Homework 3

20307100013 蒋翌坤

Solution for Problem 1

1.1

It is required to calculate $A(m,n)$ when using Kolmogorov-Smirnov Two-Sample Test. However, $A(m,n)$ involves factorio calculation. It becomes very complex when n is large. Since $n = 100$ is large enough, we use asymptotic distribution $\lim_{n\to\infty} P\left(\sqrt{\frac{n}{2}} D_{n,n}^{+} \le d\right) = 1 - e^{-2d^2}$ to determine whether to reject the null hypothesis. That is if $\sqrt{\frac{n}{2}} D_{n,n}^{+} \le d_\alpha$, where critical value $d_\alpha = -\log(\alpha)/2$, then we cannot reject the null hypothesis $H_0: F_X = F_Y$ at $\alpha = 0.05$.

We use random generator to generate $n = 1000$ $X_1, \dots, X_n \sim N(0,1), Y_1, \dots, Y_n \sim t_{10}$. Simulating for $10{,}000$ times, we get $9{,}685$ times not to reject the null hypothesis. Therefore, we **cannot reject** the null hypothesis $H_0: F_X = F_Y$ at $\alpha = 0.05$.

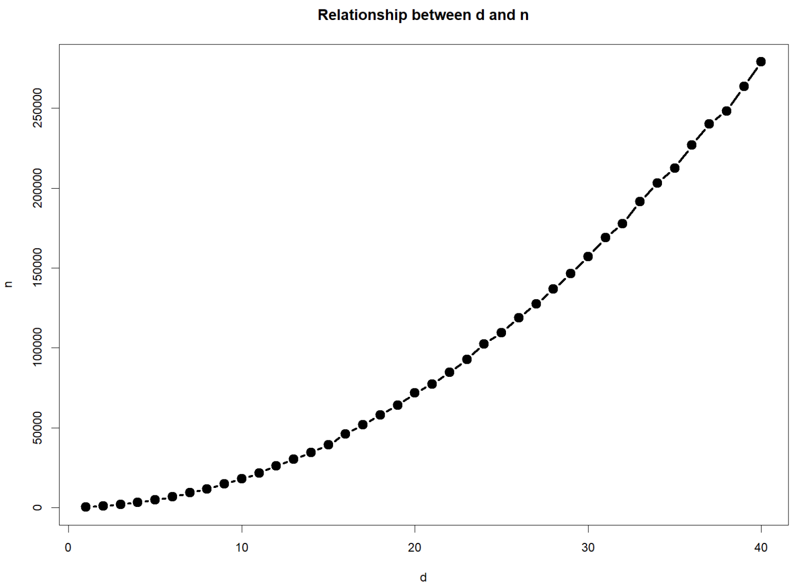
1.2

Using the asymptotic distribution above to conduct Kolmogorov-Smirnov Two-Sample Test, we can determine whether to reject the null hypothesis for every n. Since null hypothesis tends to be rejected at a larger n, we use bisection method to speed up the estimation process.

From simulation, we can get that n must be on average at least $17{,}743$ for $H_0: F_X = F_Y$ to be rejected at $\alpha = 0.05$.

1.3

We know that when $d \to \infty, t_d \xrightarrow{D} N(0,1)$. Therefore, required n such that $H_0: F_X = F_Y$ is rejected will grow larger when d grows larger. Using the same simulation method from 1.2, we can get the following graph. It illustrates their positive relationship when $d \le 40$.



Solution for Problem 2

2.1

Since $n = 30$ is large enough, we use $Z = \frac{U - n^2/2}{\sqrt{n^2(2n+1)/12}} \sim N(0,1)$ as our test statistic. A specific p-value can be calculated.

We use random generator to generate $n = 30$ $X_1, \dots, X_n \sim N(0,1), Y_1, \dots, Y_n \sim N(0,2^2)$. Simulating for $10{,}000$ times, we get $1{,}144$ times to reject the null hypothesis. Therefore, we **cannot reject** the hypothesis $H_0: F_X = F_Y$ at $\alpha = 0.1$.

2.2

Using the same simulation method above, we get $5{,}807$ times to reject the null hypothesis Therefore, we **cannot reject** the hypothesis $H_0: F_X = F_Y$ at $\alpha = 0.1$.

Solution for Problem 3

We calculate the test statistic and get $T = 2\sum_{i=1}^{9} s_i r(|X_i - 14|) - \frac{10\times 9}{2} = 45$. At $\alpha = 0.1$, from tabulation we can get critical value $t_{\alpha/2} = 5 < T$, assuming $H_1: M \neq 14$. Therefore, we **cannot reject** $H_0: M = 14$ at $\alpha = 0.1$.

Appendix

R code for Solution for Problem 1

```
# rewrite ks_test
ks_test <- function(x, y, a) {
    # return True or False. If False, then reject the null hypothesis.
    # use asymptotic distribution
    xe <- ecdf(x)
    ye <- ecdf(y)
    t <- c(x, y)
    d_mn <- max(abs(xe(t) - ye(t)))
    d <- -log(a) / 2 * sqrt(2 / length(x))
    return(d_mn <= d)
}

# 1.1
sim_num <- 10000
n <- 1000
p <- 0
for (i in 1:sim_num) {
    set.seed(i)
    x <- rnorm(n)
    y <- rt(n, df = 10)
    ks <- ks_test(x, y, 0.05)
    if (ks) {
        p <- p + 1
    }
}
cat("In 10000 simulations, number of times that we do not reject the
null hypothesis is", p, "\n") # 9685

# 1.2
t1 <- Sys.time()
sim_num <- 1000

# use bisection method for efficiency
# define useful functions
sim <- function(sim_num, n, n_min, n_max, d) {
    # for a certain n, determine number of times p that we do not reject
the null hypothesis in sim_num's simulations.
    p <- 0
    for (i in 1:sim_num) {
        set.seed((i + n_min + n_max) * d)
```

```
        x <- rnorm(n)
        y <- rt(n, df = d)
        ks <- ks_test(x, y, 0.05)
        if (ks) {
            p <- p + 1
        }
    }
    return(p)
}

check_nmax <- function(n_min, n_max, d) {
    cat("check n_max is indeed satisfy that null hypothesis is
rejected\n")
    while (TRUE) {
        cat("now n min is", n_min, "and n max is", n_max, "\n")
        p <- sim(sim_num, n_max, n_min, n_max, d)
        cat(p, "times to not reject\n")
        p_v <- p / sim_num
        if (p_v > 0.05) {
            n_min <- n_max
            n_max <- n_max + 1024
        } else {
            cat("n min is", n_min, "and n max is", n_max, "\n begin
bisection\n")
            break
        }
    }
    return(c(n_min, n_max))
}

find_n <- function(n_min, n_max, d, approx = 1) {
    # find the least n that reject null hypothesis given degree of
freedom of t distribution.
    while (TRUE) {
        n <- round((n_min + n_max) / 2)
        cat("now n min is", n_min, "and n max is", n_max, "\n")
        p <- sim(sim_num, n, n_min, n_max, d)
        p_v <- p / sim_num
        if (p_v <= 0.05) {
            n_max <- n
        } else {
            n_min <- n
        }
        if (n_max - n_min <= approx) {
```

```
            cat("n min is", n_min, "and n max is", n_max)
            break
        }
    }
    return(n_max)
}

n_min <- 1024
n_max <- n_min + 1024

tep <- check_nmax(n_min, n_max, 10)
n_min <- tep[1]
n_max <- tep[2]

n_10 <- find_n(n_min, n_max, 10)
cat("the least n is", n_10, "\n") # 17878

t2 <- Sys.time()
cat("run time is", t2 - t1, "\n") # 3.66 min

# 1.3
t3 <- Sys.time()
n_min <- 1
n_max <- 1024
r_n <- c()

for (d in 1:40) {
    # max d is too large, the code runs for a whole night.
    # d <20 is a considerable number to do simulation
    cat("\n\nnow d is", d, "\n")
    tep <- check_nmax(n_min, n_max, d)
    n_min <- tep[1]
    n_max <- tep[2]
    ap <- if (d > 15) 513 else 10 * d
    # since it is only for graph illustration, finding exact value is
not necessary
    n_d <- find_n(n_min, n_max, d, ap)
    r_n <- c(r_n, n_d)
}

t4 <- Sys.time()
print(t4 - t3) # 13 hours

png("hw3-1.3.png", width = 1600, height = 1200, res = 144)
```

```r
plot(1:40, r_n,
    type = "b", pch = 19, xlab = "d", ylab = "n",
    main = "Relationship between d and n",
    lty = 1, lwd = 3, cex = 1.5
)
dev.off()
```

R code for Solution for Problem 2

```r
# rewrite Mann_Whitney U test
mw_test <- function(x, y, n) {
    # return the p-value of Mann_Whitney U test, assume two-sized
    u <- 0
    for (y_i in y) {
        for (x_i in x) {
            if (x_i > y_i) {
                u <- u + 1
            }
        }
    }
    z <- (u - n^2 / 2) / sqrt(n^2 * (2 * n + 1) / 12)
    return(2 * (1 - pnorm(abs(z))))
}

# 2.1
n <- 30
a <- 0.1
sim_num <- 10000
p <- 0
for (i in 1:sim_num) {
    set.seed(i)
    x <- rnorm(n)
    y <- rnorm(n, sd = 2)
    mw <- mw_test(x, y, n)
    if (mw < a) {
        p <- p + 1
    }
}
cat("number of times to reject null hypothesis", p, "\n") # 1144

# 2.2
n <- 30
sim_num <- 10000
p <- 0
```

```r
for (i in 1:sim_num) {
    set.seed(i)
    x <- rnorm(n)
    y <- rnorm(n, mean = 0.5, sd = 1)
    mw <- mw_test(x, y, n)
    if (mw < a) {
        p <- p + 1
    }
}
cat("number of times to reject null hypothesis", p, "\n") # 5807
```

R code for Solution for Problem 3

```r
# rewrite Wilcoxon signed rank test
wilcox_test <- function(s, m) {
    # a = 0.1, assume two-sided, critical value t_a/2 = 5.
    # return True or False. If False, then reject the null hypothesis.
    # sample size is 9
    n <- 9
    d <- s - m
    abs_d <- abs(d)
    r_abs_d <- rank(abs(d))
    si <- 0
    for (i in 1:n) {
        if (abs_d[i] > 0) {
            si <- si + r_abs_d[i]
        }
    }
    t <- 2 * si - 10 * 9 / 2
    if (t > 5) {
        return(TRUE)
    } else {
        return(FALSE)
    }
}

sample <- c(1, 2, 2.5, 3.3, 10, 15, 15.5, 17, 20)
mw <- wilcox_test(sample, 14)

mw # TRUE
```