

HW1 Biostats

Problem 1

1a) $H_0: m_1 = m_2$ & $H_a: m_1 \neq m_2$. Null: both teaching methods produce the same results. Alt: the two teaching methods produce different results.

```
m1 = c(79,66,57,91,42,59)
m2 = c(71,43,58,78,20,56)
alpha = 0.01

normality = function(data){
  shapiro = shapiro.test(data)
  p.value = shapiro$p.value
  if(p.value <= alpha){print("Reject the Null. Normality is not met")}
  else if(p.value > alpha){print("Do not reject the Null. Normality is met")}
}

normality(m1)
```

```
## [1] "Do not reject the Null. Normality is met"
```

```
normality(m2)
```

```
## [1] "Do not reject the Null. Normality is met"
```

1b) Under the assumption that all 12 students are different, we can use the t-test. The function below will conduct a t-test with 99% confidence level.

```
t_test_99ci = function(x, y){
  t_test_result = t.test(x, y, conf.int=TRUE, conf.level=0.99)
  p.value = t_test_result$p.value
  if(p.value <= alpha){print("Reject the Null")}
  else if(p.value > alpha){print("Do not reject the Null")}
}

t_test_99ci(m1, m2)
```

```
## [1] "Do not reject the Null"
```

The conducted t-test gives us a 99% confidence level to not reject the null hypothesis given the resulted p-value.

1b) $H_0: m_1 = m_2$. $H_a: m_1 \neq m_2$.

Since the students were tested twice, we can use the paired t-test. The function below will conduct a paired t-test with 99% confidence level.

```
paired_t_test = function(x,y,alpha){
  t_test_result = t.test(x, y, conf.int=TRUE, conf.level=0.99, paired = T)
  p.value = t_test_result$p.value
  if(p.value <= alpha){print("Reject the Null")}
  else if(p.value > alpha){print("Do not reject the Null")}
}
```

```
}
```

```
paired_t_test(m1, m2, alpha)
```

```
## [1] "Do not reject the Null"
```

Null hypothesis states that method 1 yields the same improvement as method 2. If the same students are tested twice, the paired t-test shows that the p value is not sufficient enough to reject the null hypothesis.

1c)

```
conf.test = t.test(m1, mu = 80, conf.level = 0.9)
conf.test$conf.int
```

```
## [1] 51.42129 79.91204
## attr("conf.level")
## [1] 0.9
```

The 90% confidence interval is [51.42, 79.91]. Ha: true mean != 80.

Problem 2

```
library(foreign)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
wcgs = read.dta("wcgs.dta")
g0 = wcgs %>%
  filter(typchd69 == 0)
g1 = wcgs %>%
  filter(typchd69 == 1)
g2 = wcgs %>%
  filter(typchd69 == 2)
g3 = wcgs %>%
  filter(typchd69 == 3)
```

2a)

```
alpha = 0.05
```

```
normality(g0$weight)
```

```
## [1] "Reject the Null. Normality is not met"
```

```
normality(g1$weight)
```

```
## [1] "Reject the Null. Normality is not met"
```

```
normality(g2$weight)
```

```
## [1] "Reject the Null. Normality is not met"
```

```
normality(g3$weight)
```

```
## [1] "Reject the Null. Normality is not met"
```

Normality is not met for any of the group.

2b)

```
bartlett_test = function(v1.v2, data){  
  result = bartlett.test(v1.v2, data)  
  p = result$p.value  
  
  if(p < alpha){print("Reject the Null. Variances are different across groups")}  
  else if(p >= alpha){print("Cannot reject the Null. Same variances across groups")}  
}
```

```
bartlett_test(weight~typchd69, wogs)
```

```
## [1] "Reject the Null. Variances are different across groups"
```

2c)

```
bonferroni = function(x, g){  
  test = pairwise.t.test(x, g, p.adjust.method = "bonf")  
  p = test$p.value  
  print(p)  
}
```

```
anovaTest = function(v1.v2, data){  
  test = anova(lm(v1.v2, data))  
  p = test$'Pr(>F)'[1]  
  if(p < alpha){print("Reject the Null. There's no difference of means")}  
  else if(p >= alpha){print("Cannot reject the Null.")}  
}
```

```
anovaTest(weight~typchd69, wogs)
```

```
## [1] "Reject the Null. There's no difference of means"
```

```
bonferroni(wogs$weight, wogs$typchd69)
```

```
##           0  1  2  
## 1 0.57352582 NA NA  
## 2 0.04337324  1 NA  
## 3 0.10284669  1  1
```

2d)

```
kruskal = function(v1.v2, data){  
  test = kruskal.test(v1.v2, data)  
  p = test$p.value  
  if(p < alpha){print("Reject the Null. There's no difference of medians")}  
  else if(p >= alpha){print("Cannot reject the Null.")}  
}
```

```
kruskal(weight~typchd69, wogs)
```

```
## [1] "Reject the Null. There's no difference of medians"
```

The Kruskal-Wallis test rejects the null. We apply the Bonf. adj. Wilcoxon signed rank test.

```
wilcox.rank = function(v1, v2, data){
  test = wilcox.test(v1, v2, p.adj = "bonf")
  p = test$p.value
  if(p < alpha){print("Reject the Null. The 4 groups have the same distributions.")}
  else if(p >= alpha){print("Cannot reject the Null.")}
}

wilcox.rank(wcgs$weight, wcgs$typchd69, wcgs)

## [1] "Reject the Null. The 4 groups have the same distributions."
```

Problem 3

3a) $H_0: m_0 = m_1$. Null Hypothesis: diet isn't effective. $H_a: m_0 \neq m_1$. Alt Hypothesis: diet is effective.

```
diet = read.delim("diets.txt", sep = ",", header = T)
diet1 = diet %>%
  filter(Diet == 1)
diet2 = diet %>%
  filter(Diet == 2)
diet3 = diet %>%
  filter(Diet == 3)
diet4 = diet %>%
  filter(Diet == 4)
```

3b)

Preliminary tests for ANOVA:

Normality test using Shapiro-Wilk

```
normality(diet1$WeightLoss)
```

```
## [1] "Do not reject the Null. Normality is met"
```

```
normality(diet2$WeightLoss)
```

```
## [1] "Do not reject the Null. Normality is met"
```

```
normality(diet3$WeightLoss)
```

```
## [1] "Do not reject the Null. Normality is met"
```

```
normality(diet4$WeightLoss)
```

```
## [1] "Do not reject the Null. Normality is met"
```

Normality is met for all diet groups.

Testing for difference of variances, means, medians & distributions:

```
bartlett_test(WeightLoss~Diet, diet)
```

```
## [1] "Cannot reject the Null. Same variances across groups"
```

```
anovaTest(WeightLoss~Diet, diet)
```

```
## [1] "Reject the Null. There's no difference of means"
```

```
kruskal(WeightLoss~Diet, diet)
```

```
## [1] "Reject the Null. There's no difference of medians"
```

```
wilcox.rank(diet$WeightLoss, diet$Diet, diet)
```

```
## [1] "Reject the Null. The 4 groups have the same distributions."
```

3d)

```
bonferroni(diet$WeightLoss, diet$Diet)
```

```
##           1           2           3
## 2 1.000000e+00          NA          NA
## 3 3.896674e-02 0.0017768225          NA
## 4 9.544429e-05 0.0004538353 5.770801e-13
```

We reject the null since all p-values < alpha.

3f)

```
diet.aov = aov(WeightLoss~Diet, diet)
summary(diet.aov)
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Diet         1    1641   1641.0    13.44 0.000374 ***
## Residuals   116   14168    122.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p.value after using aov() is ~0.0003 < alpha. Thus we reject H0.

3g)

```
t = pairwise.t.test(diet$WeightLoss, diet$Diet)
t$p.value
```

```
##           1           2           3
## 2 5.026820e-01          NA          NA
## 3 1.298891e-02 0.0008884112          NA
## 4 7.953691e-05 0.0003025569 5.770801e-13
```

We reject the null since all p-values < alpha.

Problem 4

```
run = "aabbaaaabbbbaababbaaabbbbababbbabbbb"

letters = strsplit(run, "")[[1]]
count = table(letters)
A = count[1][1]
B = count[2][1]

i = 1
runs = 1
len = length(letters)

while (i < (len - 1)){
  if (letters[i] != letters[i+1]) {runs = runs + 1}
  i = i + 1
}
runs
```

```
## [1] 16
```

The runs is even so we can use the even probability function.

```
prob = (2*(choose(A-1,runs/2 -1))*choose(B-1, runs/2 - 1))/choose(A+B, A)
prob
```

```
## [1] 0.1064868
```

```
if (prob > alpha){print("We can assume randomness")}
```

```
## [1] "We can assume randomness"
```