STAT 222 Spring 2022 HW11

Matthew Zhao

```
## Warning: package 'knitr' was built under R version 4.0.5
```

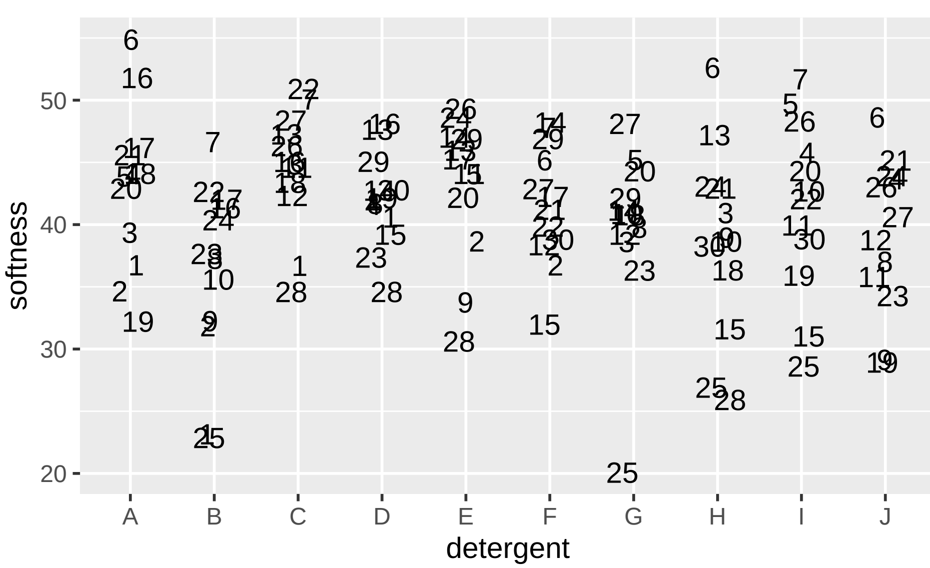
```
pr14.4 = read.table("http://users.stat.umn.edu/~gary/book/fcdae.data/pr14.4", header=T)
pr14.4$detergent = factor(pr14.4$treatment, labels=LETTERS[1:10])
```

Q1 — 6 points

Judge is the block factor and detergent is the treatment factor. There are g=10 treatments and there are b=30 blocks with k=4 size. There are $r = \frac{bk}{g} = \frac{30*4}{10} = 12$ replicates per treatment. There are $\lambda = \frac{r(k-1)}{g-1} = \frac{12(4-1)}{10-1} = 4$ times a pair of treatments show up together in a block. Since r and λ are integers, this design meets the first and second balanced conditions.

Q2 — 3 points

```
library(mosaic)
ggplot(pr14.4, aes(x=detergent, y=softness, label=judge)) +
  geom_text(position = position_jitter(width = .15))
```



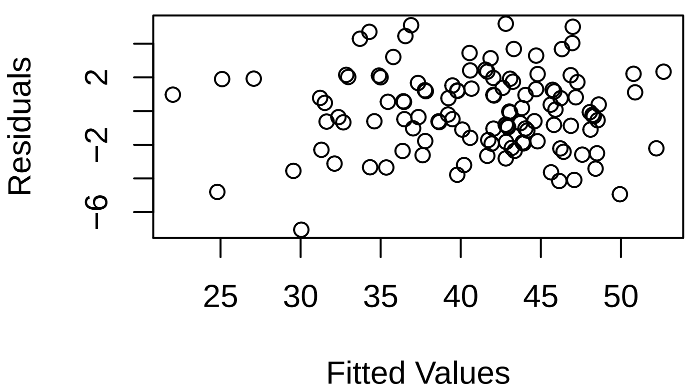
```
sort(mean(softness ~ detergent, data=pr14.4))
##        B        H        J        G        F        I        D        A
## 36.16667 38.66667 39.16667 39.66667 41.66667 41.66667 42.08333 43.08333
##        E        C
## 43.08333 44.50000
```

We should not use the mean rating because softness is subjective meaning that different judges would rate the same perceived softness (detergent treatment) differently. This leads to biased ratings by each judge as evidenced by judges who are biased downwards e.g. Judge 25 and biased upwards e.g. Judge 6.

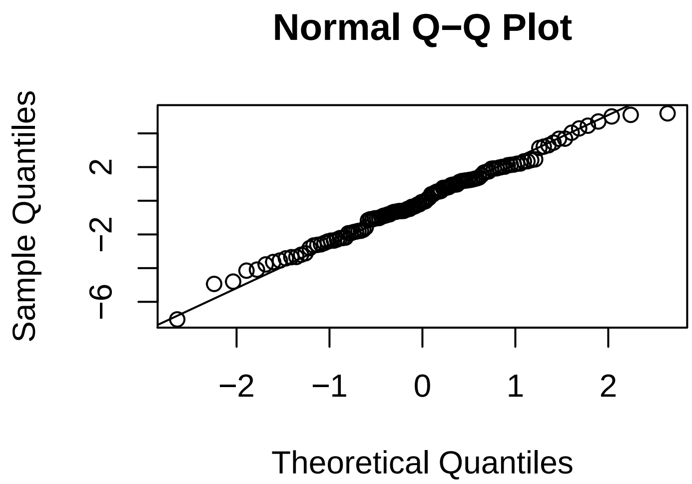
Q3 — 4 points

```
pr14.4$judge = as.factor(pr14.4$judge)
lm1 = lm(softness ~ judge + detergent, data=pr14.4)
```

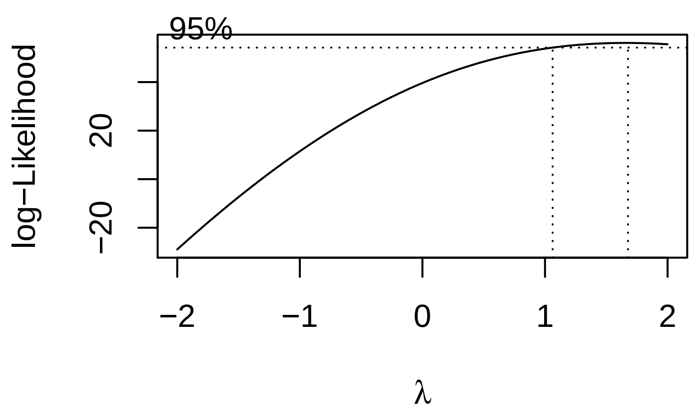
```
plot(lm1$fitted.values, lm1$residuals, xlab = "Fitted Values", ylab = "Residuals")
```




```
qqnorm(lm1$residuals)
qqline(lm1$residuals)
```




```
library(MASS)
boxcox(lm1)
```



There does not seem to be nonconstant variance in the errors as seen in the fitted values vs residuals graph. The normal qq plot also doesn't point to any potential non-normality in the data. Lastly, the boxcox plot reveals a CI that does not contain a specific value so we do not need to transform the data.

Q4 — 5 points

```
anova(lm(softness ~ judge + detergent, data=pr14.4))
## Analysis of Variance Table
##
## Response: softness
```

```
##           Df Sum Sq Mean Sq F value      Pr(>F)
## judge     29 3902.7 134.575 16.3697   < 2.2e-16
## detergent  9  454.3  50.483  6.1408 0.000001501
## Residuals 81  665.9   8.221
anova(lm(softness ~ detergent + judge, data=pr14.4))
## Analysis of Variance Table
##
## Response: softness
##           Df Sum Sq Mean Sq F value         Pr(>F)
## detergent  9  683.2  75.908  9.2335 0.000000001682
## judge     29 3673.8 126.684 15.4099      < 2.2e-16
## Residuals 81  665.9   8.221
```

The difference is because anova in R uses Type I Sum of Squares, meaning that for any term, it calculates sum of squares adjusted for terms that come before it in the model. We would use the first one where the judge block is before detergent in the model since we want to know the effect of treatment adjusted for block. Since p<0.05, we can conclude that detergent does have an effect on softness and that the detergents are not all equal.

Q5 — 2 points

```
pr14.4$dtgt.sum = pr14.4$detergent
pr14.4$judge.sum = pr14.4$judge
contrasts(pr14.4$dtgt.sum) = contr.sum(10)
contrasts(pr14.4$judge.sum) = contr.sum(30)
lm1.sum = lm(softness ~ judge.sum + dtgt.sum, data=pr14.4)
lm1.sum$coef
## (Intercept)  judge.sum1  judge.sum2  judge.sum3  judge.sum4  judge.sum5
##    40.97500    -7.23125    -4.60625    -0.56875     2.37500     3.99375
##  judge.sum6  judge.sum7  judge.sum8  judge.sum9 judge.sum10 judge.sum11
##    10.43750     8.56250    -0.81875    -6.05625    -0.25000    -0.39375
## judge.sum12 judge.sum13 judge.sum14 judge.sum15 judge.sum16 judge.sum17
##    -0.84375     4.45000     3.31250    -7.98750     4.76875     3.39375
## judge.sum18 judge.sum19 judge.sum20 judge.sum21 judge.sum22 judge.sum23
##    -0.56875    -6.87500     1.49375     3.68750     3.31250    -3.56875
## judge.sum24 judge.sum25 judge.sum26 judge.sum27 judge.sum28 judge.sum29
##     4.44375   -15.25000     4.85625     4.65625   -10.80000     3.81250
##   dtgt.sum1   dtgt.sum2   dtgt.sum3   dtgt.sum4   dtgt.sum5   dtgt.sum6
##     1.25000    -3.70000     2.67500     2.80000     1.45000    -1.47500
##   dtgt.sum7   dtgt.sum8   dtgt.sum9
##    -0.92500    -0.62500     1.35000
```

Using the table above, the alphas are given as $\alpha_1 = 1.25000, \alpha_2 = -3.70000, \alpha_3 = 2.67500, \alpha_4 = 2.80000, \alpha_5 = 1.45000, \alpha_6 = -1.47500, \alpha_7 = -0.92500, \alpha_8 = -0.62500, \alpha_9 = 1.35000, \alpha_10 = -\sum_{i=1}^{9} = -2.8$.

Q6 — 5 points

```r
mse <- 8.221
k <- 4
g <- 10
lambda <- 4
se <- sqrt(mse * (2*k/(g * lambda)))
qtukey(1-0.05,10,81)/sqrt(2) * se #hsd
## [1] 4.171851
```

```r
sort(lm1.sum$coef[(length(lm1.sum$coefficients)-8):length(lm1.sum$coefficients)])
## dtgt.sum2 dtgt.sum6 dtgt.sum7 dtgt.sum8 dtgt.sum1 dtgt.sum9 dtgt.sum5 dtgt.sum3
##    -3.700    -1.475    -0.925    -0.625     1.250     1.350     1.450     2.675
## dtgt.sum4
##     2.800
```

```
   B      J      F      G      H      A      I      E      C      D
-3.700  -2.8  -1.475 -0.925 -0.625  1.250  1.350  1.450  2.675  2.800 ----------------------
        ----------------------------------------
              ------------------------------------------------
                     ------------------------------------------------
```