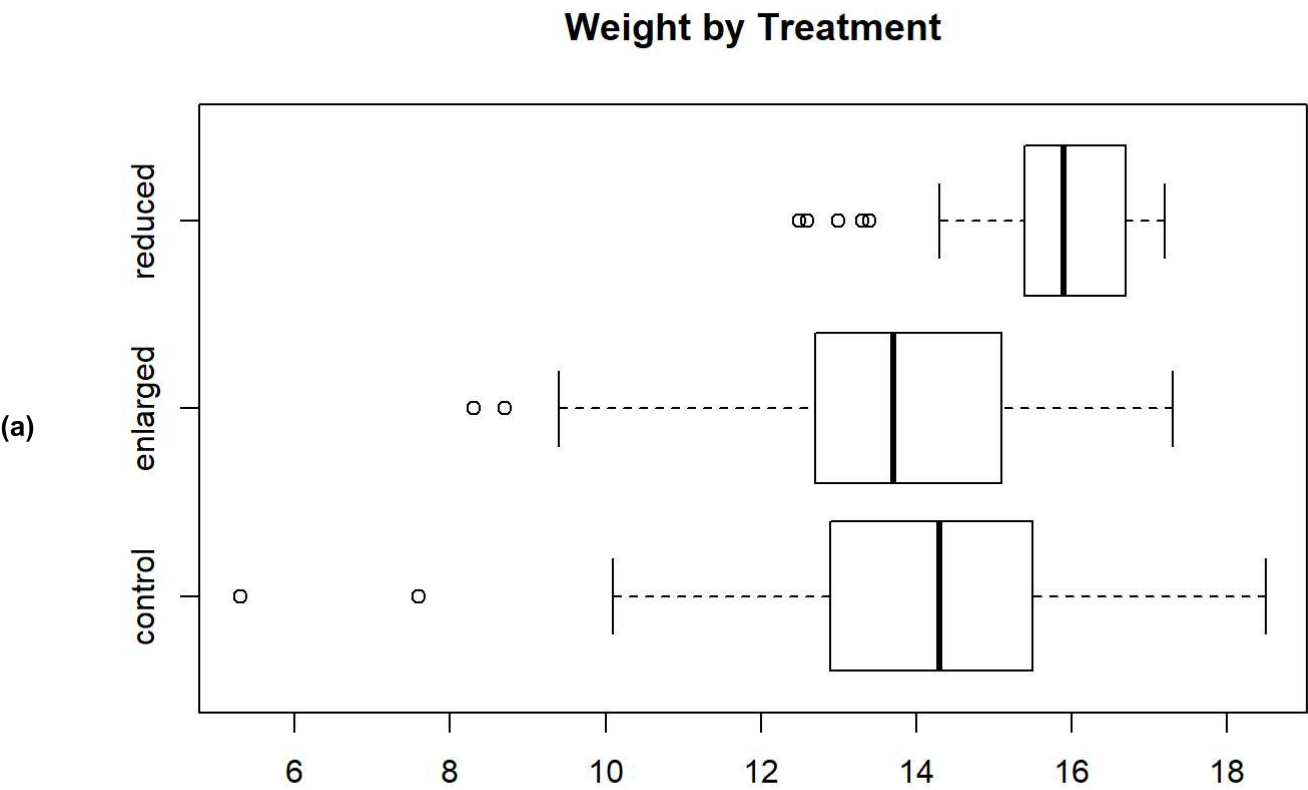


# STA 106 - HW1

Yutian Yang

## Problem 1



```
## Treatment Weight
## 1 control 13.92444
## 2 enlarged 13.51556
## 3 reduced 15.56923
```

```
## Treatment Weight
## 1 control 2.419631
## 2 enlarged 2.103996
## 3 reduced 1.459252
```

```
## Treatment Weight
## 1 control 45
## 2 enlarged 45
## 3 reduced 26
```

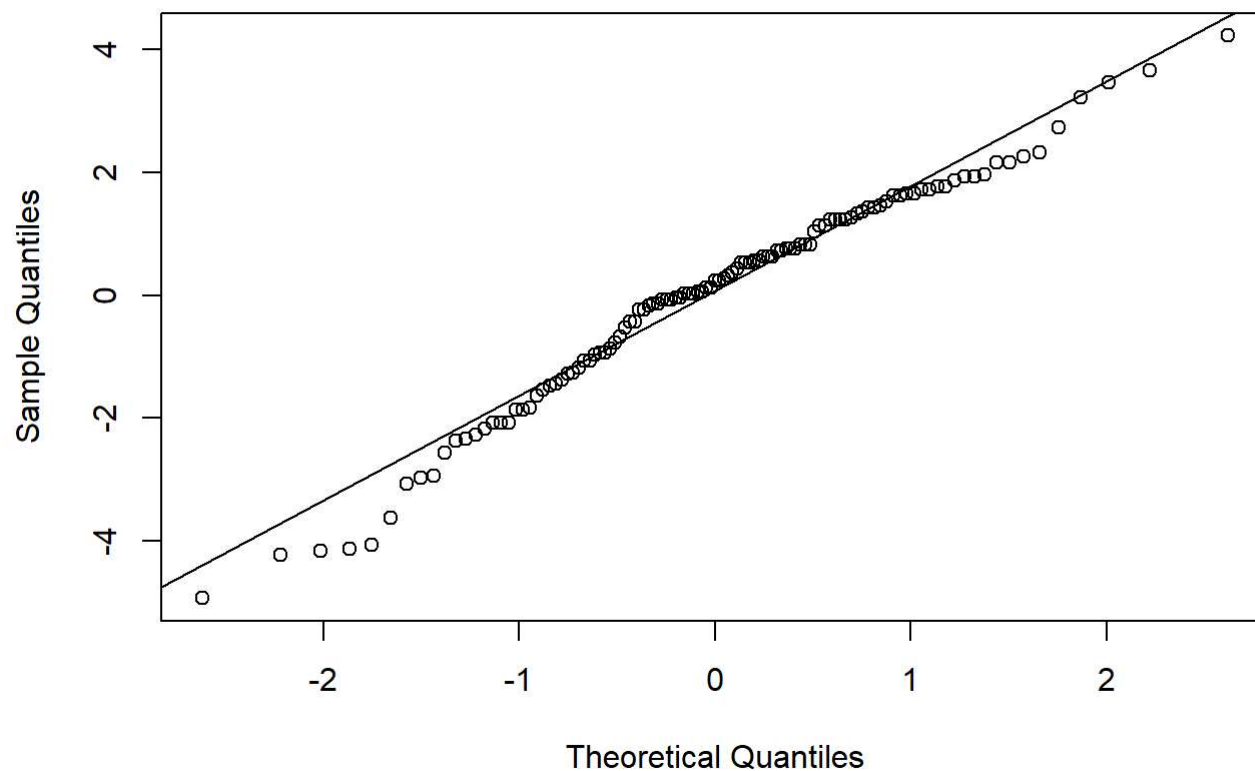
```
## [1] 14.13448
```

```
## [1] 2.242596
```

```
## 44 45 81
```

```
## 44 45 81
```

### Normal Q-Q Plot



```
##
## Shapiro-Wilk normality test
##
## data: ei
## W = 0.97568, p-value = 0.0371
```

```
## Loading required package: carData
```

```
## [1] 0.1785612
```

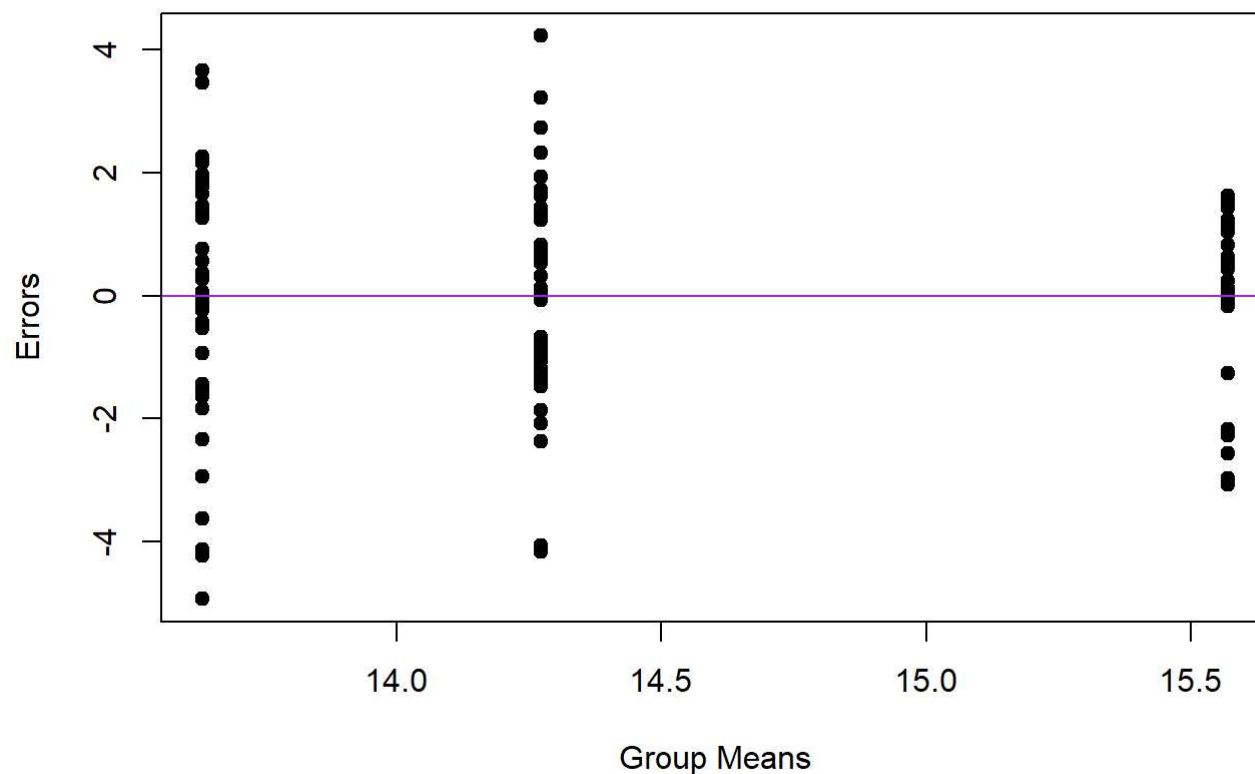
```
## [1] 2.973358
```

```
## [1] 3.097758
```

```
## [1] 2.998884
```

```
## Loading required package: tcltk
```

### Errors vs. Group Means



```
##
## 99% Tukey-Kramer confidence intervals
##
##           Diff      Lower      Upper  Decision Adj. p-value
## mucontrol-muenlarged  0.40889 -0.91706  1.73484    FTR H0    0.630776
## mucontrol-mureduced -1.64479 -3.19416 -0.09542  Reject H0    0.005765
## muenlarged-mureduced -2.05368 -3.60305 -0.5043   Reject H0    0.000412
```

```
## [1] 0.855733
```

## R Appendix

```

sparrow <- read.csv("C:/Users/charl/Downloads/sparrow.csv")
boxplot(Weight ~ Treatment, data = sparrow, main = "Weight by Treatment", horizontal = TRUE)
aggregate(Weight ~ Treatment, data = sparrow, mean)
aggregate(Weight ~ Treatment, data = sparrow, sd)
aggregate(Weight ~ Treatment, data = sparrow, length)
mean(sparrow$Weight)
sd(sparrow$Weight)
give.me.CI = function(ybar, ni, ci, MSE, multiplier) {
  if(sum(ci) != 0 & sum(ci != 0) != 1) {
    return("Error - you did not input a valid contrast")
  } else if(length(ci) != length(ni)) {
    return("Error - not enough contrasts given")
  }
  else {
    estimate = sum(ybar*ci)
    SE = sqrt(MSE*sum(ci^2/ni))
    CI = estimate + c(-1, 1)*multiplier*SE
    result = c(estimate, CI)
    names(result) = c("Estimate", "Lower Bound", "Upper Bound")
    return(result)
  }
}

group.means = by(sparrow$Weight, sparrow$Treatment, mean)
group.nis = by(sparrow$Weight, sparrow$Treatment, length)
the.model = lm(Weight ~ Treatment, data = sparrow)
anova.table = anova(the.model)
MSE = anova.table[2, 3]
nt = sum(group.nis)
a = length(group.means)
alpha = 0.05
the.model = lm(Weight ~ Treatment, data = sparrow)
sparrow$ei = the.model$residuals
nt = nrow(sparrow) #Calculates the total sample size
a = length(unique(sparrow$Treatment)) #Calculates the value of a
SSE = sum(sparrow$ei^2) #Sums and squares the errors (finds SSE)
MSE = SSE/(nt-a) #Finds MSE
eij.star = the.model$residuals/sqrt(MSE)
alpha = 0.01
t.cutoff= qt(1-alpha, nt-a)
CO.eij = which(abs(eij.star) > t.cutoff)
CO.eij

outliers = CO.eij
new.data = sparrow[-outliers,]
new.model = lm(Weight ~ Treatment, data = new.data)
qqnorm(new.model$residuals)
qqline(new.model$residuals)
ei = new.model$residuals
the.SWtest = shapiro.test(ei)
the.SWtest
plot(new.model$fitted.values, new.model$residuals, main = "Errors vs. Group Means", xlab = "Group Means", ylab = "Errors", pch = 19)
abline(h = 0, col = "purple")
library(car)

```

```

the.BFtest = leveneTest(ei~ Treatment, data=sparrow, center=median)
p.val = the.BFtest[[3]][1]
p.val
Tuk = qt(1-alpha, a, nt-a)/sqrt(2)
Tuk
Scheffe = sqrt((a-1)*qf(1-alpha, a-1, nt-a))
Scheffe
g=3
Bonferroni = qt(1-alpha/(2*g), nt-a)
Bonferroni
library(asbio)
tukeyCI(sparrow$Weight, sparrow$Treatment, conf.level = 1-alpha) #Tukey
give.me.power = function(ybar, ni, MSE, alpha) {
  a = length(ybar) # Finds a
  nt = sum(ni) #Finds the overall sample size
  overall.mean = sum(ni*ybar)/nt # Finds the overall mean
  phi = (1/sqrt(MSE))*sqrt( sum(ni*(ybar - overall.mean)^2)/a) #Finds the books value of phi
  phi.star = a *phi^2 #Finds the value of phi we will use for R
  Fc = qf(1-alpha, a-1, nt-a) #The critical value of F, use in R's function
  power = 1 - pf(Fc, a-1, nt-a, phi.star) # The power, calculated using a non-central F
  return(power)
}
the.power = give.me.power(group.means, group.nis, MSE, 0.01)
the.power

```