

# Hw5 Stat139

*Callin Switzer*

*October 5, 2014*

**1a**

On other paper

**1b**

On other paper

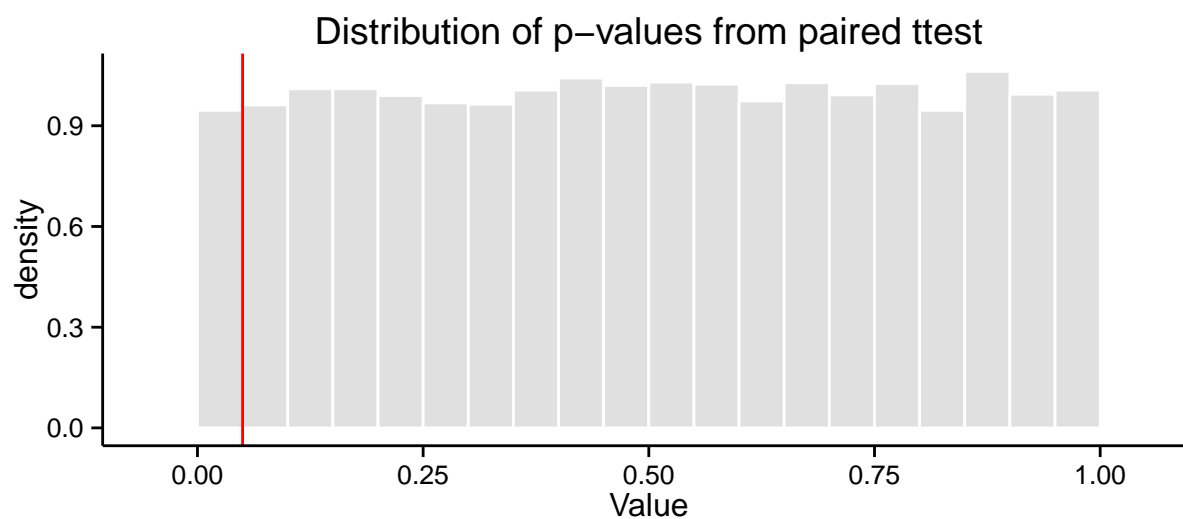
**1c**

On other paper

**1d**

On other paper

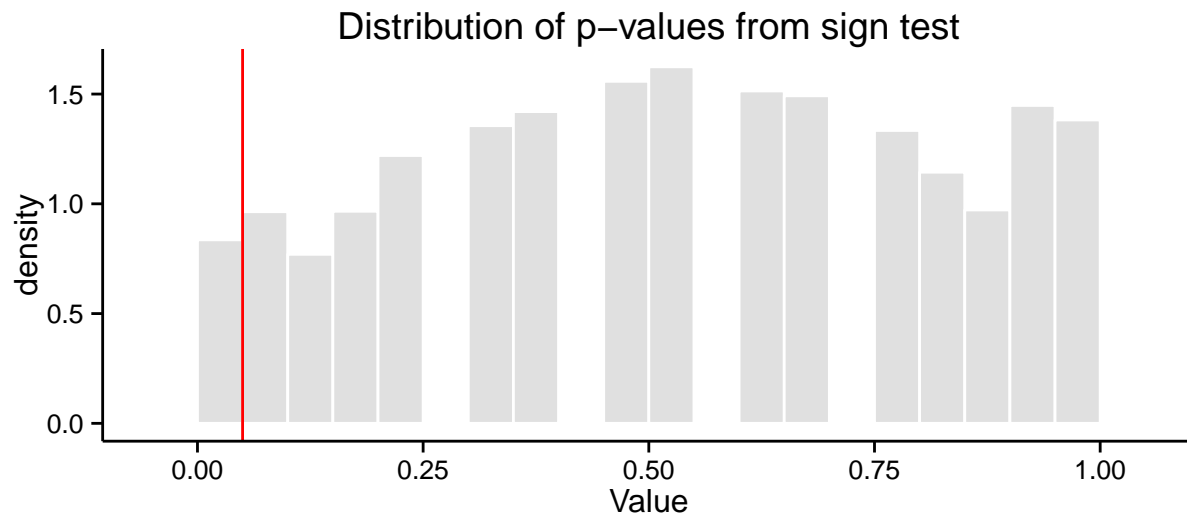
**2a**



The proportion of pvalues that are less than 0.05 is 0.0473, which is very close to the expected proportion.

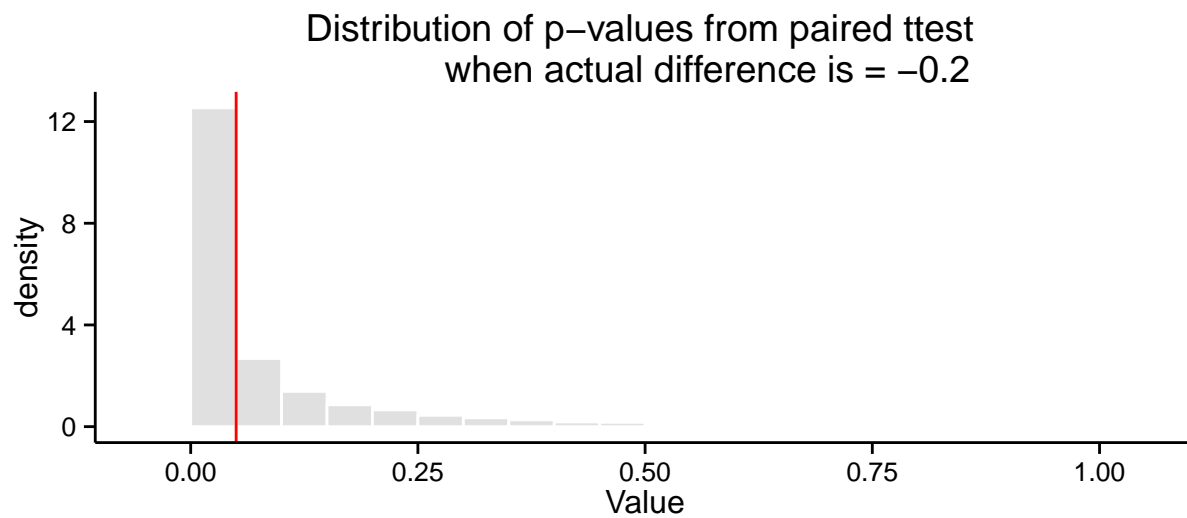
2b

```
## Loading required package: e1071
## Loading required package: lattice
##
## Attaching package: 'BSDA'
##
## The following object is masked from 'package:datasets':
##
##   Orange
```

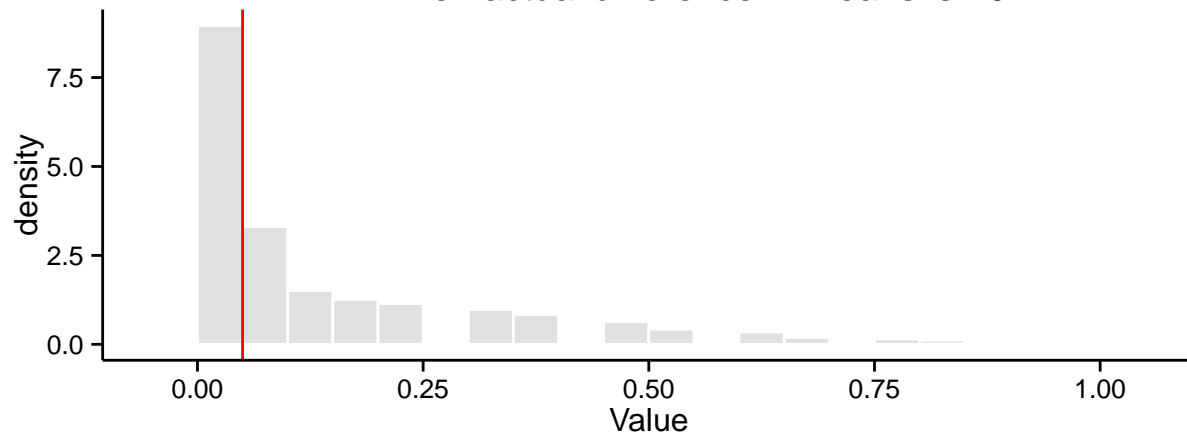


The proportion of left-sided p-values that are less than 0.05 is 0.0418, which is very close to the nominal proportion (though it is slightly lower)

2c



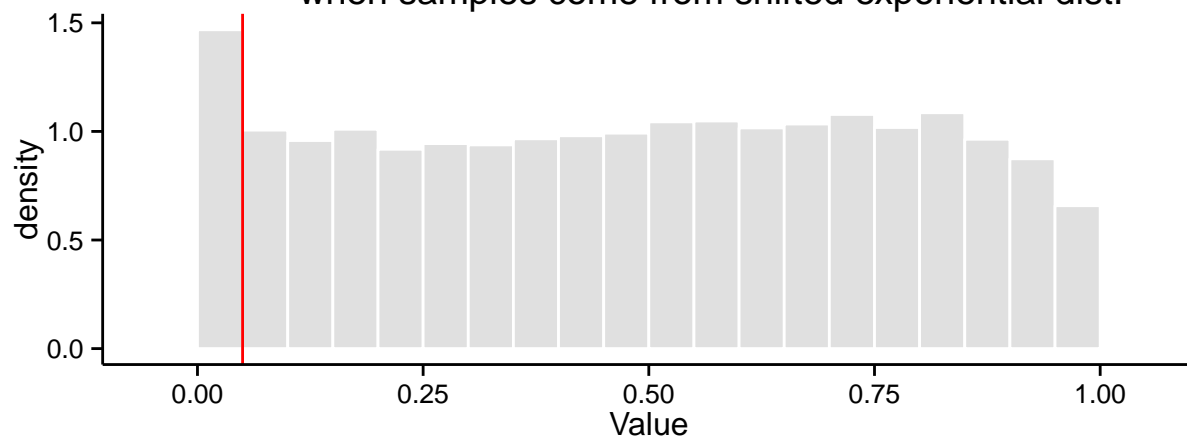
Distribution of p-values from sign test  
when actual difference in means is  $-0.2$



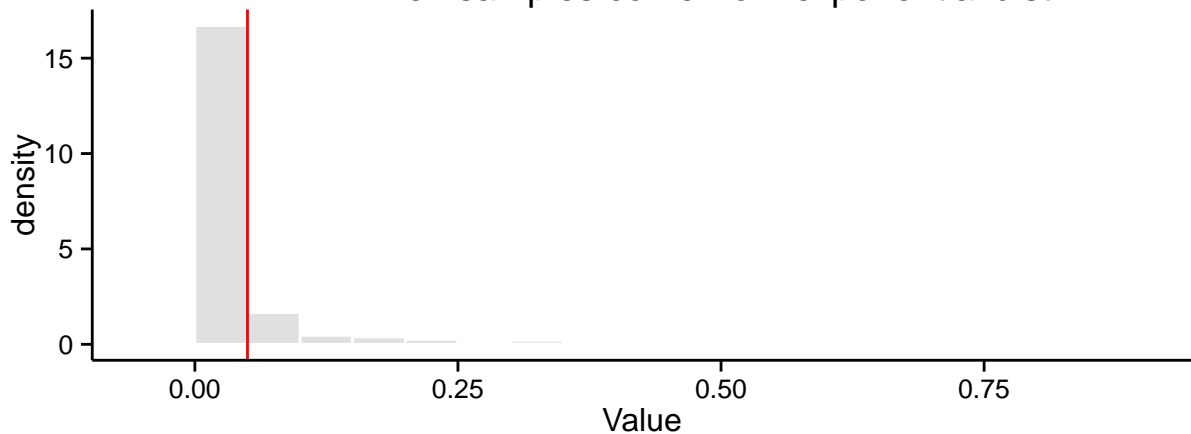
The proportion of left sided p-values that are less than 0.05 for paired t-test is 0.627, and the proportion of left sided p-values that are less than 0.05 for the sign test is 0.4482. When the t assumptions hold, the paired t-test is much more powerful than the sign test.

2d

Distribution of p-values from paired ttest  
when samples come from shifted exponential dist.



## Distribution of p-values from sign test when samples come from exponential dist



The proportion of left sided p-values that are less than 0.05 for paired t-test is 0.0734, and the proportion of left sided p-values that are less than 0.05 for the sign test is 0.8357.

## 2di

The paired t-test is not very robust to the assumption of normality. You can see from the histogram above that the alpha level will be higher than the nominal alpha level, even when the means are the same.

## 2dii

The sign test rejection rate is different from 0.05 when the differences follow an exponential distribution. This is due to the fact that the sign test is not testing for the mean = 0. Since the distribution of differences are continuous r.v.'s, the sign test is testing whether or not the median(differences) = 0. In the case of our shifted exponential distribution, the mean is 0, but the median is not zero. In this case, our rejection rate is not alpha, but it is power.

## 3 (Ch.4 #21)

```
#install.packages("coin")
library(coin)
nums <- c(20.1, 22.9, 18.8, 20.9, 20.9, 22.7, 21.4, 20,
          38.5, 25.8, 22.0, 23, 37.6, 30, 24.5)
gp <- c(rep("NT", 8), rep("T", 7))
obsRank <- rank(nums, ties.method = "average")
met <- data.frame(gp, nums, obsRank)
mt <- sum(obsRank[9:15])
average <- mean(obsRank)
sd <- sd(obsRank)
n = 7 # number of observations in smallest group
MEANT <- n * average
SDT <- sd*sqrt((8*7)/(8+7))
z <- (mt - MEANT)/SDT
```

```

p <- pnorm(-abs(z))
a <- wilcox_test(nums~gp, data = met, distribution = "exact",
                 conf.level = 0.05, alternative = "less")
b <- wilcox.test(nums~gp, alternative = "less", data = met, correct = F, exact = T)
b; a

```

```

##
## Wilcoxon rank sum test
##
## data:  nums by gp
## W = 2, p-value = 0.001299
## alternative hypothesis: true location shift is less than 0

##
## Exact Wilcoxon Mann-Whitney Rank Sum Test
##
## data:  nums by gp (NT, T)
## Z = -3.012, p-value = 0.0006216
## alternative hypothesis: true mu is less than 0

```

I used the code from the Amherst handout to calculate the Test statistic ( $T = 82$ ), Z-statistic ( $Z = 3.0116$ ), and p-value ( $p = 0.0013$ ). I also used the `wilcox_test()` (package: `coin`) function to check my Z-statistic (see below). The p-value are very close to my calculated p-value for `wilcox.test`. However the p-value from `wilcox__test` is lower. Neither of the pre-made methods in R give the exact same p-value that I calculated. Using, `wilcox.test()`, I didn't apply a continuity correction. Using the other form of the test, continuity correction is not applied either – I can tell, b/c the Z-value is the same.

### 3 (Ch.4 #22)

```

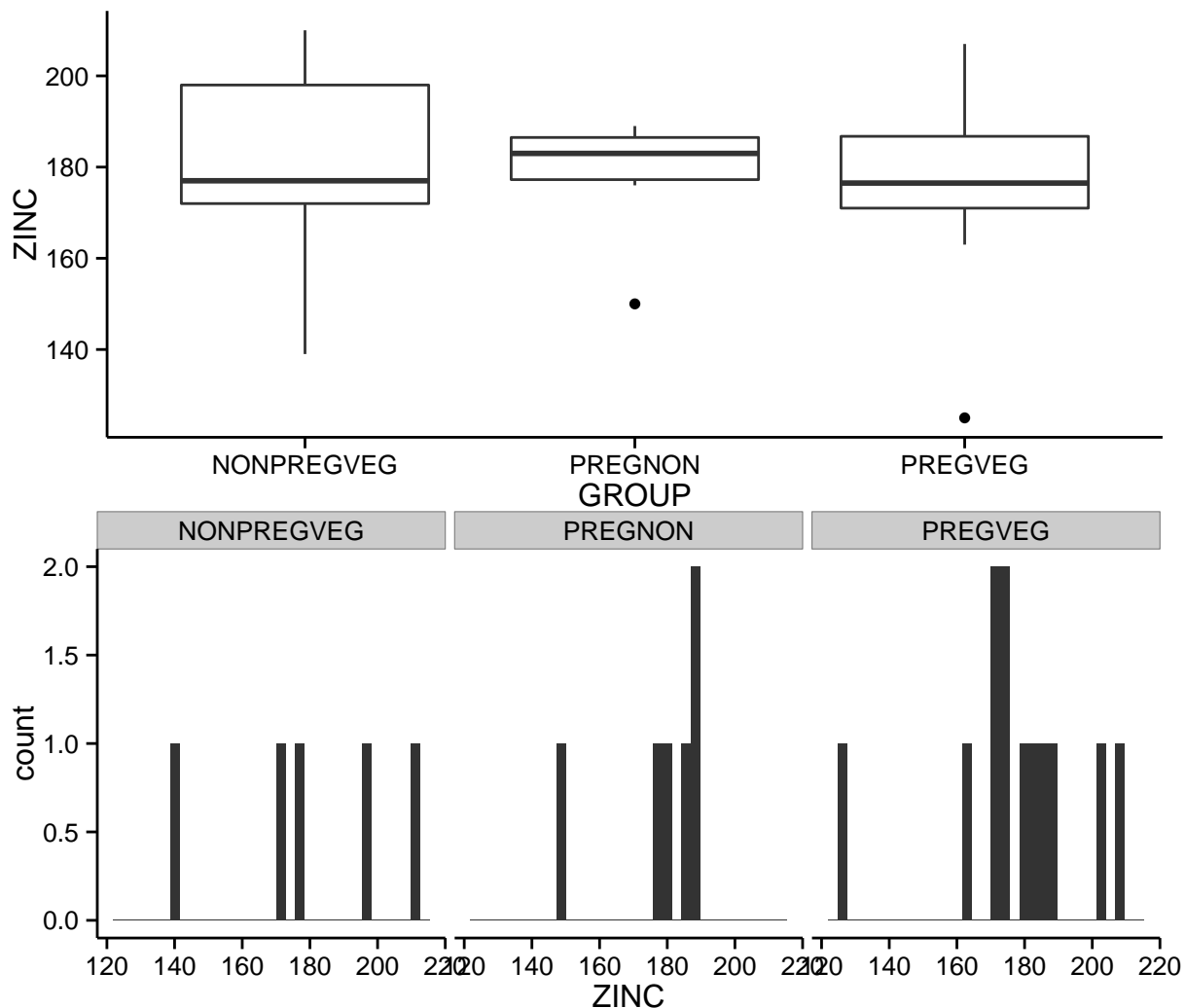
##
## Exact Wilcoxon Mann-Whitney Rank Sum Test
##
## data:  nums by gp (NT, T)
## Z = -3.012, p-value = 0.001243
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  -16.7  -1.9
## sample estimates:
## difference in location
##                -5.3

```

The way to calculate the confidence interval (95%) is to try subtracting multiple hypothesized values for delta (that is, subtracting delta from one of your groups). You keep plugging in values for delta until you find a place where the rank-sum test is not significant. This is the first side of the confidence interval. When you find the delta that makes the test become significant again, that is the other end of your confidence interval. I did this method and found that delta values between about  $[-17, -2]$  made the test non significant. I compared that with the output from `wilcox__test()` and it agreed – see output above.

From the `wilcox__test()`,  $[-16.7, -1.9]$  is the 95% confidence interval for median(trauma patients) - median(non trauma patients) in metabolic expenditures (kcal/kg/day). This is very close to the confidence interval I calculated via trial and error.

#### 4 (Ch.5, #24)



4

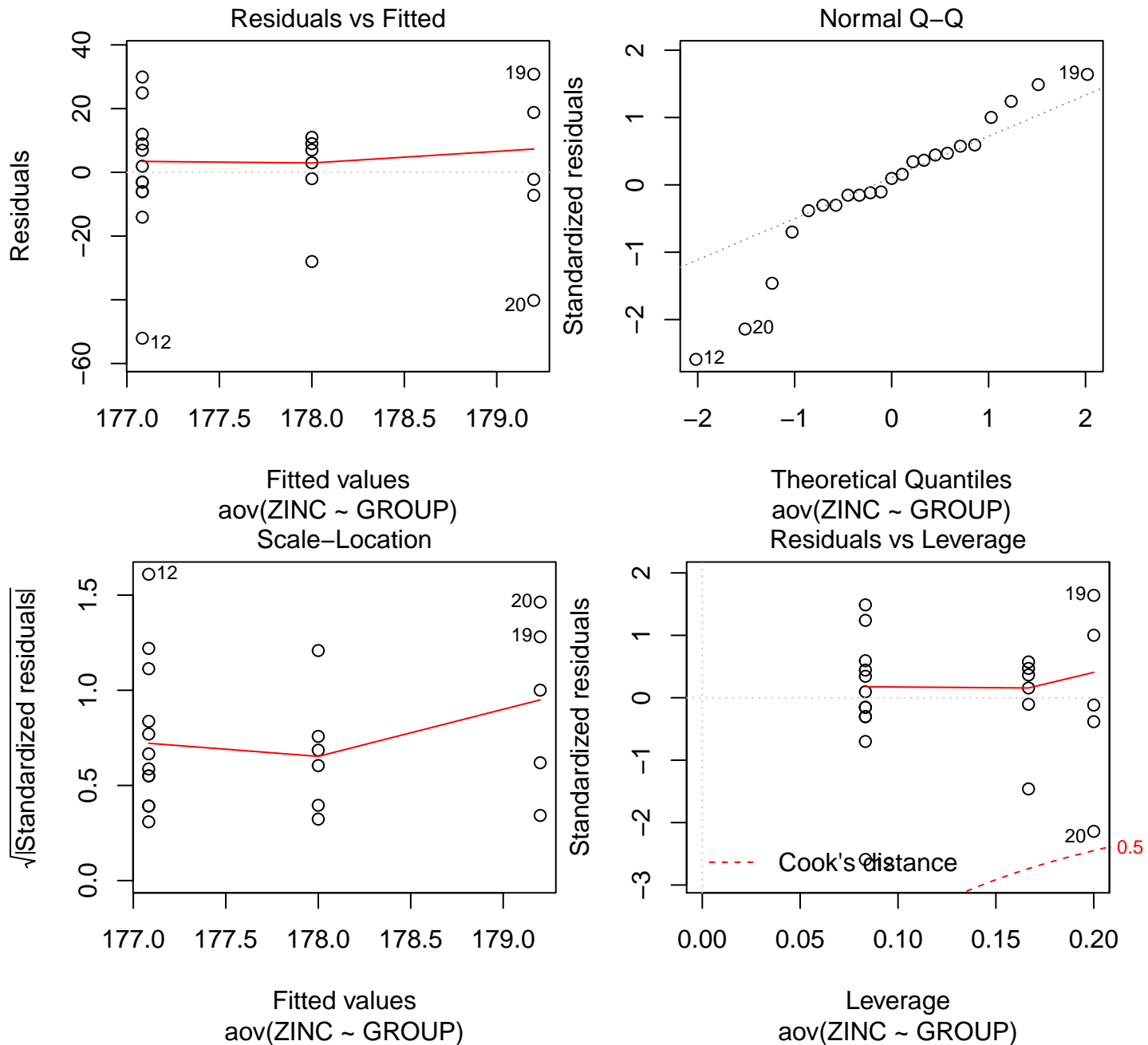
T-test Assumptions: Independence between units, \* Independent within populations, \* Independent between populations, Homogeneity of units within each population, \* equal means and variances within each population, \* equal variances between populations, Populations are normally distributed, Random sampling from populations

Comments: It's difficult to tell if the populations are normally distributed, because the sample sizes are small. However, I don't see any strong evidence that the populations are not normal. There is independence between populations and within populations. The populations also seem homogeneous. The samples may not be randomly selected from the population.

Scope of inference: Since the samples may not have been randomly selected from the population, inference to some population is speculative. We can say that the women in this study didn't seem to show evidence that pregnant vegetarians have different zinc levels than pregnant nonvegetarians.

Assumptions of ANOVA: 1. Normality of populations 2. Equal population variances for all groups 3. Independence within each group 4. Independence between groups 5. Homogeneity within each group 6.

Random sampling



Comments: Once again, the most suspicious assumption is that there was random sampling. Inference to the population is speculative. Also, looking at the Q-Q plot suggests that the populations might not be normally distributed. I looked at a boxplot and histograms of the data to help myself evaluate the assumptions (see above).

## Code

```
## #install.packages("BSDA")
## require(BSDA)
## zt <- function(o){
##     foo <- rnorm(100)
##     pv <- z.test(foo, alternative = "less", mu = 0, sigma.x = 1)$p.value
##     if(pv < 0.05) return(mean(foo))
##     else return(NA)
## }
##
##
## fg <- replicate(1000000, zt())
## minDiff <- max(fg, na.rm = T)
## ```
##
## ##### 1a #####
## On other paper
## ```{r}
## powPow <- function(o){
##     #foo <- rnorm(100)
##     bar <- rnorm(100, mean = 1)
##     #z.test(foo, bar, alternative = "less", sigma.x = 1, sigma.y = 1)
##     return(mean(bar))
## }
##
## bb <- replicate(10000, mean(rnorm(100, mean= -1)))
## hist(bb)
##
## length(bb[bb < minDiff])/length(bb)
##
## ```
##
## ## 1c
## On other paper
##
## ```{r}
## qnorm(p = 0.05)
## pnorm(8.355)
##
## ```
##
## ```{r, echo = F, fig.height=3}
## ##### function
## histT <- function(diffs = diffs, title = "Distribution"){
##     diffs <- data.frame(diffs = diffs)
##     #colnames(diffs) <- "diffs"
##     ggplot(diffs, aes(x = diffs)) +
##     geom_histogram(aes(y = ..density..), color = "white",
##                   alpha = 0.2, fill = "grey40", binwidth = 0.05)+
##     ## density line
##     #geom_line(aes(y = ..density.., lty = "Density"), stat = 'density')+
##     #geom_line(aes(y = ..density.., lty = "Normal"), stat = 'normal')+
## }
```



```

##      ## normal approx
##      #stat_function(aes(lty = "Normal"), fun=dnorm,
##                    #args=list(mean=mean(diffs$diffs), sd=sd(diffs$diffs)))+
##
##      labs(x = "Value", title = title) +
##      theme_bw() +
##      theme(legend.background = element_rect(colour = "black"),
##            plot.background = element_blank()
##            ,panel.grid.major = element_blank()
##            ,panel.grid.minor = element_blank()
##            ,panel.border = element_blank()
##            ,axis.line = element_line(color = 'black'))
##      ## Names for the legend
##      #scale_linetype(name = "Line")+
##      #scale_fill_manual(name = "Histogram", values = c("black"))
## }
##
## library(ggplot2)
##
##
## ##### simulations
## set.seed(100614)
## nsim <- 10000
## pair <- function(distr = rnorm(100), o){
##   t.test(distr, alternative = "less")$p.value
## }
##
## pvalues <- replicate(nsim, pair(rnorm(100)))
##
## vv <- histT(diffs = pvalues, title = "Distribution of p-values from paired ttest")
## vv + geom_vline(xintercept = 0.05, color = "red")
## prop <- sum(pvalues < 0.05)/ nsim
##
## ```
##
## ## 2b
## ```{r, echo = F}
## ### modify SIGN.test to output p.values
## my.SIGN.test <- function (x, y = NULL, md = 0, alternative = "two.sided", conf.level = 0.95)
## {
##   choices <- c("two.sided", "greater", "less")
##   alt <- pmatch(alternative, choices)
##   alternative <- choices[alt]
##   if (length(alternative) > 1 || is.na(alternative))
##     stop("alternative must be one \"greater\", \"less\", \"two.sided\"")
##   if (!missing(md))
##     if (length(md) != 1 || is.na(md))
##       stop("median must be a single number")
##   if (!missing(conf.level))
##     if (length(conf.level) != 1 || is.na(conf.level) || conf.level <
##         0 || conf.level > 1)
##       stop("conf.level must be a number between 0 and 1")

```

```

##   if (is.null(y)) {
##     dname <- paste(deparse(substitute(x)))
##     x <- sort(x)
##     diff <- (x - md)
##     n <- length(x)
##     nt <- length(x) - sum(diff == 0)
##     s <- sum(diff > 0)
##     estimate <- median(x)
##     method <- c("One-sample Sign-Test")
##     names(estimate) <- c("median of x")
##     names(md) <- "median"
##     names(s) <- "s"
##     CIS <- "Conf Intervals"
##     if (alternative == "less") {
##       pval <- sum(dbinom(0:s, nt, 0.5))
##       loc <- c(0:n)
##       prov <- (dbinom(loc, n, 0.5))
##       k <- loc[cumsum(prov) > (1 - conf.level)][1]
##       if (k < 1) {
##         conf.level <- (1 - (sum(dbinom(k, n, 0.5))))
##         xl <- -Inf
##         xu <- x[n]
##         ici <- c(xl, xu)
##       }
##       else {
##         ci1 <- c(-Inf, x[n - k + 1])
##         acl1 <- (1 - (sum(dbinom(0:k - 1, n, 0.5))))
##         ci2 <- c(-Inf, x[n - k])
##         acl2 <- (1 - (sum(dbinom(0:k, n, 0.5))))
##         xl <- -Inf
##         xu <- (((x[n - k + 1] - x[n - k]) * (conf.level -
##           acl2))/(acl1 - acl2)) + x[n - k]
##         ici <- c(xl, xu)
##       }
##     }
##     else if (alternative == "greater") {
##       pval <- (1 - sum(dbinom(0:s - 1, nt, 0.5)))
##       loc <- c(0:n)
##       prov <- (dbinom(loc, n, 0.5))
##       k <- loc[cumsum(prov) > (1 - conf.level)][1]
##       if (k < 1) {
##         conf.level <- (1 - (sum(dbinom(k, n, 0.5))))
##         xl <- x[1]
##         xu <- Inf
##         ici <- c(xl, xu)
##       }
##       else {
##         ci1 <- c(x[k], Inf)
##         acl1 <- (1 - (sum(dbinom(0:k - 1, n, 0.5))))
##         ci2 <- c(x[k + 1], Inf)
##         acl2 <- (1 - (sum(dbinom(0:k, n, 0.5))))
##         xl <- (((x[k] - x[k + 1]) * (conf.level - acl2))/(acl1 -
##           acl2)) + x[k + 1]

```

```

##          xu <- Inf
##          ici <- c(xl, xu)
##      }
##  }
##  else {
##      p1 <- sum(dbinom(0:s, nt, 0.5))
##      p2 <- (1 - sum(dbinom(0:s - 1, nt, 0.5)))
##      pval <- min(2 * p1, 2 * p2, 1)
##      loc <- c(0:n)
##      prov <- (dbinom(loc, n, 0.5))
##      k <- loc[cumsum(prov) > (1 - conf.level)/2][1]
##      if (k < 1) {
##          conf.level <- (1 - 2 * (sum(dbinom(k, n, 0.5))))
##          xl <- x[1]
##          xu <- x[n]
##          ici <- c(xl, xu)
##      }
##      else {
##          ci1 <- c(x[k], x[n - k + 1])
##          acl1 <- (1 - 2 * (sum(dbinom(0:k - 1, n, 0.5))))
##          ci2 <- c(x[k + 1], x[n - k])
##          acl2 <- (1 - 2 * (sum(dbinom(0:k, n, 0.5))))
##          xl <- (((x[k] - x[k + 1]) * (conf.level - acl2))/(acl1 -
##              acl2)) + x[k + 1]
##          xu <- (((x[n - k + 1] - x[n - k]) * (conf.level -
##              acl2))/(acl1 - acl2)) + x[n - k]
##          ici <- c(xl, xu)
##      }
##  }
##  }
##  }
##  else {
##      if (length(x) != length(y))
##          stop("Length of x must equal length of y")
##      xy <- sort(x - y)
##      diff <- (xy - md)
##      n <- length(xy)
##      nt <- length(xy) - sum(diff == 0)
##      s <- sum(diff > 0)
##      dname <- paste(deparse(substitute(x)), " and ", deparse(substitute(y)),
##          sep = "")
##      estimate <- median(xy)
##      method <- c("Dependent-samples Sign-Test")
##      names(estimate) <- c("median of x-y")
##      names(md) <- "median difference"
##      names(s) <- "S"
##      CIS <- "Conf Intervals"
##      if (alternative == "less") {
##          pval <- sum(dbinom(0:s, nt, 0.5))
##          loc <- c(0:n)
##          prov <- (dbinom(loc, n, 0.5))
##          k <- loc[cumsum(prov) > (1 - conf.level)][1]
##          if (k < 1) {
##              conf.level <- (1 - (sum(dbinom(k, n, 0.5))))

```

```

##          xl <- -Inf
##          xu <- xy[n]
##          ici <- c(xl, xu)
##      }
##      else {
##          ci1 <- c(-Inf, xy[n - k + 1])
##          acl1 <- (1 - (sum(dbinom(0:k - 1, n, 0.5))))
##          ci2 <- c(-Inf, xy[n - k])
##          acl2 <- (1 - (sum(dbinom(0:k, n, 0.5))))
##          xl <- -Inf
##          xu <- (((xy[n - k + 1] - xy[n - k]) * (conf.level -
##              acl2))/(acl1 - acl2)) + xy[n - k]
##          ici <- c(xl, xu)
##      }
##  }
##  else if (alternative == "greater") {
##      pval <- (1 - sum(dbinom(0:s - 1, nt, 0.5)))
##      loc <- c(0:n)
##      prov <- (dbinom(loc, n, 0.5))
##      k <- loc[cumsum(prov) > (1 - conf.level)][1]
##      if (k < 1) {
##          conf.level <- (1 - (sum(dbinom(k, n, 0.5))))
##          xl <- xy[1]
##          xu <- Inf
##          ici <- c(xl, xu)
##      }
##      else {
##          ci1 <- c(xy[k], Inf)
##          acl1 <- (1 - (sum(dbinom(0:k - 1, n, 0.5))))
##          ci2 <- c(xy[k + 1], Inf)
##          acl2 <- (1 - (sum(dbinom(0:k, n, 0.5))))
##          xl <- (((xy[k] - xy[k + 1]) * (conf.level - acl2))/(acl1 -
##              acl2)) + xy[k + 1]
##          xu <- Inf
##          ici <- c(xl, xu)
##      }
##  }
##  else {
##      p1 <- sum(dbinom(0:s, nt, 0.5))
##      p2 <- (1 - sum(dbinom(0:s - 1, nt, 0.5)))
##      pval <- min(2 * p1, 2 * p2, 1)
##      loc <- c(0:n)
##      prov <- (dbinom(loc, n, 0.5))
##      k <- loc[cumsum(prov) > (1 - conf.level)/2][1]
##      if (k < 1) {
##          conf.level <- (1 - 2 * (sum(dbinom(k, n, 0.5))))
##          xl <- xy[1]
##          xu <- xy[n]
##          ici <- c(xl, xu)
##      }
##      else {
##          ci1 <- c(xy[k], xy[n - k + 1])
##          acl1 <- (1 - 2 * (sum(dbinom(0:k - 1, n, 0.5))))

```

```

##         ci2 <- c(xy[k + 1], xy[n - k])
##         acl2 <- (1 - 2 * (sum(dbinom(0:k, n, 0.5))))
##         x1 <- (((xy[k] - xy[k + 1]) * (conf.level - acl2))/(acl1 -
##         acl2)) + xy[k + 1]
##         xu <- (((xy[n - k + 1] - xy[n - k]) * (conf.level -
##         acl2))/(acl1 - acl2)) + xy[n - k]
##         ici <- c(x1, xu)
##     }
## }
## }
## if (k < 1) {
##     cint <- ici
##     attr(cint, "conf.level") <- conf.level
##     rval <- structure(list(statistic = s, p.value = pval,
##         estimate = estimate, null.value = md, alternative = alternative,
##         method = method, data.name = dname, conf.int = cint))
##     oldClass(rval) <- "htest"
##     return(rval)
## }
## else {
##     result1 <- c(acl2, ci2)
##     result2 <- c(conf.level, ici)
##     result3 <- c(acl1, ci1)
##     Confidence.Intervals <- round(as.matrix(rbind(result1,
##         result2, result3)), 4)
##     cnames <- c("Conf.Level", "L.E.pt", "U.E.pt")
##     rnames <- c("Lower Achieved CI", "Interpolated CI", "Upper Achieved CI")
##     dimnames(Confidence.Intervals) <- list(rnames, cnames)
##     cint <- ici
##     attr(cint, "conf.level") <- conf.level
##     rval <- structure(list(statistic = s, parameter = NULL,
##         p.value = pval, conf.int = cint, estimate = estimate,
##         null.value = md, alternative = alternative, method = method,
##         data.name = dname))
##     oldClass(rval) <- "htest"
##     #print(rval)
##     list(rval, Confidence.Intervals)
## }
## }
##
## library("BSDA")
## set.seed(100614)
## nsim <- 10000
## sigN <- function(distr = rnorm(100), o){
##     foo <- my.SIGN.test(distr, md = 0, alternative = "less", conf.level = 0.95)[[1]]$p.value
##     return(foo)
## }
##
## #sigN()
## sig <- replicate(nsim, sigN())
##
## ss <- histT(sig, "Distribution of p-values from sign test") +
##     geom_vline(xintercept = 0.05, col = "red")

```

```

##
## ss
##
## stpv <- sum(sig < 0.05) / nsim
## ```

##
##
## ## 2c
## ```{r, echo = F, fig.height = 3}
##
## ## paired ttest
## set.seed(100614)
## nsim <- 10000
## pvalues <- replicate(nsim, pair(rnorm(100, mean = -0.2, sd = 1)))
## vv <- histT(diffs = pvalues, title = "Distribution of p-values from paired ttest
##           when actual difference is = -0.2")
## vv + geom_vline(xintercept = 0.05, color = "red")
## prop <- sum(pvalues < 0.05)/ nsim
##
##
## ## sign test
## set.seed(100614)
## sig <- replicate(nsim, sigN(rnorm(100, mean = -0.2, sd = 1)))
## ss <- histT(sig, "Distribution of p-values from sign test
##           when actual difference in means is -0.2") +
##       geom_vline(xintercept = 0.05, col = "red")
## ss
## stpv <- sum(sig < 0.05) / nsim
## ```
##

##
## ##### 2d#####
## ```{r, echo = F, fig.height=3}
## ## paired ttest
## set.seed(100614)
## nsim <- 10000
## pvalues <- replicate(nsim, pair((rexp(100) - 1)))
## vv <- histT(diffs = pvalues, title = "Distribution of p-values from paired ttest
##           when samples come from shifted exponential dist.")
## vv + geom_vline(xintercept = 0.05, color = "red")
## prop <- sum(pvalues < 0.05)/ nsim
##
##
## ## sign test
## set.seed(100614)
## sig <- replicate(nsim, sigN((rexp(100) - 1)))
## ss <- histT(sig, "Distribution of p-values from sign test
##           when samples come from exponential dist") +
##       geom_vline(xintercept = 0.05, col = "red")
## ss
## stpv <- sum(sig < 0.05) / nsim

```

```

##
## ```

##
## ##### 2dii#####

## ```{r, echo =F, eval = F}
## nums <- c(20.1, 22.9, 18.8, 20.9, 20.9, 22.7, 21.4, 20,
##          38.5, 25.8, 22.0, 23, 37.6, 30, 24.5)
## gp <- c(rep("NT", 8), rep("T", 7))
##
## data.frame(gp, nums, rank = sort(nums), n = 1:(7+8))
##
## tt = sum(15, 12, 7, 10, 14, 13, 11)
## tt
##
## sum(15, 12, 7, 10, 14, 13, 11)/7
## 82/7
##
##
## ranks <- c(1,2,3,4.5,4.5,6,7,8,9,10,11,12,13,14,15)
## mean(ranks)
##
## sqs <- numeric()
## for(i in 1:length(ranks)) {
##     sqs[i] <- (8-ranks[i])^2
## }
## sqrt(sum(sqs) / 14)
## sd(ranks)
##
##
## 4.4681*sqrt((8*7) / (8+7))
##
## sd(ranks)
##
## zz <- (82 - 56) / 8.633189
##
## 1- pnorm(zz)
## ```

##
##
##### ## 3 (Ch.4 #21) #####
## ```{r, warning = F,echo = T}
## #install.packages("coin")
## library(coin)
## nums <- c(20.1, 22.9, 18.8, 20.9, 20.9, 22.7, 21.4, 20,
##          38.5, 25.8, 22.0, 23, 37.6, 30, 24.5)
## gp <- c(rep("NT", 8), rep("T", 7))
## obsRank <- rank(nums, ties.method = "average")
## met <- data.frame(gp, nums, obsRank)
## mt <- sum(obsRank[9:15])
## average <- mean(obsRank)
## sd <- sd(obsRank)

```

```

## n = 7 ## number of observations in smallest group
## MEANT <- n * average
## SDT <- sd*sqrt((8*7)/(8+7))
## z <- (mt - MEANT)/SDT
## p <- pnorm(-abs(z))
## a <- wilcox_test(nums~gp, data = met, distribution = "exact", conf.level = 0.05, alternative = "less")
## b <- wilcox.test(nums~gp, alternative = "less", data = met, correct = F, exact = T)
## b; a
## ```
##

##
## ## 3 (Ch.4 #22)
## ```{r, echo = F}
##
## deltaGuess <- -2
## numsCopy <- nums
## numsCopy[1:8] <- numsCopy[1:8] - deltaGuess
## wilcox_test(numsCopy~gp, data = met, distribution = "exact")
##
## wilcox_test(nums~gp, data = met, distribution = "exact", conf.int = T)
## ?wilcox_test()
## ```

##
##### #4 #####
## ```{r}
## zinc <- read.csv("data/ex0524.csv")
## zinc
## nrow(zinc)
## table(zinc$GROUP)
## tapply(zinc$ZINC,FUN = mean, INDEX =zinc$GROUP)
## tapply(zinc$ZINC,FUN = var, INDEX =zinc$GROUP)
##
##
## sp <- sqrt((5*209.6 + 11*435.7197 + 4 * 743.7) / (6 + 12 + +5 - 3))
## sp^2
##
## TT <- (178-177.0833)/ (sp*sqrt(1/6 + 1/12))
## TT
##
## 1-pt(TT, df = 20)
## class(zinc$GROUP)
##
## mod1 <- aov(ZINC~GROUP, data = zinc)
## summary(mod1)
##
##
##
## ggplot(zinc, aes(x = GROUP, y = ZINC)) +
##   geom_boxplot()
##
##

```



```

## ### Look at assumptions
## ggplot(zinc,(aes(ZINC))) +
##     geom_histogram() +
##     facet_wrap(~GROUP) +
##     theme_bw()
##
## ```

## ```{r}
## Ybar <- mean(zinc$ZINC)
##
## tapply(X = zinc$ZINC, FUN = mean, INDEX = zinc$GROUP)
## tapply(X = zinc$ZINC, FUN = length, INDEX = zinc$GROUP)
##
## ### SS for equal means model
## mod <- lm(zinc$ZINC ~ rep(Ybar, length(zinc$ZINC)))
## ResidSumSqReduced <- sum((mod$residuals)^2)
##
## ### SS for separate means model
## zinc$groupMeans <- c(rep(178, 6), rep(177.0833, 12), rep(179.2, 5))
##
## ResidSumSqFull <- sum((zinc$ZINC - zinc$groupMeans)^2)
##
##
## ExtraSumSq <- ResidSumSqReduced - ResidSumSqFull
##
##
## mseBetween <- ExtraSumSq / 2
##
## mseWithin <- ResidSumSqFull / 20
##
## FStat <- mseBetween / mseWithin
##
## 1-pf(FStat, 2, 20)
##
## ```

## ```{r, fig.width = 4, fig.height = 4}
## plot(mod1)

```