

Homework 4

20307100013 蒋翌坤

Solution for Problem 1

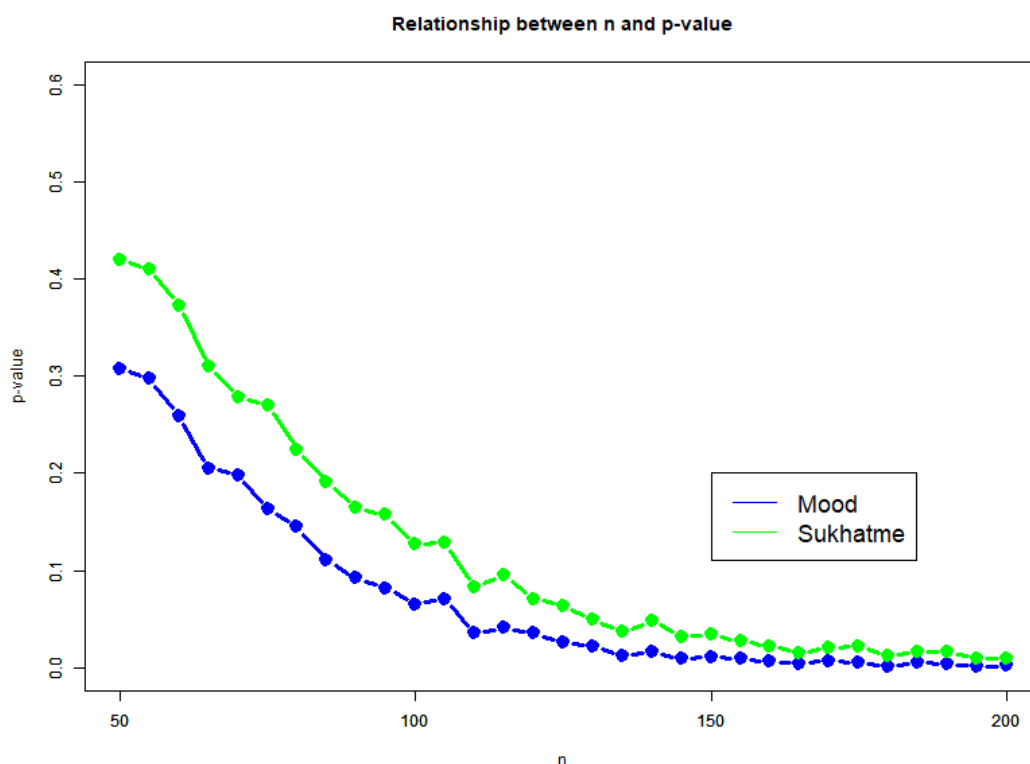
1.1

In Mood and Sukhatme test, we use normal approximation to calculate p-value.

We use random generator to generate $n = 100$ $X_1, \dots, X_n \sim N(0,1), Y_1, \dots, Y_n \sim N(0,1.2)$. Simulating for 10,000 times, we get 8,805 times not to reject the null hypothesis from Mood test, 8,952 times not to reject the null hypothesis from Sukhatme test. Therefore, we **cannot reject** the null hypothesis $H_0: F_X = F_Y$ at $\alpha = 0.05$ from both Mood test and Sukhatme test.

1.2

Consider the following scenario: X and Y are from normal distribution. Without loss of generality, assume $X_1, \dots, X_n \sim N(0,1), Y_1, \dots, Y_n \sim N(0,1.5^2)$. We take n samples from X and Y respectively. Simulating the sampling and test process for 1,000 times for some n , we get a p-value corresponding to n from Mood test and Sukhatme test shown in the following graph.

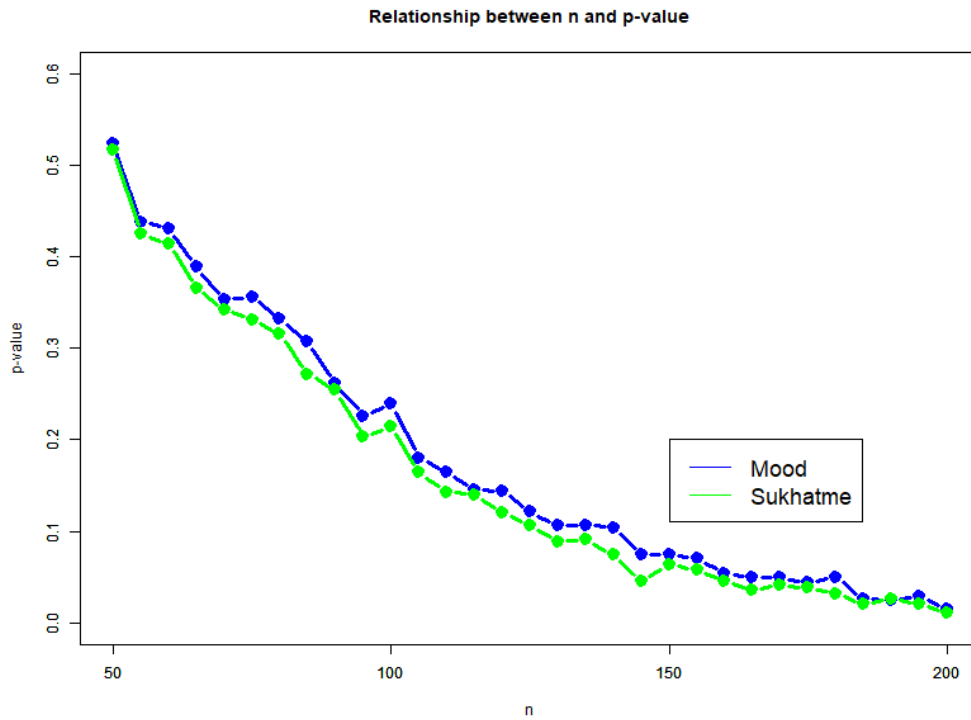


Compared to Sukhatme test, Mood test can reject the null hypothesis under the same α level from a smaller sample size in this scenario. Therefore, Mood test is more powerful than Sukhatme test in this scenario.

1.3

Consider the following scenario: X and Y are from Cauchy distribution. Without loss of generality, assume $X_1, \dots, X_n \sim \text{Cauchy}(0,1), Y_1, \dots, Y_n \sim \text{Cauchy}(0,1.8)$. We take n samples from

X and Y respectively. Simulating the sampling and test process for 1,000 times for some n , we get a p-value corresponding to n from Mood test and Sukhatme test shown in the following graph.



Compared to Mood test, Sukhatme test can reject the null hypothesis under the same α level from a smaller sample size in this scenario. Therefore, Sukhatme test is more powerful than Mood test in this scenario.

Solution for Problem 2

Suppose X and Y follows the same distribution F_0 . Since the median of the distribution is 0, then $F_0(0) = \frac{1}{2}$.

$$\begin{aligned}
 P(Y < X < 0) \cup P(0 < X < Y) &= \int_{-\infty}^0 \int_{-\infty}^x f_0(y) f_0(x) dy dx + \int_0^{\infty} \int_x^{\infty} f_0(y) f_0(x) dy dx \\
 &= \int_{-\infty}^0 F_0(x) dF_0(x) + \int_0^{\infty} (1 - F_0(x)) dF_0(x) \\
 &= \int_0^{\frac{1}{2}} u du + \int_{\frac{1}{2}}^1 (1 - u) du, \text{ where } u = F_0(x) \\
 &= \frac{1}{8} + \frac{1}{8} \\
 &= \frac{1}{4}
 \end{aligned}$$

Therefore, $P(Y < X < 0) \cup P(0 < X < Y) = \frac{1}{4}$.

Solution for Problem 3

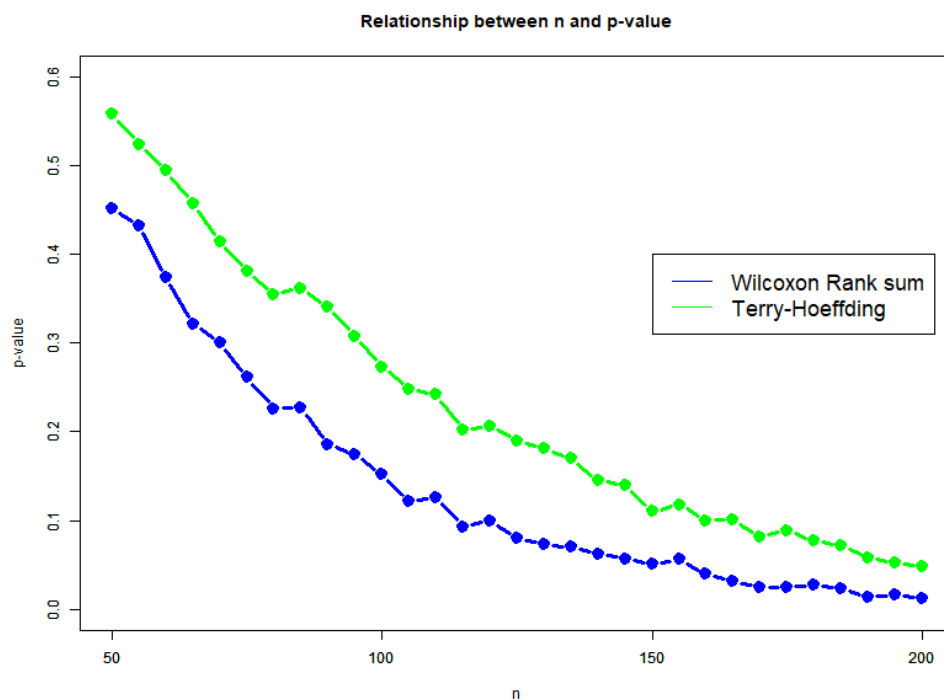
3.1

We use first approximation to approximate $E(\xi_{(i)}), E(\xi_{(i)}) \approx \Phi^{-1}(\frac{i}{N+1})$

We use random generator to generate $n = 100$ $X_1, \dots, X_n \sim N(0,1), Y_1, \dots, Y_n \sim N(0.1,1)$. Simulating for 10,000 times, we get 8,942 times not to reject the null hypothesis from Wilcoxon Rank-sum test, 8,931 times not to reject the null hypothesis from Terry-Hoeffding test. Therefore, we **cannot reject** the null hypothesis $H_0: F_X = F_Y$ at $\alpha = 0.05$ from both Wilcoxon Rank-sum test and Terry-Hoeffding test.

3.2

Consider the following scenario: X and Y are from Cauchy distribution. Without loss of generality, assume $X_1, \dots, X_n \sim \text{Cauchy}(0,1), Y_1, \dots, Y_n \sim \text{Cauchy}(0.8,1)$. We take n samples from X and Y respectively. Simulating the sampling and test process for 1,000 times for some n , we get a p-value corresponding to n from Wilcoxon Rank-sum test and Terry-Hoeffding test shown in the following graph.

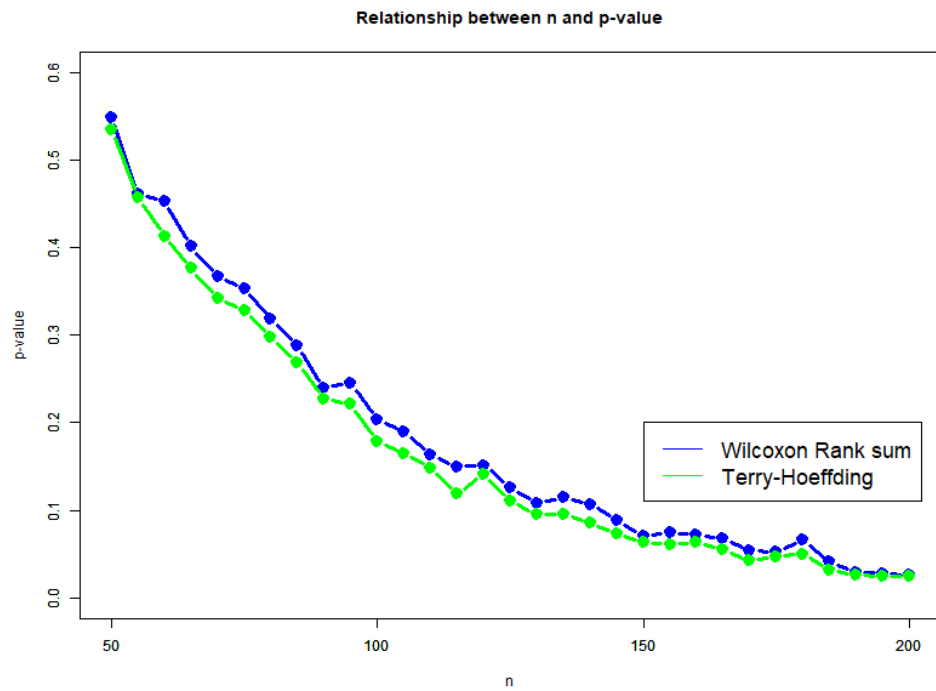


Compared to Terry-Hoeffding test, Wilcoxon Rank-sum test can reject the null hypothesis under the same α level from a smaller sample size in this scenario. Therefore, Wilcoxon Rank-sum test is more powerful than Terry-Hoeffding test in this scenario.

3.3

Consider the following scenario: X and Y are from normal distribution. Without loss of generality, assume $X_1, \dots, X_n \sim N(0,1), Y_1, \dots, Y_n \sim N(0.4,1)$. We take n samples from X and Y respectively. Simulating the sampling and test process for 1,000 times for some n , we get a p-value corresponding to n from Wilcoxon Rank-sum test and Terry-Hoeffding test shown in the

following graph.



Compared to Wilcoxon Rank-sum test, Terry-Hoeffding test can reject the null hypothesis under the same α level from a smaller sample size in this scenario. Therefore, Terry-Hoeffding test is more powerful than Wilcoxon Rank-sum test in this scenario.

Appendix

The following provides R code I used to complete this homework. You can also find the R code and output at <https://thisiskunmeng.github.io/nonparametric/hw4.html>

R code for Solution for Problem 1

```
#' rewrite Mood test
#' return True if null hypothesis is not rejected, else return False
mood_test <- function(x, y, a) {
  xy <- c(x, y)
  N <- length(x) + length(y)
  xy_sorted <- sort(xy)
  z <- rep(0, times = N)
  z[match(x, xy_sorted)] <- 1
  m_n <- 0
  for (i in 1:N) {
    m_n <- m_n + (i - (N + 1) / 2)^2 * z[i]
  }
  e_m_n <- length(x) * (N^2 - 1) / 12
  var_m_n <- length(x) * length(y) * (N + 1) * (N^2 - 4) / 180
  z_value <- (m_n - e_m_n) / sqrt(var_m_n)
  z_a <- qnorm(1 - a / 2)
  return(abs(z_value) <= z_a)
}

#' rewrite Sukhatme test
#' return True if null hypothesis is not rejected, else return False
sukhatme_test <- function(x, y, a) {
  t <- 0
  for (i in x) {
    for (j in y) {
      if ((i > j && i < 0) || (i < j && i > 0)) {
        t <- t + 1
      }
    }
  }
  e_t <- length(x) * length(y) / 4
  var_t <- (length(x) * length(y) * (length(x) + length(y) + 7)) / 48
  z_value <- (t - e_t) / sqrt(var_t)
  z_a <- qnorm(1 - a / 2)
  return(abs(z_value) <= z_a)
}

# 1.1
sim_num <- 10000
n <- 100
a <- 0.05
x_mean <- 0
y_mean <- 0
var_x <- 1
var_y <- 1.2
mood <- c()
suk <- c()
for (i in 1:sim_num) {
  set.seed(i)
  x <- rnorm(n, mean = x_mean, sd = sqrt(var_x))
  y <- rnorm(n, mean = y_mean, sd = sqrt(var_y))
  mood <- c(mood, mood_test(x, y, a))
}
```

```

    suk <- c(suk, sukhatme_test(x, y, a))
  }
  print(sum(mood)) # 8805
  print(sum(suk)) # 8952
# 1.2
n = seq(from = 50, to = 200, by = 5)
sim_num = 1000
mood_n = c()
suk_n = c()
for (i in n){
  mood <- c()
  suk <- c()
  for (j in 1:sim_num) {
    set.seed(i*j)
    x <- rnorm(i, 0, 1)
    y <- rnorm(i, 0, 1.5)
    mood <- c(mood, mood_test(x, y, a))
    suk <- c(suk, sukhatme_test(x, y, a))
  }
  mood_n = c(mood_n, sum(mood)/sim_num)
  suk_n = c(suk_n, sum(suk)/sim_num)
}
png("hw4-1.2.png", width=800, height=600)
plot(n, mood_n,
     type = "b", pch = 19, xlab = "n", ylab = "p-value",
     col = "blue", ylim = c(0,0.6),
     main = "Relationship between n and p-value",
     lty = 1, lwd = 3, cex = 1.5
)
par(new=TRUE)
plot(n, suk_n,
     type = "b", pch = 19, col = "green", xlab = "", ylab = "",
     ylim = c(0,0.6),
     lty = 1, lwd = 3, cex = 1.5)
legend(150,0.2, legend=c("Mood", "Sukhatme"),
      col = c("blue", "green"), lty=c(1, 1), cex=1.5)
dev.off()
# 1.3
n = seq(from = 50, to = 200, by = 5)
sim_num = 1000
mood_n = c()
suk_n = c()
for (i in n){
  mood <- c()
  suk <- c()
  for (j in 1:sim_num) {
    set.seed(i*j)
    x <- rcauchy(i, 0, 1)
    y <- rcauchy(i, 0, 1.8)
    mood <- c(mood, mood_test(x, y, a))
    suk <- c(suk, sukhatme_test(x, y, a))
  }
  mood_n = c(mood_n, sum(mood)/sim_num)
  suk_n = c(suk_n, sum(suk)/sim_num)
}
png("hw4-1.3.png", width=800, height=600)
plot(n, mood_n,

```

```

    type = "b", pch = 19, xlab = "n", ylab = "p-value", col = "blue",
    ylim = c(0,0.6),
    main = "Relationship between n and p-value",
    lty = 1, lwd = 3, cex = 1.5
  )
  par(new=TRUE)
  plot(n, suk_n,
    type = "b", pch = 19, col = "green", xlab = "", ylab = "", ylim =
    c(0,0.6),
    lty = 1, lwd = 3, cex = 1.5)
  legend(150,0.2, legend=c("Mood", "Sukhatme"), col = c("blue", "green"),
  lty=c(1, 1), cex=1.5)
  dev.off()

```

R code for Solution for Problem 3

```

#' rewrite Wilcoxon Rank-sum test
#' return True if null hypothesis is not rejected, else return False
wilcox_ranksum_test <- function(x, y, a) {
  xy <- c(x, y)
  N <- length(x) + length(y)
  xy_sorted <- sort(xy)
  z <- rep(0, times = N)
  z[match(x, xy_sorted)] <- 1
  w_n <- 0
  for (i in 1:N) {
    w_n <- w_n + i * z[i]
  }
  e_w_n <- length(x) * (N + 1) / 2
  var_w_n <- length(x) * length(y) * (N + 1) / 12
  z_value <- (w_n - e_w_n) / sqrt(var_w_n)
  z_a <- qnorm(1 - a / 2)
  return(abs(z_value) <= z_a)
}

#' rewrite Terry-Hoeffding test
#' return True if null hypothesis is not rejected, else return False
terry_hoeffding_test <- function(x, y, a) {
  xy <- c(x, y)
  N <- length(x) + length(y)
  xy_sorted <- sort(xy)
  z <- rep(0, times = N)
  z[match(x, xy_sorted)] <- 1
  c1 <- 0
  e_i_all <- c()
  for (i in 1:N) {
    e_i <- qnorm(i / (N + 1))
    e_i_all <- c(e_i_all, e_i)
    c1 <- c1 + e_i * z[i]
  }
  e_c1 <- 0
  var_c1 <- length(x) * length(y) * sum((e_i_all)^2) / ((N + 1) * N)
  z_value <- (c1 - e_c1) / sqrt(var_c1)
  z_a <- qnorm(1 - a / 2)
  return(abs(z_value) <= z_a)
}

# 3.1
sim_num <- 10000
n <- 100
a <- 0.05

```

```

x_mean <- 0
y_mean <- 0.1
var_x <- 1
var_y <- 1
wilcox <- c()
terry <- c()
for (i in 1:sim_num) {
  set.seed(i)
  x <- rnorm(n, mean = x_mean, sd = sqrt(var_x))
  y <- rnorm(n, mean = y_mean, sd = sqrt(var_y))
  wilcox <- c(wilcox, wilcox_ranksum_test(x, y, a))
  terry <- c(terry, terry_hoeffding_test(x, y, a))
}
print(sum(wilcox)) # 8942
print(sum(terry)) # 8931
# 3.2
n = seq(from = 50, to = 200, by = 5)
sim_num = 1000
wilcox_n = c()
terry_n = c()
for (i in n){
  wilcox <- c()
  terry <- c()
  for (j in 1:sim_num) {
    set.seed(i*j)
    x <- rcauchy(i, 0, 1)
    y <- rcauchy(i, 0.8, 1)
    wilcox <- c(wilcox, wilcox_ranksum_test(x, y, a))
    terry <- c(terry, terry_hoeffding_test(x, y, a))
  }
  wilcox_n = c(wilcox_n, sum(wilcox)/sim_num)
  terry_n = c(terry_n, sum(terry)/sim_num)
}
png("hw4-3.2.png", width=800, height=600)
plot(n, wilcox_n,
  type = "b", pch = 19, xlab = "n", ylab = "p-value",
  col = "blue", ylim = c(0,0.6),
  main = "Relationship between n and p-value",
  lty = 1, lwd = 3, cex = 1.5
)
par(new=TRUE)
plot(n, terry_n,
  type = "b", pch = 19, col = "green", xlab = "", ylab = "",
  ylim = c(0,0.6),
  lty = 1, lwd = 3, cex = 1.5)
legend(150,0.4, legend=c("Wilcoxon Rank sum", "Terry-Hoeffding"),
  col = c("blue", "green"), lty=c(1, 1), cex=1.5)
dev.off()
# 3.3
n = seq(from = 50, to = 200, by = 5)
sim_num = 1000
wilcox_n = c()
terry_n = c()
for (i in n){
  wilcox <- c()
  terry <- c()
  for (j in 1:sim_num) {
    set.seed(i*j)

```



```

        x <- rnorm(i, 0, 1)
        y <- rnorm(i, 0.4, 1)
        wilcox <- c(wilcox, wilcox_ranksum_test(x, y, a))
        terry <- c(terry, terry_hoeffding_test(x, y, a))
    }
    wilcox_n = c(wilcox_n, sum(wilcox)/sim_num)
    terry_n = c(terry_n, sum(terry)/sim_num)
}
png("hw4-3.3.png", width=800, height=600)
plot(n, wilcox_n,
     type = "b", pch = 19, xlab = "n", ylab = "p-value",
     col = "blue", ylim = c(0,0.6),
     main = "Relationship between n and p-value",
     lty = 1, lwd = 3, cex = 1.5
)
par(new=TRUE)
plot(n, terry_n,
     type = "b", pch = 19, col = "green", xlab = "", ylab = "",
     ylim = c(0,0.6),
     lty = 1, lwd = 3, cex = 1.5)
legend(150,0.2, legend=c("Wilcoxon Rank sum", "Terry-Hoeffding"),
      col = c("blue", "green"), lty=c(1, 1), cex=1.5)
dev.off()

```