HW2_STA104

Filip Wilhelm Sjostrand

2022-10-18

Appendix

Code: Question 2 & 3.

```
# Get packages ----------
library(dplyr)
library(tidyr)

# Create data frame of 6C3 ----------
t1 <- c(10, 15, 50)
t2 <- c(12, 17, 19)
obs <- append(t1, t2)

df <- data.frame(combn(obs, 3))
df <- as.data.frame(t(df))

# Find mean difference and treatment sum ----------
perm <- df %>%
  mutate(sumTreatment1=V1+V2+V3) %>%
  mutate(sumother=123-sumTreatment1) %>%
  mutate(mean1=sumTreatment1/3, mean2=sumother/3) %>%
  mutate(diff=mean1-mean2) %>%
  select(V1, V2, V3, diff, sumTreatment1)

name <- rownames(perm)

# Add the rest of the observations as second treatment group ----------
df2 <- data.frame()

for (row in 1:nrow(perm)) {
  vec <- obs

  v1 <- perm[row, "V1"]
  v2 <- perm[row, "V2"]
  v3 <- perm[row, "V3"]

  df2 <- df2 %>% rbind(vec[! vec %in% c(v1, v2, v3)])
}

df2 <- df2 %>% select(V4=X12, V5=X17, V6=X19)
```

```
perm <- perm %>%
  cbind(df2)

# Create final table ----------
perm <- perm %>%
  rowwise() %>%
  mutate(
    median1 =  median(c(V1, V2, V3)),
    median2 =  median(c(V4, V5, V6))
  ) %>%
  mutate(medianDiff = median1 - median2) %>%
  select(V1, V2, V3, meanDiff=diff, sumTreatment1, medianDiff)

perm <- data.frame(perm)
rownames(perm) <- name
rownames(perm)[rownames(perm) == "X1"] <- "X1*****" # Observed sample

perm <- perm %>% arrange(desc(meanDiff))
```

Table: Question 2 & 3.

	V1	V2	V3	meanDiff	sumTreatment1	medianDiff
X19	50	17	19	16.33333	86	7
X13	15	50	19	15.00000	84	7
X12	15	50	17	13.66667	82	5
X18	50	12	19	13.00000	81	4
X7	10	50	19	11.66667	79	4
X17	50	12	17	11.66667	79	2
X6	10	50	17	10.33333	77	2
X11	15	50	12	10.33333	77	-2
X1*****	10	15	50	9.00000	75	-2
X5	10	50	12	7.00000	72	-5
X16	15	17	19	-7.00000	51	5
X20	12	17	19	-9.00000	48	2
X10	10	17	19	-10.33333	46	2
X15	15	12	19	-10.33333	46	-2
X4	10	15	19	-11.66667	44	-2
X14	15	12	17	-11.66667	44	-4
X3	10	15	17	-13.00000	42	-4
X9	10	12	19	-13.66667	41	-5
X8	10	12	17	-15.00000	39	-7
X2	10	15	12	-16.33333	37	-7

Code: Question 7 b).

```
# Two sample t-test ----------
rural <- c(3,2,1,1,2,1,3,2,2,2,2,5,1,4,1,1,1,1,6,2,2,2,1,1)
```

```r
urban <- c(1,0,1,1,0,0,1,1,1,8,1,1,1,0,1,1,2)

t.test(rural, urban, alternative = c("greater"))
```

```
##
##  Welch Two Sample t-test
##
## data:  rural and urban
## t = 1.554, df = 27.699, p-value = 0.06577
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -0.07669184         Inf
## sample estimates:
## mean of x mean of y
##  2.041667  1.235294
```

Code: Question 8.

```r
#simulation with b permutations randomly selected.
outcome=c(rural, urban)
b=100000
treat=c(rep(1,length(rural)),rep(2,length(urban)))
diffobs=mean(rural)-mean(urban)
d=c()
p=c()
for(i in 1:b){
permut=sample(outcome)
d[i]=mean(permut[treat==1])-mean(permut[treat==2])
p[i]=(d[i]>=diffobs)+0
}
pvalue=sum(p)/b
```

Code: Question 10.

```r
# Create data frame of VW scores ----------
s1 <- c(5, 11, 16, 8, 12)
s2 <- c(17, 14, 15, 21, 19, 13)
data <- c(s1, s2)

df <- data.frame(data)

section <- c(rep(1, length(s1)), rep(2, length(s2)))

VW <- c(-1.383, -0.967, -0.674, -0.431, -0.210, 0, 0.210, 0.431, 0.674, 0.967, 1.383)


V <- sum(
  df %>%
  cbind(section) %>%
```

```
  mutate(rank=rank(data)) %>%
  arrange(rank) %>%
  cbind(VW) %>%
  filter(section==1) %>%
  select(VW)
  )

# Create dictionary of value:VW pairs ----------
dictdata <- df %>%
  cbind(section) %>%
  mutate(rank=rank(data)) %>%
  arrange(rank) %>%
  cbind(VW) %>%
  select(data, VW)

dict <- list()
for(i in 1:nrow(dictdata)){
  dict[[as.character(dictdata[i,1])]] <- dictdata[i,2]
}


# Create data frame of 11C5 ----------
df <- data.frame(combn(data, 5))

df <- as.data.frame(t(df))

df["VW"] <- 0 # Empty vector to fill

# Add VW score for each permutation group -----------
for(row in 1:nrow(df)){
  for(col in 1:(ncol(df)-1)){
    df[row, "VW"] <- df[row, "VW"] + as.numeric(unlist(dict[as.character(df[row,col])]))
  }
}

rownames(df)[rownames(df)=="X1"] <- "X1******" # Mark our observed value

# Observed less than or equal to V ----------
D <- nrow(
  df %>%
  filter(VW <= V)
  )

tot <- nrow(df)

pval <- D/tot

# Order df for table ----------
ordDf <- df %>% arrange(VW)
```

Table: Question 10.

*Only the head of data frame given amount of permutations

	V1	V2	V3	V4	V5	VW
X34	5	11	8	12	13	-3.665
X30	5	11	8	12	14	-3.455
X31	5	11	8	12	15	-3.245
X43	5	11	8	14	13	-3.234
X1******	5	11	16	8	12	-3.024
X46	5	11	8	15	13	-3.024

Code: Question 12.

```
# Create data frame of pairwise differences ----------
s1 <- c(5, 11, 16, 8, 12)
s2 <- c(17, 14, 15, 21, 19, 13)

diffs <- c()
for(row in 1:length(s2)){
  for(col in 1:length(s1)){
    diff <- s1[col] - s2[row]
    diffs <- append(diffs, diff)
  }
}

diffs <- sort(diffs)

# Table for visual aid ----------
pairwise <- data.frame("5"=rep(0, length(s2)),
                       "11"=rep(0, length(s2)),
                       "16"=rep(0, length(s2)),
                       "8"=rep(0, length(s2)),
                       "12"=rep(0, length(s2)))

rownames(pairwise) <- as.character(s2)

for(row in 1:length(s2)){
  for(col in 1:length(s1)){
    pairwise[row, col] <- s1[col] - s2[row]
  }
}

# Find ka and kb ----------
df <- data.frame(diffs)

rank <- 1:nrow(df)

limits <- df %>%
  cbind(rank) %>%
  filter(rank==(3+1) | rank==27) # From A4 table
```

```
rownames(limits) <- c("Lower", "Upper")

hl <- median(diffs)
```

Tables: Question 12.

Pairwise Comparison

	X5	X11	X16	X8	X12
17	-12	-6	-1	-9	-5
14	-9	-3	2	-6	-2
15	-10	-4	1	-7	-3
21	-16	-10	-5	-13	-9
19	-14	-8	-3	-11	-7
13	-8	-2	3	-5	-1

Confidence Interval Limits

	diffs	rank
Lower	-12	4
Upper	-1	27

Code: Question 18.

```
exp <- c(11,33,48,34,112,369,64,44)
con <- c(177,80,141,332)




# ---------- Wilcoxon ----------
p <- wilcox.test(exp, con, alternative = ("less"))$p.value
W <- wilcox.test(exp, con, alternative = ("less"))$statistic + 8*9/2




# ---------- Van der Waerden ----------
# Get V1 ----------
data <- c(exp, con)

df <- data.frame(data)

group <- c(rep("Experiment", length(exp)), rep("Control", length(con)))

VW <- c(-1.426, -1.020, -0.736, -0.502, -0.293, -0.097, 0.097, 0.293, 0.502, 0.736, 1.020, 1.426)

V <- sum(
```

```
  df %>%
  cbind(group) %>%
  mutate(rank=rank(data)) %>%
  arrange(rank) %>%
  cbind(VW) %>%
  filter(group=="Experiment") %>%
  select(VW)
  )

# Create dictionary of value:VW pairs ----------
dictdata <- df %>%
  cbind(group) %>%
  mutate(rank=rank(data)) %>%
  arrange(rank) %>%
  cbind(VW) %>%
  select(data, VW)

dict <- list()
for(i in 1:nrow(dictdata)){
  dict[[as.character(dictdata[i,1])]] <- dictdata[i,2]
}


# Savage scores dict ----------
N <- nrow(df)

savage <- c(1/N)
for(i in 2:N){
  savage[i] <- savage[i-1] + 1/(N-(i-1))
}

savDictData <- df %>%
  cbind(group) %>%
  mutate(rank=rank(data)) %>%
  arrange(rank) %>%
  cbind(savage) %>%
  select(data, savage)

dictSavage <- list()
for(i in 1:nrow(savDictData)){
  dictSavage[[as.character(savDictData[i,1])]] <- savDictData[i,2]
}

# Create data frame of 12C8 ----------
df <- data.frame(combn(data, 8))

df <- as.data.frame(t(df))

df["VW"] <- 0 # Empty vector to fill

# Add VW score for each permutation group -----------
for(row in 1:nrow(df)){
  for(col in 1:(ncol(df)-1)){
```

```
    df[row, "VW"] <- df[row, "VW"] + as.numeric(unlist(dict[as.character(df[row,col])]))
  }
}

rownames(df)[rownames(df)=="X1"] <- "X1******" # Mark our observed value

# Observed less than or equal to V ----------
D <- nrow(
  df %>%
  filter(VW <= V)
  )

tot <- nrow(df)

pval <- D/tot



# ---------- Exponential ----------
# Add savage score to existing df ---------
df["sav"] <- 0

for(row in 1:nrow(df)){
  for(col in 1:(ncol(df)-2)){
    df[row, "sav"] <- df[row, "sav"] + as.numeric(unlist(dict[as.character(df[row,col])]))
  }
}

savObs <- as.numeric(df %>%
  filter(row.names(df) == "X1******") %>%
  select(sav))

dSav <- nrow(
  df %>%
  filter(sav <= savObs )
  )

total <- nrow(df)

p <- dSav/total


# ---------- Permutation ----------
Dperm <- nrow(df %>%
  select(1:8) %>%
  mutate(sumOfExperiment=V1+V2+V3+V4+V5+V6+V7+V8) %>%
  arrange(sumOfExperiment) %>%
  filter(sumOfExperiment <= 715))

pvaluePerm <- Dperm/total
```