

# CH6: Inference for comparing two means

## 1 Two-sample t-test and Wilcoxon Rank-Sum test

The two-sample t-test is used to compare means using independent samples. The Wilcoxon Rank-sum test is a non-parametric alternative that does not require the assumption of normality.

```
library(tidyverse)
library(broom)
ratlead <- read.csv("C:/hess/STAT511_FA11/RData/CH6_RatLead.csv")
str(ratlead)

## 'data.frame': 20 obs. of 2 variables:
## $ trt: Factor w/ 2 levels "control","deficient": 1 1 1 1 1 1 1 1 1 1 ...
## $ y : num 5.4 6.2 3.1 3.8 6.5 5.8 6.4 4.5 4.9 4 ...

ratlead

##      trt      y
## 1 control 5.4
## 2 control 6.2
## 3 control 3.1
## 4 control 3.8
## 5 control 6.5
## 6 control 5.8
## 7 control 6.4
## 8 control 4.5
## 9 control 4.9
## 10 control 4.0
## 11 deficient 8.8
## 12 deficient 9.5
## 13 deficient 10.6
## 14 deficient 9.6
## 15 deficient 7.5
## 16 deficient 6.9
## 17 deficient 7.4
## 18 deficient 6.5
## 19 deficient 10.5
## 20 deficient 8.3
```

### 1.1 Summary Statistics and Graphics

In this example, we use the `summarize()` and `group_by()` functions from the `dplyr` package to calculate summary statistics for each group. Another option is to use the `aggregate` function.

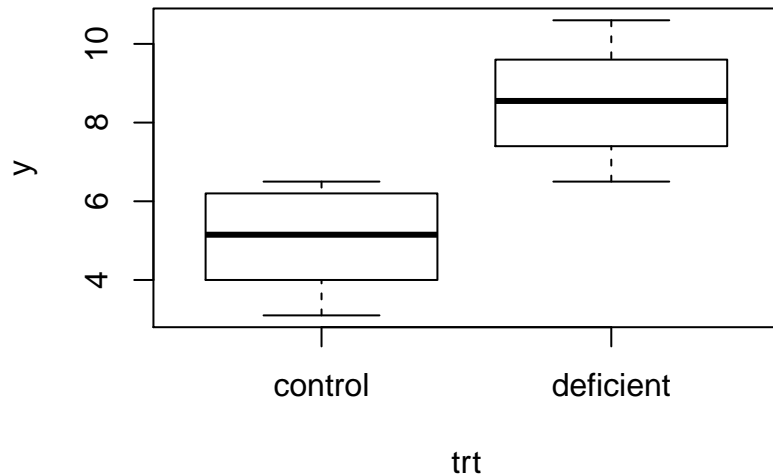
```
SumStats <- summarise(group_by(ratlead, trt),
                        n = n(),
                        mean = mean(y),
                        sd = sd(y),
                        se = sd/sqrt(n))

SumStats
```

```
## # A tibble: 2 x 5
```

```
##   trt          n mean    sd    se
##   <fct>      <int> <dbl> <dbl> <dbl>
## 1 control      10  5.06  1.19 0.376
## 2 deficient     10  8.56  1.47 0.465
```

```
boxplot(y ~ trt, data = ratlead)
```



## 1.2 Two-sample t-test (Approach #1)

This approach works well for this data which is in “long” format, with group/trt information provided in a column. Default for `t.test` is Welch/Satterthwaite t-test (NOT assuming equal variances). Use `var.equal = T` to run pooled variance t-test (assuming equal variances).

```
t.test(y ~ trt, data = ratlead, var.equal = TRUE)
```

```
##
## Two Sample t-test
##
## data: y by trt
## t = -5.8507, df = 18, p-value = 1.532e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -4.756813 -2.243187
## sample estimates:
## mean in group control mean in group deficient
## 5.06 8.56
```

```
t.test(y ~ trt, data = ratlead)
```

```
##
## Welch Two Sample t-test
##
## data: y by trt
```

```
## t = -5.8507, df = 17.241, p-value = 1.822e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -4.760793 -2.239207
## sample estimates:
## mean in group control mean in group deficient
## 5.06 8.56
```

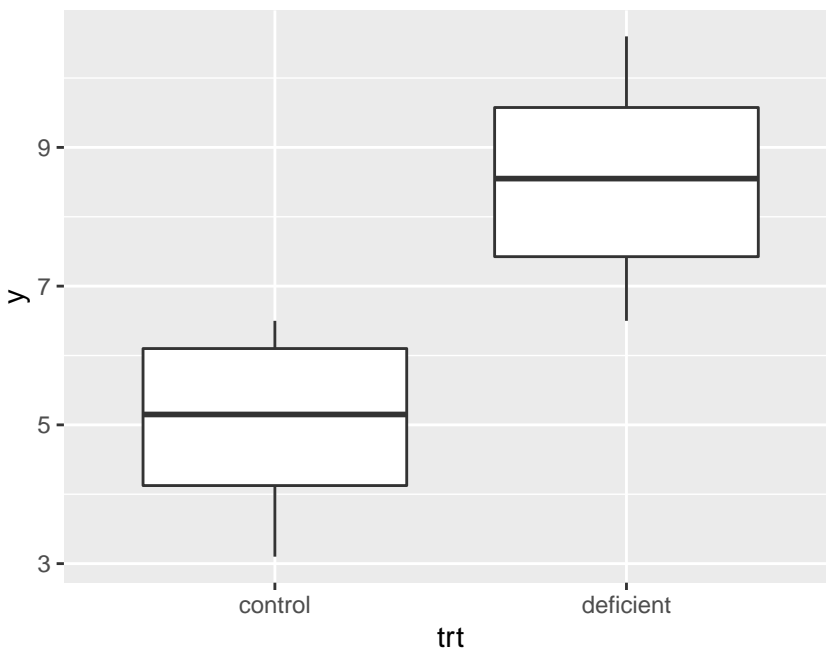
### 1.3 tidyverse

```
SumStats <- ratlead %>%
  group_by(trt) %>%
  summarise(n = n(),
            mean = mean(y),
            sd = sd(y),
            se = sd/sqrt(n))
```

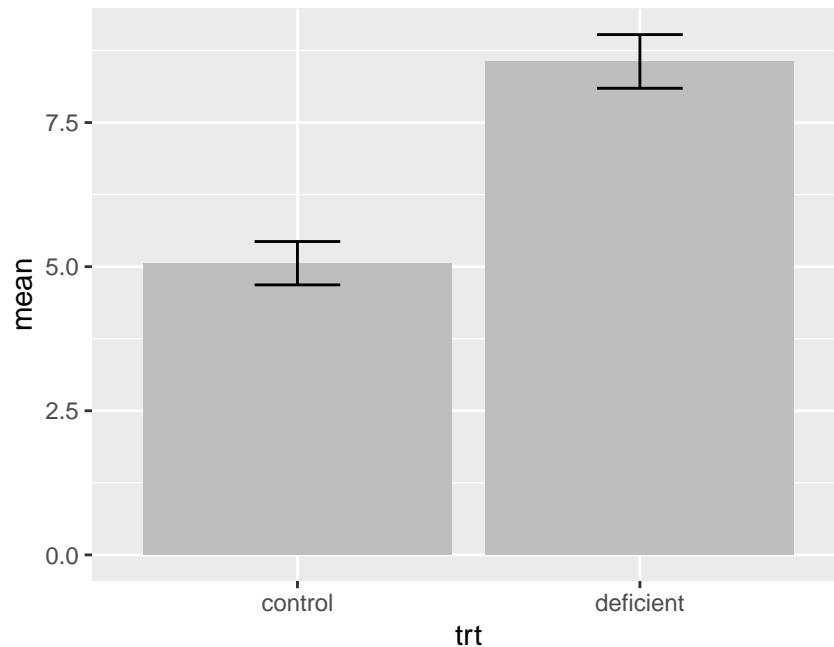
SumStats

```
## # A tibble: 2 x 5
##   trt      n mean  sd  se
##   <fct> <int> <dbl> <dbl> <dbl>
## 1 control    10  5.06  1.19 0.376
## 2 deficient   10  8.56  1.47 0.465
```

```
qplot(x = trt, y = y, geom = "boxplot", data = ratlead)
```



```
ggplot(aes(x = trt, y = mean), data = SumStats) +
  geom_col(fill = "grey") +
  geom_errorbar(aes(ymin = mean-se, ymax = mean+se), width = .25)
```



```
tidy(t.test(y ~ trt, data = ratlead, var.equal = TRUE))
```

```
## # A tibble: 1 x 9
##   estimate1 estimate2 statistic p.value parameter conf.low conf.high method
##   <dbl>      <dbl>      <dbl>  <dbl>      <dbl>    <dbl>    <dbl> <chr>
## 1     5.06     8.56    -5.85 1.53e-5      18     -4.76    -2.24 "Two~
## # ... with 1 more variable: alternative <chr>
```

**For Illustration:** Transpose from “long” to “wide” format using `spread()` from `tidyr`. In practice, long format is usually preferred!

```
Wide <- ratlead %>%
  mutate(ID = c(rep(1:10), rep(1:10))) %>%
  spread(key = trt, value = y)
str(Wide)
```

```
## 'data.frame': 10 obs. of 3 variables:
## $ ID : int 1 2 3 4 5 6 7 8 9 10
## $ control : num 5.4 6.2 3.1 3.8 6.5 5.8 6.4 4.5 4.9 4
## $ deficient: num 8.8 9.5 10.6 9.6 7.5 6.9 7.4 6.5 10.5 8.3
```

```
Wide
```

```
##   ID control deficient
## 1  1     5.4      8.8
## 2  2     6.2      9.5
## 3  3     3.1     10.6
## 4  4     3.8      9.6
## 5  5     6.5      7.5
## 6  6     5.8      6.9
## 7  7     6.4      7.4
## 8  8     4.5      6.5
## 9  9     4.9     10.5
## 10 10     4.0      8.3
```

## 1.4 Two-sample t-test (Approach #2)

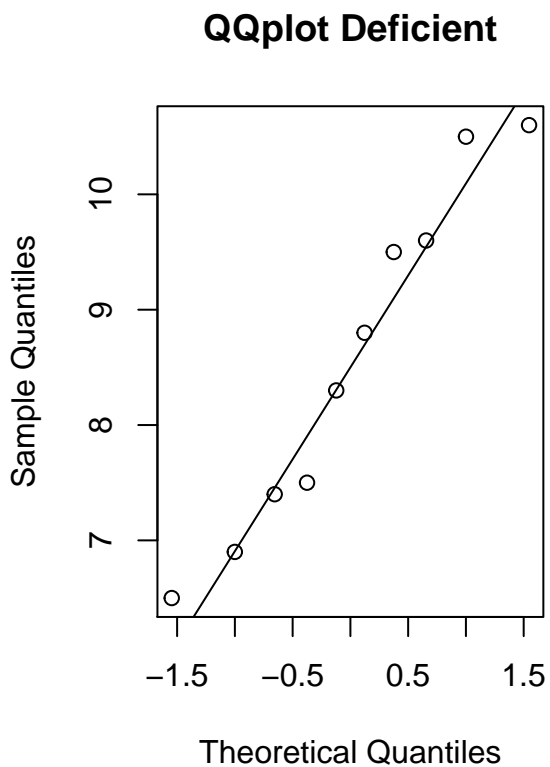
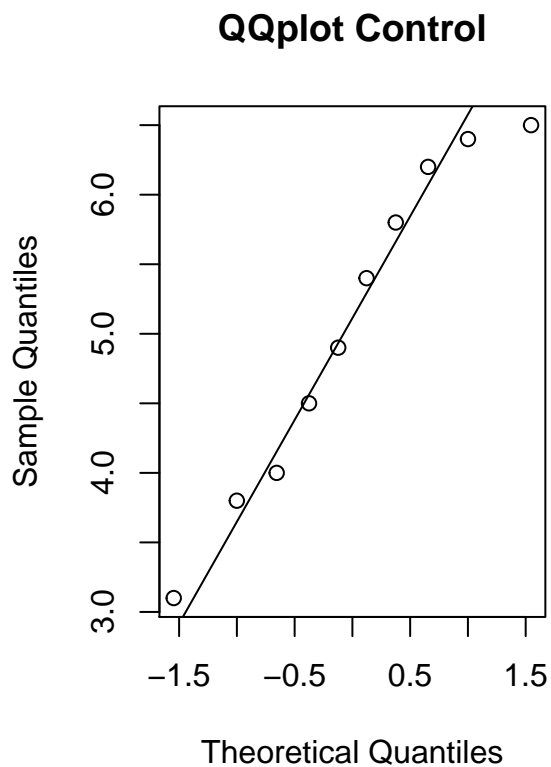
Just for illustration here. This approach can be helpful when working with data in “wide” format.

```
t.test(Wide$control, Wide$deficient)
```

```
##
##  Welch Two Sample t-test
##
## data:  Wide$control and Wide$deficient
## t = -5.8507, df = 17.241, p-value = 1.822e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   -4.760793 -2.239207
## sample estimates:
## mean of x mean of y
##      5.06      8.56
```

## 1.5 Evaluating Normality

```
par(mfrow=c(1,2))
qqnorm(Wide$control, main = "QQplot Control")
qqline(Wide$control)
qqnorm(Wide$deficient, main = "QQplot Deficient")
qqline(Wide$deficient)
```



## 1.6 Wilcoxon Two-sample (Rank-sum) test

Also see later example using exact test.

```
wilcox.test(y ~ trt, data = ratlead)
```

```
## Warning in wilcox.test.default(x = c(5.4, 6.2, 3.1, 3.8, 6.5, 5.8, 6.4, :
```

```
## cannot compute exact p-value with ties
```

```
##
```

```
## Wilcoxon rank sum test with continuity correction
```

```
##
```

```
## data: y by trt
```

```
## W = 0.5, p-value = 0.000211
```

```
## alternative hypothesis: true location shift is not equal to 0
```

```
rm(ratlead, SumStats, Wide)
```

## 2 Power for Two Sample t-test

### 2.1 Confidence Interval Width

Calculate ME for sample sizes between 5-10

```
n <- seq(5, 10, 1)
sd <- 4
alpha <- 0.05
ME <- qt(1-alpha/2, 2*n-2)*sd*sqrt(2/n)
out <- data.frame(n1 = n, n2 = n, ME)
out
```

```
##   n1 n2      ME
## 1  5  5 5.833780
## 2  6  6 5.145666
## 3  7  7 4.658498
## 4  8  8 4.289573
## 5  9  9 3.997332
## 6 10 10 3.758244
```

### 2.2 Power for ONE-sided two-sample t-test

```
#Using power.t.test
power.t.test(n = 9, delta = 6, sd = 4, sig.level = 0.05,
             type = "two.sample", alternative = "one.sided")
```

```
##
##      Two-sample t test power calculation
##
##              n = 9
##            delta = 6
##              sd = 4
##          sig.level = 0.05
##            power = 0.9189915
##      alternative = one.sided
##
## NOTE: n is number in each group
```

```
#For illustration: power "by hand" using noncentrality parameter
1-pt(1.746, df = 16, ncp = 3.18)
```

```
## [1] 0.9186899
```

### 2.3 Power for Two-sided one-sample t-test

```
#Using power.t.test
power.t.test(n = 9, delta = 6, sd = 4, sig.level = 0.05,
             type = "two.sample", alternative = "two.sided")
```

```
##
##      Two-sample t test power calculation
##
```

```
##           n = 9
##         delta = 6
##           sd = 4
##       sig.level = 0.05
##         power = 0.8476098
##   alternative = two.sided
##
## NOTE: n is number in *each* group
#For illustration: power "by hand" using noncentrality parameter
pt(-2.120, df = 16, ncp = 3.18)+(1-pt(2.120, df = 16, ncp = 3.18))

## [1] 0.8471516
```

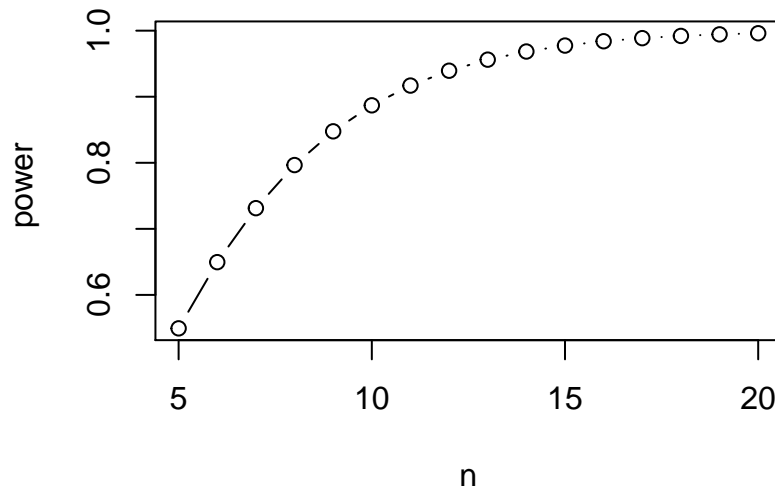
## 2.4 Graph of Power vs Sample Size

```
nvec <- seq(5, 20, 1)
powerout1 <- power.t.test(n = nvec, delta = 6, sd = 4, sig.level = 0.05,
                          type = "two.sample", alternative = "two.sided")
powerout1

##
##   Two-sample t test power calculation
##
##           n = 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
##         delta = 6
##           sd = 4
##       sig.level = 0.05
##         power = 0.5493642, 0.6495744, 0.7313279, 0.7965441, 0.8476098, 0.8869701, 0.9168991, 0.939
##   alternative = two.sided
##
## NOTE: n is number in *each* group
plot(powerout1$power ~ powerout1$n,
     type = "b", xlab = "n", ylab = "power ",
     main = "Power vs Sample Size")
```

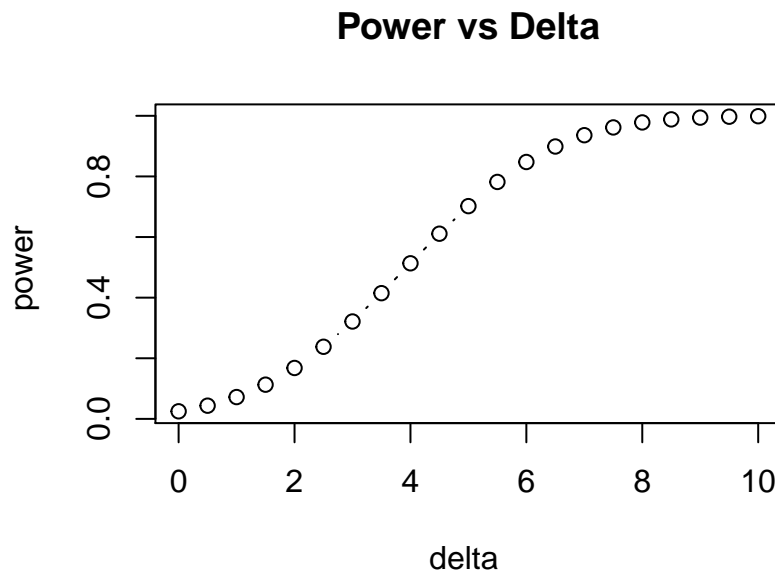


## Power vs Sample Size



## 2.5 Graph of Power vs Delta (Difference between means)

```
deltavec <- seq(0, 10, 0.5)
powerout2 <- power.t.test(n = 9, delta = deltavec, sd = 4, sig.level = 0.05,
                          type = "two.sample", alternative = "two.sided")
plot(powerout2$power ~ powerout2$delta,
     type = "b", xlab = "delta ", ylab = "power ",
     main = "Power vs Delta")
```



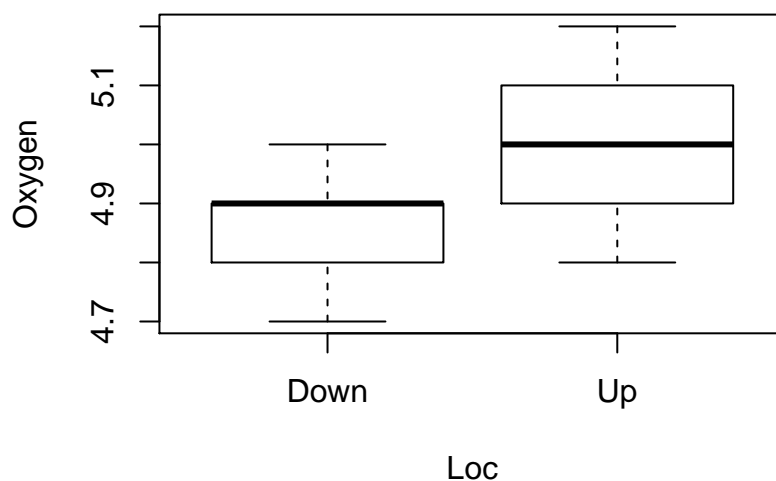
### 3 Wilcoxon (Two-Sample) Rank-Sum Test

The Wilcoxon Rank-sum test is a non-parametric alternative to the two-sample t-test that does not require the assumption of normality.

```
library(coin)
oxygen <- read.csv("C:/hess/STAT511_FA11/RData/CH6_oxygen.csv")
oxygen
```

```
##      Loc Oxygen
## 1    Up    4.8
## 2    Up    5.2
## 3    Up    5.0
## 4    Up    4.9
## 5    Up    5.1
## 6  Down    5.0
## 7  Down    4.7
## 8  Down    4.9
## 9  Down    4.8
## 10 Down    4.9
```

```
boxplot(Oxygen ~ Loc, data = oxygen)
```



#### 3.1 Using wilcox.test

A version of this test can be run using the `wilcox.test()` function from base R. But this version will not compute the exact test when there are “ties” in the data. Also note that for the one-sided test the ordering is determined by alphabetical order. For this data, Down = Group1.

```
wilcox.test(Oxygen ~ Loc, data = oxygen)
```

```
## Warning in wilcox.test.default(x = c(5, 4.7, 4.9, 4.8, 4.9), y = c(4.8, :
## cannot compute exact p-value with ties
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: Oxygen by Loc
## W = 6, p-value = 0.2017
## alternative hypothesis: true location shift is not equal to 0
wilcox.test(Oxygen ~ Loc, data = oxygen, alternative = "greater")

## Warning in wilcox.test.default(x = c(5, 4.7, 4.9, 4.8, 4.9), y = c(4.8, :
## cannot compute exact p-value with ties

##
## Wilcoxon rank sum test with continuity correction
##
## data: Oxygen by Loc
## W = 6, p-value = 0.9318
## alternative hypothesis: true location shift is greater than 0
```

### 3.2 Using the coin package

If we want an exact test in the case that we have “ties” in our data, we can use `wilcox_test()` from the `coin` package.

```
wilcox_test(Oxygen ~ Loc, data = oxygen, distribution = "exact")

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: Oxygen by Loc (Down, Up)
## Z = -1.3832, p-value = 0.246
## alternative hypothesis: true mu is not equal to 0
rm(oxygen)
```

## 4 Paired t-test and Wilcoxon Signed Rank test

The paired t-test is used to compare means using paired data. The Wilcoxon Sign-Rank test is a non-parametric alternative that does not require the assumption of normality.

```
library(coin)
dogs <- read.csv("C:/hess/STAT511_FA11/RData/CH6_Dogs.csv")
str(dogs)
```

```
## 'data.frame': 14 obs. of 3 variables:
## $ Dog: int 1 2 3 4 5 6 7 8 9 10 ...
## $ P : int 250 271 243 252 266 272 293 296 301 298 ...
## $ B : int 258 285 245 250 268 278 280 305 319 308 ...
```

### 4.1 Paired t-test (Approach #1)

**Primarily for illustration.** We can calculate the differences and then use the one-sample t-test.

```
dogs$Diff <- dogs$P - dogs$B
str(dogs)
```

```
## 'data.frame': 14 obs. of 4 variables:
## $ Dog : int 1 2 3 4 5 6 7 8 9 10 ...
## $ P : int 250 271 243 252 266 272 293 296 301 298 ...
## $ B : int 258 285 245 250 268 278 280 305 319 308 ...
## $ Diff: int -8 -14 -2 2 -2 -6 13 -9 -18 -10 ...
```

```
mean(dogs$Diff); sd(dogs$Diff)
```

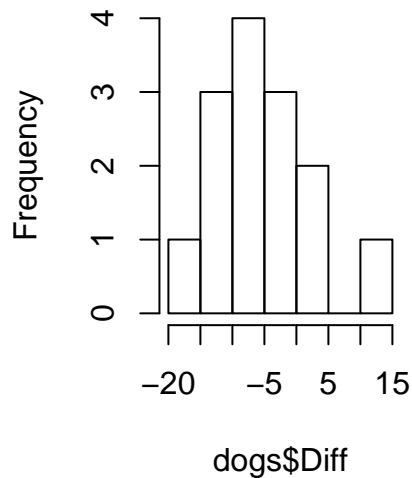
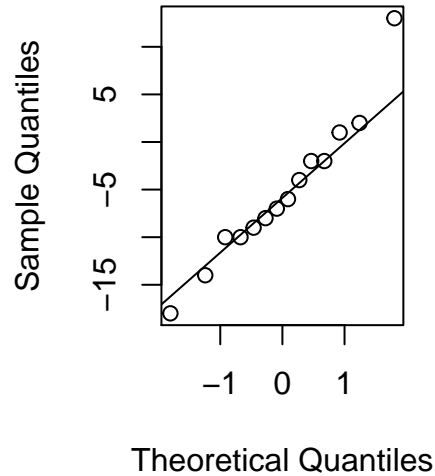
```
## [1] -5.285714
```

```
## [1] 7.630189
```

```
t.test(dogs$Diff)
```

```
##
## One Sample t-test
##
## data: dogs$Diff
## t = -2.592, df = 13, p-value = 0.02234
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -9.6912541 -0.8801745
## sample estimates:
## mean of x
## -5.285714
```

```
par(mfrow=c(1,2))
hist(dogs$Diff)
qqnorm(dogs$Diff); qqline(dogs$Diff)
```

**Histogram of dogs\$Diff****Normal Q-Q Plot**

## 4.2 Paired t-test (Approach #2)

Exactly the same results as above.

```
t.test(dogs$P, dogs$B, paired = TRUE)
```

```
##
## Paired t-test
##
## data: dogs$P and dogs$B
## t = -2.592, df = 13, p-value = 0.02234
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -9.6912541 -0.8801745
## sample estimates:
## mean of the differences
## -5.285714
```

## 4.3 Wilcoxon (Paired) Signed-Rank test

Exact test (with ties) available from the coin package, but note the change in syntax.

```
wilcox.test(dogs$P, dogs$B, paired = T)
```

```
## Warning in wilcox.test.default(dogs$P, dogs$B, paired = T): cannot compute
## exact p-value with ties
##
## Wilcoxon signed rank test with continuity correction
##
## data: dogs$P and dogs$B
## V = 16, p-value = 0.02365
```

```
## alternative hypothesis: true location shift is not equal to 0
wilcoxsign_test(P ~ B, data = dogs, distribution = "exact")

##
## Exact Wilcoxon-Pratt Signed-Rank Test
##
## data: y by x (pos, neg)
## stratified by block
## Z = -2.2942, p-value = 0.01941
## alternative hypothesis: true mu is not equal to 0
rm(dogs)
```

## 5 Bootstrap Two Sample t-confidence interval

```
arsenic <- read.csv("c:/hess/STAT511_FA11/RData/CH6_Arsenic.csv")
str(arsenic)

## 'data.frame': 42 obs. of 2 variables:
## $ Group: int 1 1 1 1 1 1 1 1 1 1 ...
## $ y : int 40 110 70 70 40 60 70 90 80 80 ...

aggregate(y ~ Group, data = arsenic, mean)

## Group y
## 1 1 95.00000
## 2 2 84.44444

aggregate(y ~ Group, data = arsenic, sd)

## Group y
## 1 1 92.21902
## 2 2 42.03951

#Standard two-sample t-CI
t.test(y ~ Group, data = arsenic)

##
## Welch Two Sample t-test
##
## data: y by Group
## t = 0.4962, df = 33.982, p-value = 0.6229
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -32.67683 53.78795
## sample estimates:
## mean in group 1 mean in group 2
## 95.00000 84.44444

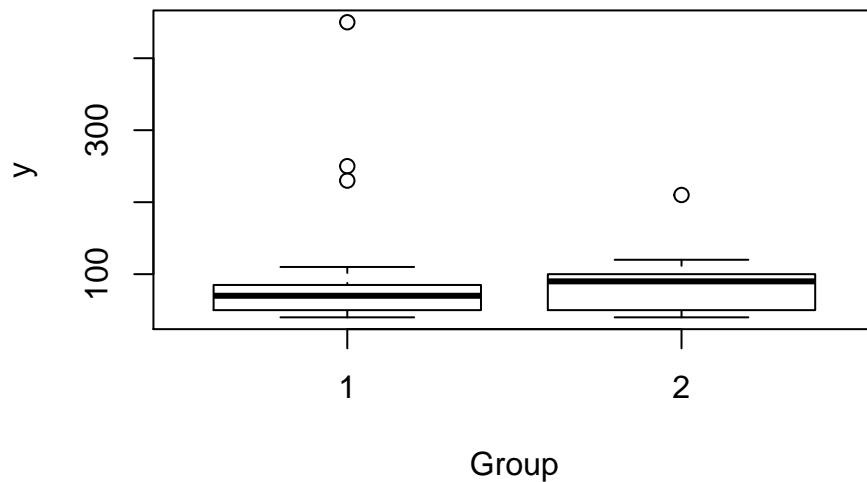
#Subtract off the group means
Group1 <- arsenic[arsenic$Group==1, 2] - 95; length(Group1)

## [1] 24

Group2 <- arsenic[arsenic$Group==2, 2] - 84.44444; length(Group2)

## [1] 18

boxplot(y ~ Group, data = arsenic)
```



## 5.1 “By Hand” For Illustration

Use lapply and sample to take 10000 bootstrap samples WITH replacement, for each group separately.

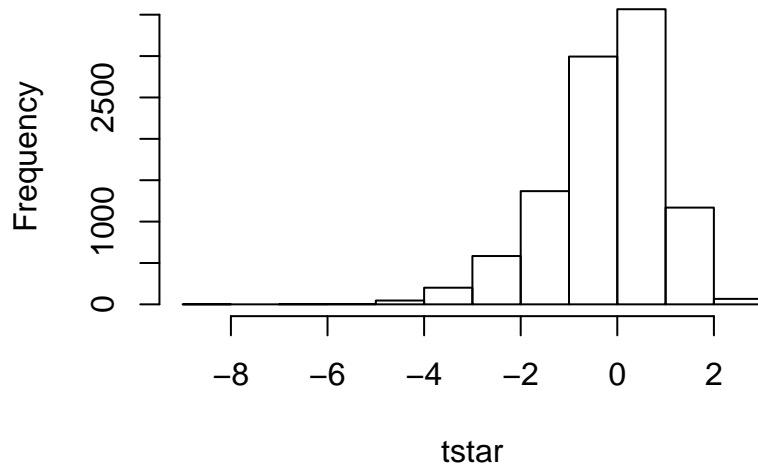
```
set.seed(3012)
resamples1 <- lapply(1:10000, function(i)
  sample(Group1, size=length(Group1), replace = T))
resamples2 <- lapply(1:10000, function(i)
  sample(Group2, size=length(Group2), replace = T))
resamples1 <- simplify2array(resamples1)
resamples2 <- simplify2array(resamples2)
dim(resamples1)

## [1] 24 10000

colmeans1 <- apply(resamples1, 2, mean)
colmeans2 <- apply(resamples2, 2, mean)
colsd1 <- apply(resamples1, 2, sd)
colsd2 <- apply(resamples2, 2, sd)
tstar <- (colmeans1 - colmeans2) / sqrt((colsd1^2 / length(Group1) + (colsd2^2) / length(Group2))
hist(tstar, main = "Histogram of tstar")
```



## Histogram of tstar



```
#Bootstrap CI
t025 <- quantile(tstar, prob = 0.975)
t975 <- quantile(tstar, prob = 0.025)
LB <- (95-84.44444)-t025*sqrt((92.2190^2)/24+(42.0395^2)/18)
UB <- (95-84.44444)-t975*sqrt((92.2190^2)/24+(42.0395^2)/18)
CI <- c(LB,UB); names(CI)<-c(); CI

## [1] -23.73967 74.62476
```

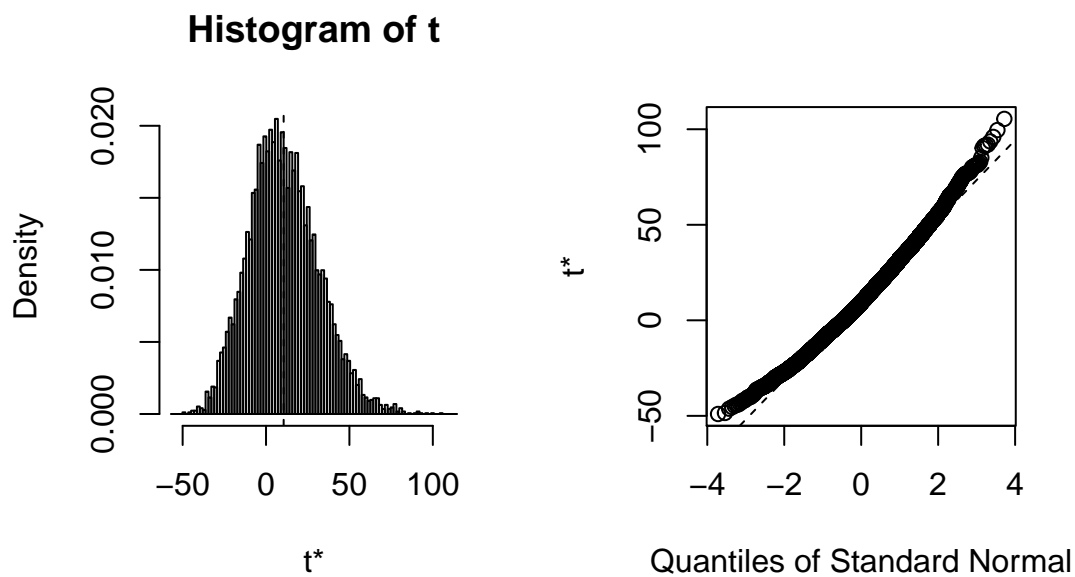
## 5.2 Boot Example

```
library(boot)
diff.means <- function(d, f)
{
  n <- nrow(d)
  gp1 <- 1:table(as.numeric(d$Group))[1]
  m1 <- sum(d[gp1,2] * f[gp1])/sum(f[gp1])
  m2 <- sum(d[-gp1,2] * f[-gp1])/sum(f[-gp1])
  ss1 <- sum(d[gp1,2]^2 * f[gp1]) - (m1 * m1 * sum(f[gp1]))
  ss2 <- sum(d[-gp1,2]^2 * f[-gp1]) - (m2 * m2 * sum(f[-gp1]))
  c(m1 - m2, (ss1 + ss2)/(sum(f) - 2))
}
set.seed(5964)
results<-boot(arsenic, diff.means, R = 10000, stype = "f", strata = arsenic[,1])
results

##
## STRATIFIED BOOTSTRAP
##
##
## Call:
## boot(data = arsenic, statistic = diff.means, R = 10000, stype = "f",
##       strata = arsenic[, 1])
```

```
##
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1*   10.55556    0.1852917   20.89038
## t2* 5641.11111 -214.1927674  3025.31346
```

```
plot(results)
```



```
boot.ci(results, type = "all")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, type = "all")
##
## Intervals :
## Level      Normal          Basic          Studentized
## 95%  (-30.57,  51.31 )  (-33.33,  47.64 )  (-22.84,  85.18 )
##
## Level      Percentile      BCa
## 95%  (-26.53,  54.44 )  (-21.39,  67.50 )
## Calculations and Intervals on Original Scale
```