STA207 Homework 1

Juanjuan Hu, Zhen Zhang

January 11, 2016

```r
library(xlsx)
```

```
## Loading required package: rJava
## Loading required package: xlsxjars
```

```r
kidney <- xlsx::read.xlsx2("kidney.xlsx", 1)
```

19.18

(a)

```r
transform_function <- function(x) {
  log10(x + 1)
}
kidney$Days <- transform_function(as.numeric(as.character(kidney$Days)))

mean_kidney <- with(kidney, tapply(Days, list(Duration = Duration, Weight = Weight), mean))
kidney$residual <- kidney$Days - rep(as.vector(t(mean_kidney)), each = 10)
kidney$residual
```

```
##  [1] -0.44348243  0.03363883 -0.14245243  0.15857756 -0.44348243
##  [6]  0.03363883 -0.44348243  0.33466882  0.40161561  0.51076008
## [11] -0.33284626 -0.11099751  0.09312248  0.30397584  0.39415247
## [16] -0.11099751 -0.20790752 -0.50893752 -0.03181626  0.51225178
## [21]  0.09742014 -0.06530715 -0.15245733 -0.32854859  0.30827351
## [26]  0.12374908 -0.20360985  0.38466185 -0.50463985  0.34045819
## [31] -0.39822712 -0.09719713 -0.09719713 -0.39822712  0.30074288
## [36]  0.07889413  0.50486286  0.30074288 -0.39822712  0.20383287
## [41]  0.19718720  0.02109594 -0.10384280 -0.58096405 -0.27993405
## [46] -0.27993405  0.02109594  0.26413399  0.32212594  0.41903595
## [51]  0.17499803  0.08784785  0.24754870 -0.26433466  0.03669533
## [56]  0.33772533 -0.16742465  0.13360535 -0.02129661 -0.56536466
```

(b)

```r
stripchart(split(kidney$residual, with(kidney, list(Duration = Duration, Weight = Weight))),
           method = "stack",  pch = 19)
abline(h = seq(2, 6) - 0.1)
```
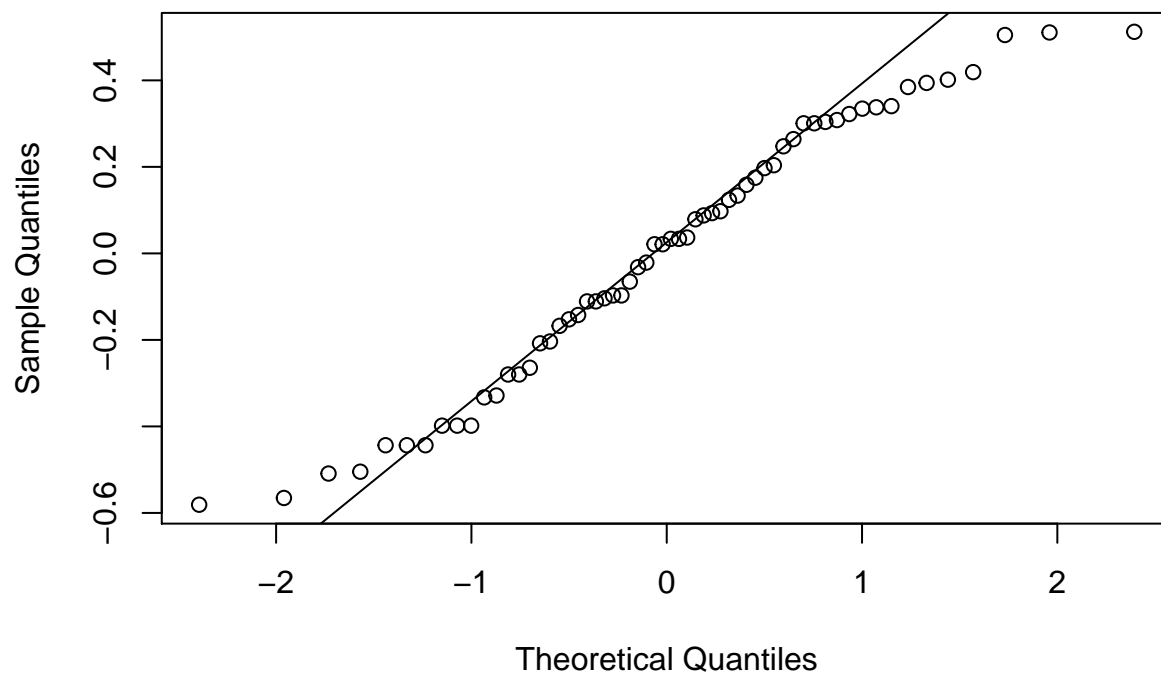
1
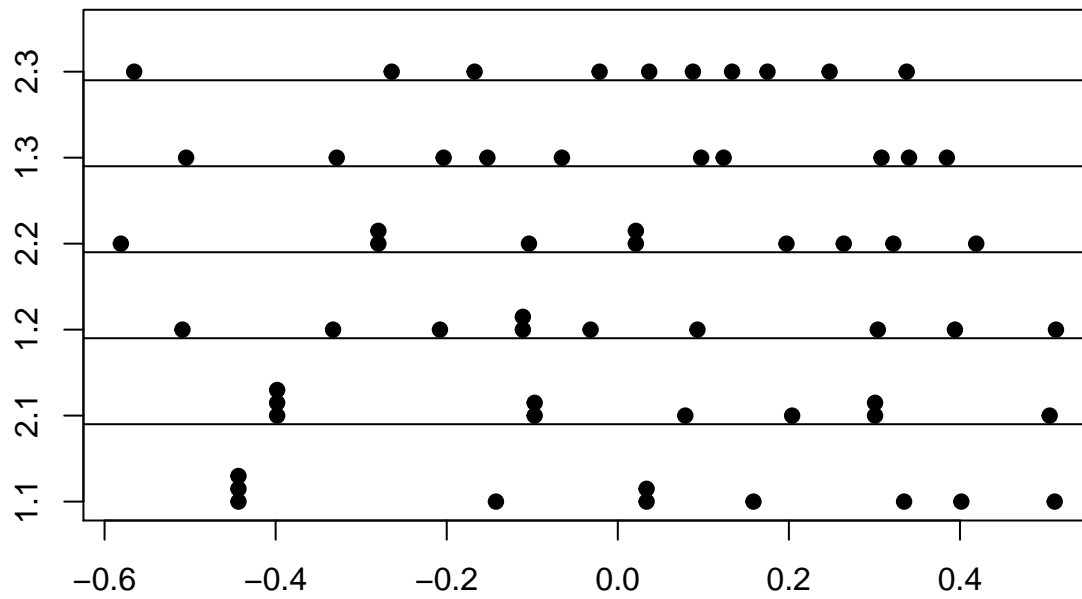
we can know that this ANOVA model has equal variance since the departure is almost average.

(c)

```
qqp = qqnorm(kidney$residual)
qqline(kidney$residual)
```

Normal Q–Q Plot

```r
cor(qqp$x, qqp$y)
```
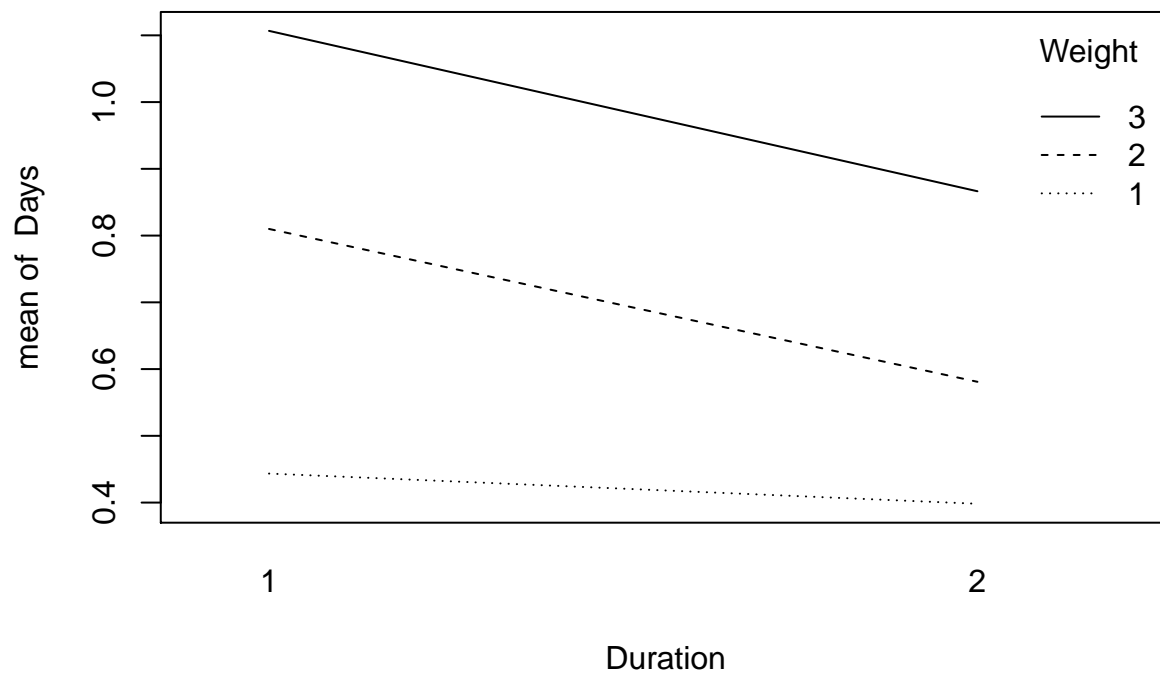
```
## [1] 0.9850992
```

The relationship coefficient is 0.985.
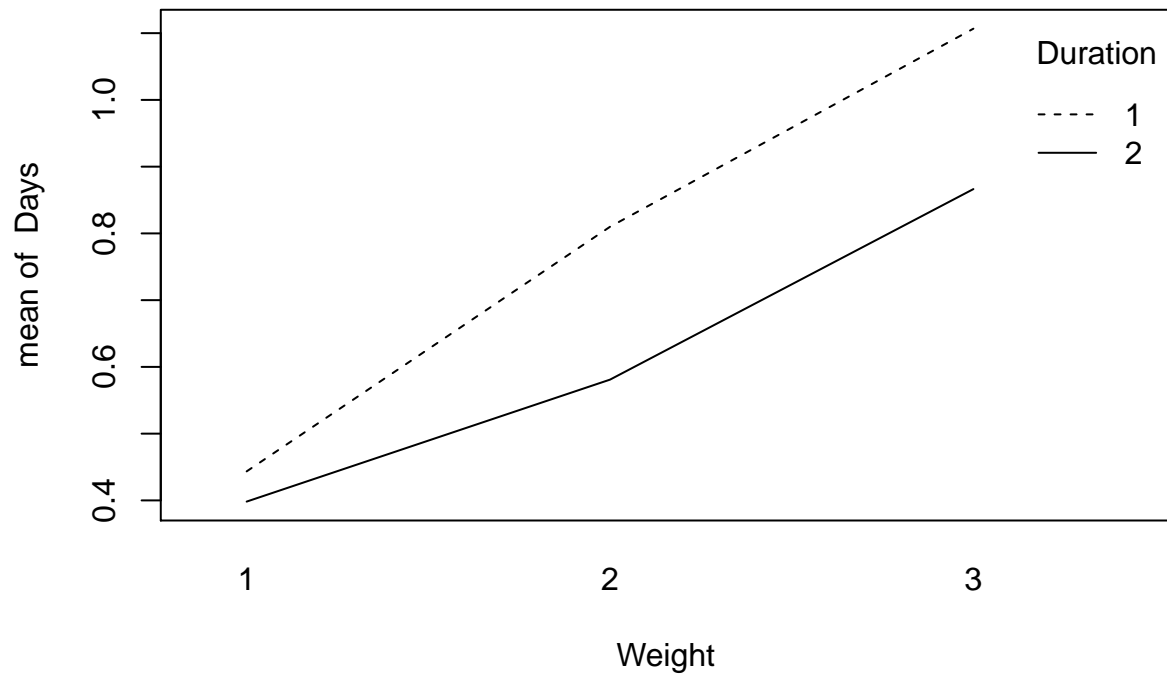
Not reasonable, it skewed to the left.

19.19

(a)

```r
with(kidney, interaction.plot(Duration, Weight, Days))
```



```r
with(kidney, interaction.plot(Weight, Duration, Days))
```

Factor Duation, Weight effect all exist.

(b)

```r
aov.out = aov(Days ~ Duration * Weight, data = kidney)
summary(aov.out)
```

```
##                 Df Sum Sq Mean Sq F value    Pr(>F)
## Duration         1  0.441  0.4413   4.358    0.0416 *
## Weight           2  3.201  1.6005  15.807 3.94e-06 ***
## Duration:Weight  2  0.120  0.0599   0.592    0.5567
## Residuals       54  5.468  0.1013
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Factor Weight account for the most variability, since it has the most SSE.

(c)

H_0: All $(\alpha\beta_{ij})$ equal zero.

H_a: Not all $(\alpha\beta_{ij})$ equal zero.

Test statistic: $F^* = MSAB/MSE = \frac{0.0599}{0.1013} = 0.5913$

Decision Rule: $F(0.95; 2, 54) = $ qf(0.95, 2, 54) $= 3.1682$. So reject H_0 if $F^* > F(0.95; 2, 54)$.

Conclusion: Since $0.5913 < 3.1682$, we cannot reject H_0. P-value is 0.5567.

(d)

For α_i:

H_0: All (α_i) equal zero.

H_a: Not all (α_i) equal zero.

Test statistic: $F^* = MSA/MSE = \frac{0.4413}{0.1013} = 4.3564$

Decision Rule: $F(0.95; 1, 54) = $ `qf(0.95, 1, 54)` $= 4.0195$. So reject H_0 if $F^* > F(0.95; 1, 54)$.

Conclusion: Since $4.3564 > 4.0195$, we cannot reject H_0. P-value is 0.0416.

For β_j:

H_0: All (β_j) equal zero.

H_a: Not all (β_j) equal zero.

Test statistic: $F^* = MSB/MSE = \frac{1.6005}{0.1013} = 15.807$

Decision Rule: $F(0.95; 2, 54) = $ `qf(0.95, 2, 54)` $= 3.1682$. So reject H_0 if $F^* > F(0.95; 2, 54)$.

Conclusion: Since $15.807 > 3.1682$, we cannot reject H_0. P-value is 0.0000.

(e)

The kimball inequlity bound:

$\alpha = 1 - (1 - 0.05) * (1 - 0.05) * (1 - 0.05) = 0.143$

(f)

Yes, it confirm the result I get in part (a).

19.34

(a)

$\hat{\mu_{22}} = 0.58096$, $s(\hat{\mu_{22}}) = \sqrt{\frac{MSE}{n}} = 0.10065$, $t(0.975; 54) = 2.0049$.

So the 95% interval is $(0.58096 \pm 2.0049 * 0.10065) = (0.37917, 0.78275)$

(b)

$\hat{D} = 0.86639 - 0.39823 = 0.46816$, $s(\hat{D}) = \sqrt{\frac{2MSE}{n}} = 0.14234$, $t(0.975; 54) = 2.0049$.

So the 95% interval is $(0.46816 \pm 2.0049 * 0.14234) = (0.18278, 0.75354)$

(c)

```r
model.tables(aov.out, type = 'means', se = T)
```

```
## Tables of means
## Grand mean
##
## 0.7009559
##
##   Duration
## Duration
##      1      2
## 0.7867 0.6152
##
##   Weight
## Weight
##      1      2      3
## 0.4209 0.6955 0.9865
##
##   Duration:Weight
##          Weight
## Duration 1      2      3
##        1 0.4435 0.8100 1.1067
##        2 0.3982 0.5810 0.8664
##
## Standard errors for differences of means
##          Duration  Weight Duration:Weight
##           0.08216 0.10062         0.14231
## replic.        30      20              10
```

```r
barplot(model.tables(aov.out, type = 'means', se = T)$tables$Duration)
```
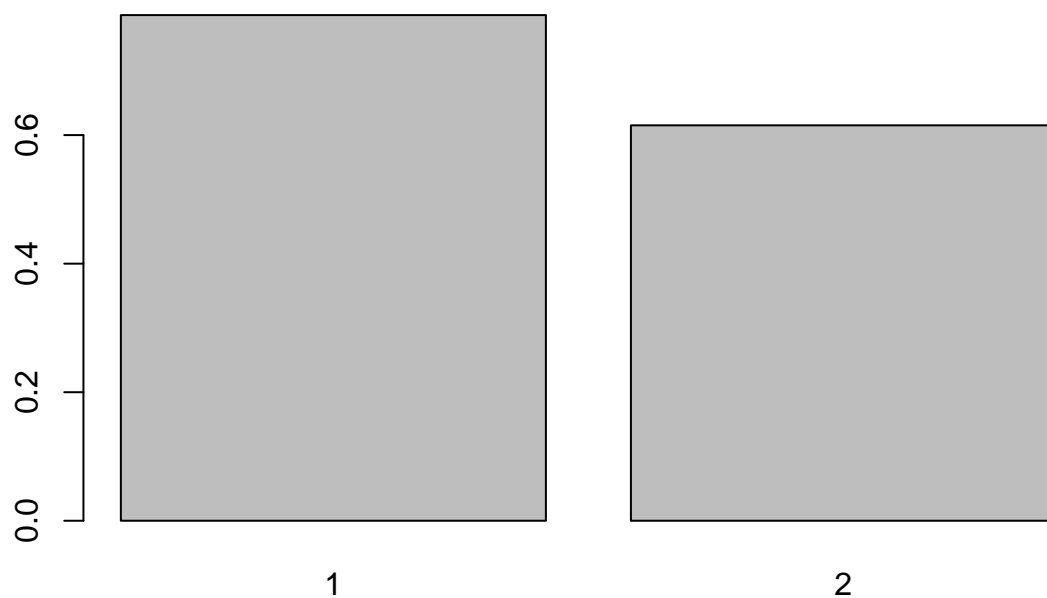
```
barplot(model.tables(aov.out, type = 'means', se = T)$tables$Weight)
```
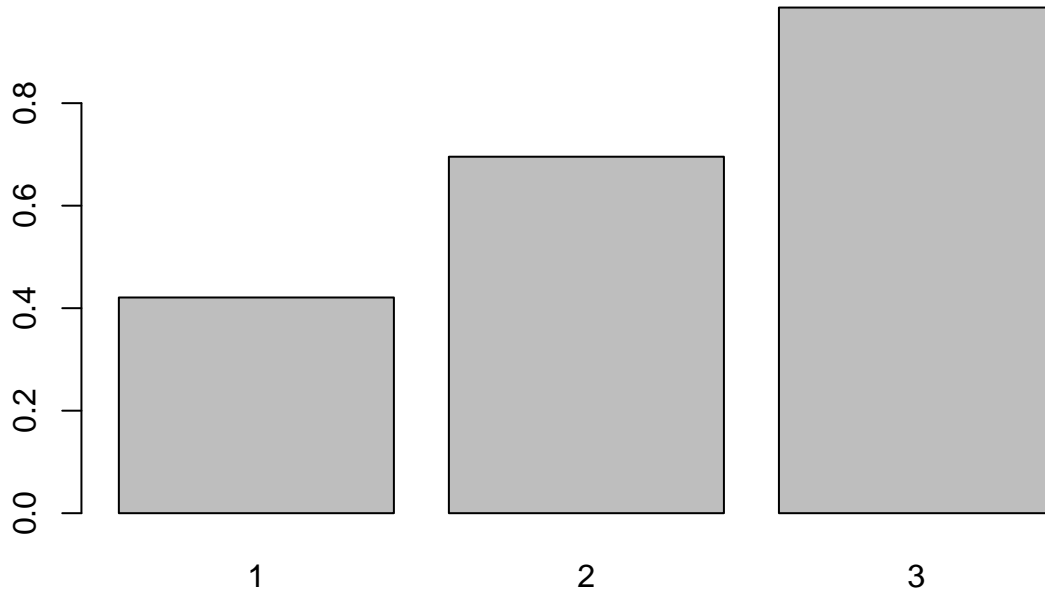

$\bar{Y}'_{1..} = 0.7867, \bar{Y}'_{2..} = 0.6152$

$\bar{Y}'_{.1.} = 0.4209, \bar{Y}'_{.2.} = 0.6955, \bar{Y}'_{.3.} = 0.9865,$

The factor main effects exist.

(d)

Bonferroni: $B = t(1 - \frac{0.05}{4}, 54) = t(0.9875, 54) = 2.306$

Tukey:

for α_i: $T = \frac{1}{\sqrt{2}} q(0.95; 2, 54) = 2.008$

for β_j: $T = \frac{1}{\sqrt{2}} q(0.95; 3, 54) = 2.411$

Scheffe:

$S^2 = 3 * F(0.9; 3, 54) = 6.56436$, so $S = 2.5621$.

So Bonferroni is best here.

(e)

$\hat{D}_1 = \bar{Y}'_{1..} - \bar{Y}'_{2..} = 0.1715$

$\hat{D}_2 = \bar{Y}'_{.1.} - \bar{Y}'_{.2.} = -0.2746$

$\hat{D}_3 = \bar{Y}'_{.1.} - \bar{Y}'_{.3.} = -0.5656$

$\hat{D}_4 = \bar{Y}'_{.2.} - \bar{Y}'_{.3.} = -0.291$

$s(\hat{D}_1) = \sqrt{\frac{2MSE}{30}} = 0.0822, s(\hat{D}_2) = s(\hat{D}_3) = s(\hat{D}_4) = \sqrt{\frac{2MSE}{20}} = 0.1006.$

$B = 2.306$

Then the intervals are:

D_1: $(0.1715 \pm 0.0822 * 2.306) = (0.01805, 0.36105)$

D_2: $(-0.2746 \pm 0.1006 * 2.306) = (-0.5066, -0.0426)$

D_3: $(-0.5656 \pm 0.1006 * 2.306) = (-0.7976, -0.3336)$

D_4: $(-0.291 \pm 0.1006 * 2.306) = (-0.5230, -0.0590)$

(f)

$\hat{D} = 0.3\mu_{.1} + 0.4\mu_{.2} + 0.3\mu_{.3} = 0.70041$, $s(\hat{D}) = \sqrt{\frac{MSE}{20} * (0.3^2 + 0.4^2 + 0.3^2)} = 0.04150$, $t(0.975, 54) = 2.0049$.

So the interval should be $(0.70041 \pm 2.0049 * 0.04150) = (0.61721, 0.78361)$.

To convert them to the original unit, we get $(3.142, 5.076)$, which means less than 7.

19.41

We can define a function here:

```
find_sample_size <- function(n) {
  B = qt(0.9875, 5*(n - 1))
  sigma = 0.32 * (1/n) ^ 0.5
  return (B * sigma)
}
```

Now run this function with some values:

```
find_sample_size(15)
```

```
## [1] 0.1892609
```

```
find_sample_size(14)
```

```
## [1] 0.1962349
```

```
find_sample_size(13)
```

```
## [1] 0.204045
```

So we know $n = 14$ is the smallest sample size needed to maintain a precision of ± 0.20.