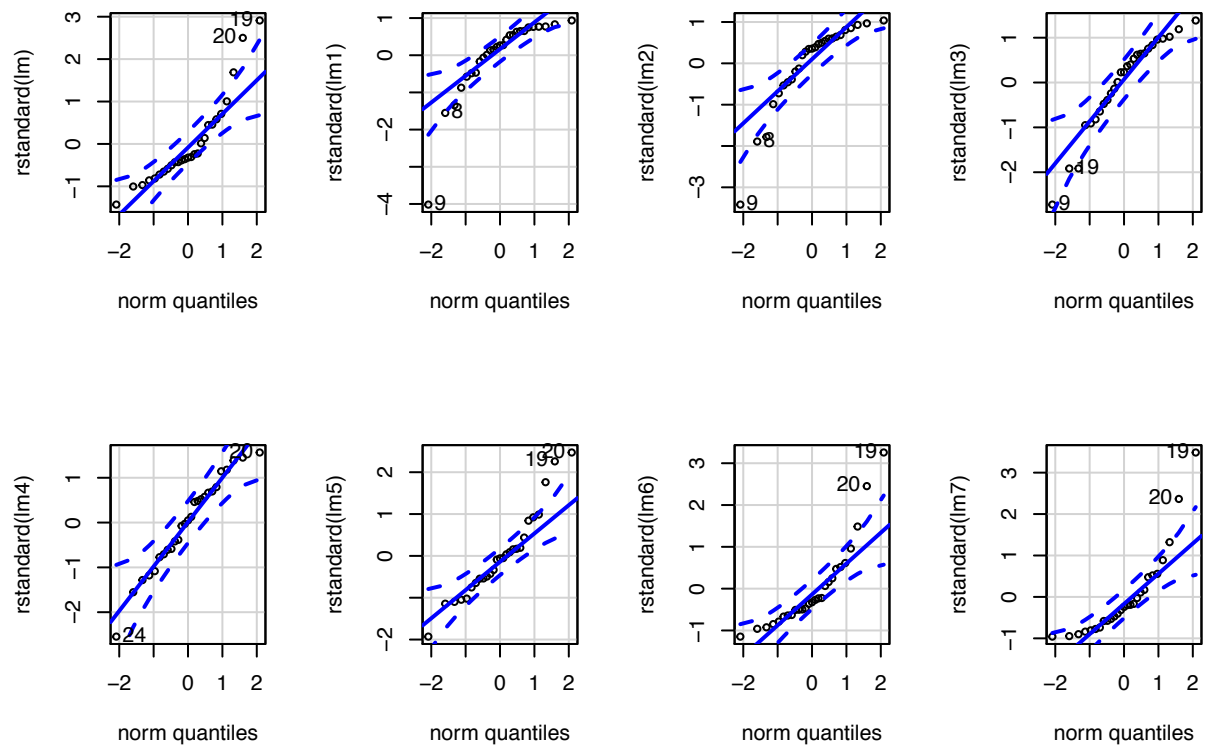
```
## [1] 9 8
```

```
## [1] 9 19
```

```
## [1] 24 20
```

```
## [1] 20 19
```

```
## [1] 19 20
```



```
## [1] 19 20
```

```
## [1] 423.0677
```

```
## [1] -532.1996
```

```
## [1] -269.6112
```

```
## [1] -142.6942
```

```
## [1] -2.164296
```

```
## [1] 209.8421
```

```
## [1] 633.6747
```

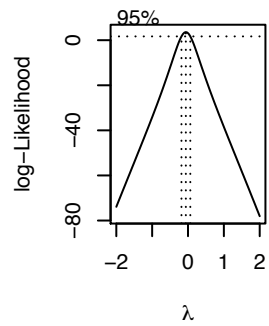
```
## [1] 844.4019
```

```
##          lambda          lik
```

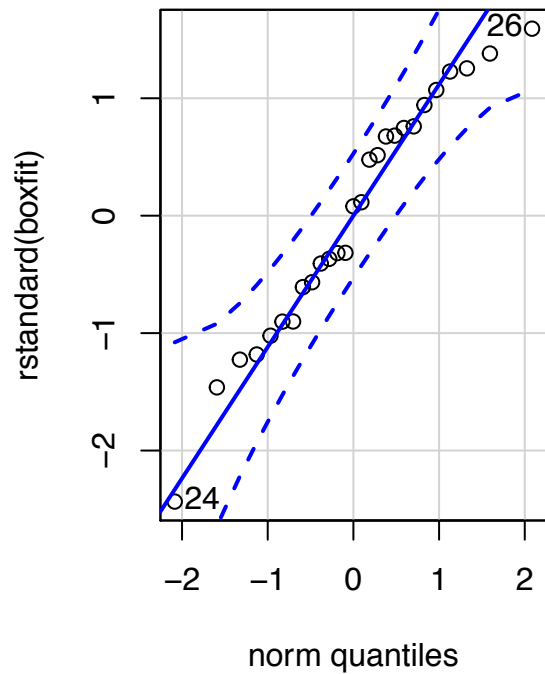
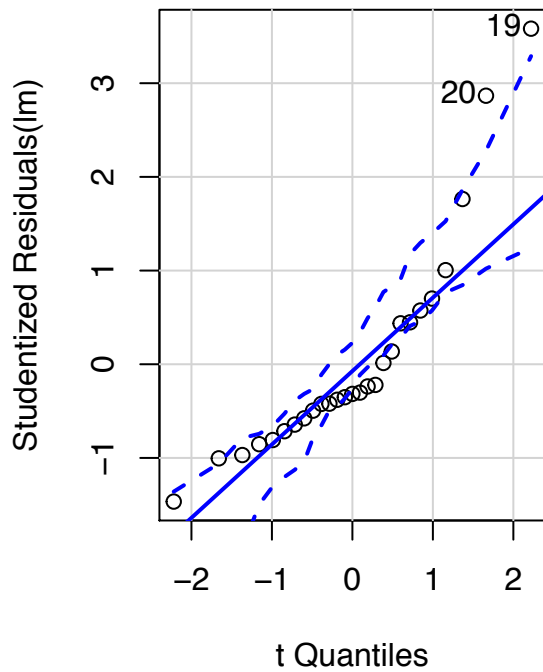
```
## [1,] -0.06060606  3.5966255
```

```
## [2,] -0.02020202  3.3962071
## [3,] -0.10101010  3.3703897
## [4,]  0.02020202  2.7809559
## [5,] -0.14141414  2.7210236
## [6,]  0.06060606  1.7931648
## [7,] -0.18181818  1.7040835
## [8,]  0.10101010  0.5055984
## [9,] -0.22222222  0.3871777
## [10,] 0.14141414 -1.0101715

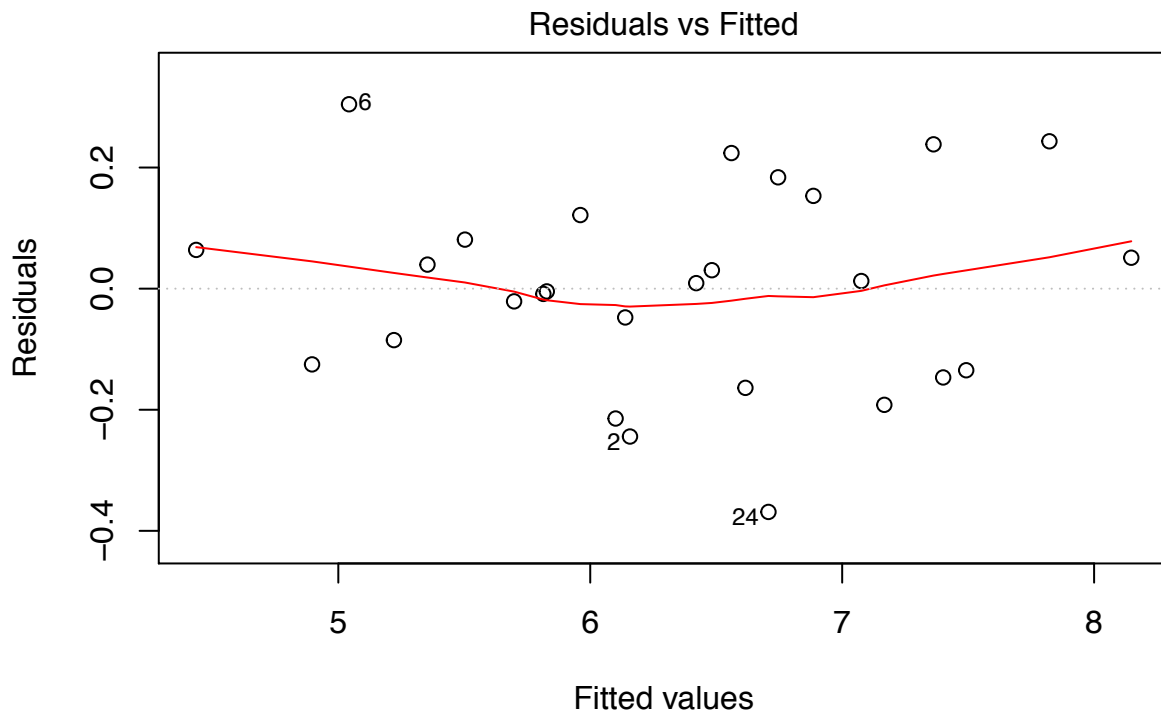
## [1] -23.82085
## [1] 423.0677
## [1] -532.1996
## [1] -269.6112
## [1] -142.6942
## [1] -2.164296
## [1] 209.8421
## [1] 633.6747
## [1] 844.4019
```



```
## [1] 19 20
```



```
## [1] 24 26
```



`lm(log(cycles) ~ as.factor(length) + as.factor(amplitude) + as.factor(load) ...`

The best transformation is the log transformation where  $\lambda = 0$ . This is because it gives the model with the least effects and interaction. The qqplot looks the msot normal and there are no obvious extremities or violations in linearity and homoskedacity in the residuals plot.

#### Question 4 (Textbook 19)

```
##
## Pairwise comparisons using t tests with pooled SD
##
## data: wear$wear and wear$material
##
##      A      B      C
## B 0.0071 -      -
## C 0.2821 0.4079 -
## D 0.0417 1.0000 1.0000
##
## P value adjustment method: bonferroni
## [1] 3.862991
## [1] A vs B = -8.27
## [3] A vs C = -4.34
## [5] A vs D = -6.37
## [7] B vs C = -3.93
## [9] B vs D = 1.9
## [11] C vs D = -2.03
## [13] T critical for Tukey =3.46
```

Both Tukey and Bonferroni renders the same results, that is AB and AD are the only ones that are significant. However, we are interested in all pairwise comparisons for Latin Square designs, so we would prefer Tukey over Bonferroni. We also see that the Tukey critical value is smaller than the Bonferroni critical value. This means Tukey is more favourable for latin squares.

## Question 5

**Table 3.22 ANOVA Table for Graeco-Latin Square Design**

Source	Degrees of Freedom	Sum of Squares
Row	$k - 1$	$k \sum_{i=1}^k (\bar{y}_{i...} - \bar{y}_{....})^2$
Column	$k - 1$	$k \sum_{j=1}^k (\bar{y}_{.j..} - \bar{y}_{....})^2$
Latin letter	$k - 1$	$k \sum_{l=1}^k (\bar{y}_{..l.} - \bar{y}_{....})^2$
Greek letter	$k - 1$	$k \sum_{m=1}^k (\bar{y}_{...m} - \bar{y}_{....})^2$
Residual	$(k - 3)(k - 1)$	By subtraction
Total	$k^2 - 1$	$\sum_{(i,j,l,m) \in S} (y_{ijlm} - \bar{y}_{....})^2$

## Question 6 (Textbook 29)

- a) I would use the Balanced Incomplete Block Design. We use BIBD when we have  $t$  treatments,  $b$  blocks of size  $k$  where  $t > k$ , with each treatment replicated  $r$  times. This experiment has 6 brands ( $t=6$ ),  $k=3$  because each tester can only try 3 brands, and 10 testers ( $b = 10$ ). So,  $t$  is indeed greater than  $k$ . By using  $bk = rt$ , we find that we should replicate each brand tasted  $r=5$  times, meaning in total, every brand should be tasted 5 times. We also know that  $r(k - 1) = \lambda(t - 1)$ . So, the pair of treatments that appear in the same number of blocks should be  $\lambda = 2$ . This experimental design is good because it gives us the most diverse set of data in a situation where it is difficult to form blocks of large size, since we only have 10 tasters.
- b) I would change the BIBD to a Randomized Block Design. In this design, if we stick to having each taster only being able to taste a maximum of 3 brands, ( $k=3$ ,  $b=7$ ), I would recommend doing the entire experiment twice in which each taster gets the different sets of ice cream brands. For example, in the first round, the 7 testers taste brands 1,2,3 and then repeat this and let them taste set 4,5,6. In my opinion, this would give us the most diverse set of data given the unfortunate situation.

## Appendix

```
knitr::opts_chunk$set(echo = F)
library(dplyr)
bolt <- read.table("bolt.txt", header = T)
bolt_fitted <- lm(bolt)
plot(bolt_fitted, which=1)

library(MASS)
library(car)

# Box-Cox Transformation
yarn = read.table("yarn.txt", header = T)
y = yarn$cycles
x1 = yarn$length
x2 = yarn$amplitude
x3 = yarn$load

# not transformed
lm = lm(y~x1+x2+x3)
summary(lm)

# transforming the data to find best
lambda1 = -2
Trans.y1 = (y^lambda1-1)/lambda1
lm1 = lm(Trans.y1~x1+x2+x3)

lambda2 = -1
Trans.y2 = (y^lambda2-1)/lambda2
lm2 = lm(Trans.y2~x1+x2+x3)

lambda3 = -1/2
Trans.y3 = (y^lambda3-1)/lambda3
lm3 = lm(Trans.y3~x1+x2+x3)

lambda4 = 0
```



```

Trans.y4 = log(y)
lm4 = lm(Trans.y4~x1+x2+x3)

lambda5 = 1/2
Trans.y5 = (y^lambda5-1)/lambda5
lm5 = lm(Trans.y5~x1+x2+x3)

lambda6 = 3/2
Trans.y6 = (y^lambda6-1)/lambda6
lm6 = lm(Trans.y6~x1+x2+x3)

lambda7 = 2
Trans.y7 = (y^lambda7-1)/lambda7
lm7 = lm(Trans.y7~x1+x2+x3)

par(mfrow=c(2,4))
qqPlot(rstandard(lm))
qqPlot(rstandard(lm1))
qqPlot(rstandard(lm2))
qqPlot(rstandard(lm3))
qqPlot(rstandard(lm4))
qqPlot(rstandard(lm5))
qqPlot(rstandard(lm6))
qqPlot(rstandard(lm7))

# Compare
BIC(lm)
BIC(lm1)
BIC(lm2)
BIC(lm3)
BIC(lm4)
BIC(lm5)
BIC(lm6)
BIC(lm7)

# finding the best lambda for the model
b = boxcox(y~x1+x2+x3)
lambda = b$x # lambda values
lik = b$y # log likelihood values for SSE
bc = cbind(lambda, lik) # combine lambda and lik
sorted_bc = bc[order(-lik),]
# values are sorted to identify the lambda value
# for the maximum log likelihood for obtaining minimum SSE
head(sorted_bc, n = 10)
# Take -1.15 for easy transformation of values
# back to their original scale.
# (Or can use lambda = b$x[b$y==max(b$y)])
lambda=b$x[b$y==max(b$y)]
boxy=(y^lambda-1)/lambda
boxfit=lm(boxy~x1+x2+x3)

# compare with BIC

```

```

BIC(boxfit)
BIC(lm);BIC(lm1);BIC(lm2);BIC(lm3);BIC(lm4);BIC(lm5);BIC(lm6);BIC(lm7)

#compare y~x1+x2 vs transformed y~ x1+x2
par(mfrow=c(1,2))
qqPlot(lm)
qqPlot(rstandard(boxfit))

#Residuals check
par(mfrow=c(1,1))
plot(lm(log(cycles)~as.factor(length)+as.factor(amplitude)+as.factor(load),data=yarn), which=1)

wear <- read.csv("wear.csv", header=T)
pairwise.t.test(wear$wear, wear$material, p.adj = "bonferroni")
#Bonferroni
(t_crit<-qt(0.05/12,6,lower.tail = F))

#Tukey
c<- c("A vs B = ", -8.27,"A vs C = ",-4.34,"A vs D = ",-6.37,"B vs C = ",-3.93,"B vs D = ",1.9,"C vs D = ")
print(c, quote=F)

```