Stat 3005 – Nonparametric Statistics

Mid-term project

Time limit: 48 hours

Fall 2024

Instructions

1. The full mark of the project is 100.
2. There is an optional bonus question that is worth 10 marks. (E.g., if you get $95+10$, your project score is 100.)
3. Complete the project solely by yourself. No collaboration is allowed. You may consult lecture notes, tutorial notes, homework assignments, quizzes, textbooks, online materials, etc, however, you cannot solicit or obtain assistance from or provide assistance to other people in any form for any specific content on the project. Activities considered cheating include, but are not limited to, copying, rephrasing or modifying contents from websites, discussing project questions with other people, and asking for help on Internet forums. All cheating cases will be passed to the the Senate Committee on Student Discipline with no exception.
4. Generative AI tools (e.g., ChatGPT) can be used for auxiliary purposes so that the answers reflect your own intellectual ideas and independent effort. The graders have the ultimate power to decide the appropriateness and the level of originality of the submitted answers.
5. Quote the results that appear in the lecture notes if you wish to use them, e.g., write "by Theorem 3.1", "by the CLT for rank statistics", etc.
6. Prove the results that do not appear in lecture notes if you wish to use them.
7. Define your abbreviations clearly unless they have been defined in the lecture notes.
8. Submit the following two documents to Blackboard before the deadline.
   - Written part. Either (i) handwrite and scan your answers, or (ii) type your answers in LaTeX. Use A4-size papers. Compile your answers in one single `.pdf` file. Sign the Honor Code below, and attach this page as a cover of your submitted file. Name the document in the format `S3005_M_sid_name.pdf`. E.g., `S3005_M_1155001234_ChanKinWai.pdf`. Note that all plots, numerical answers, simulation results, etc must be included in the written part. Graders will not run your submitted codes to check the answers.
   - Computation part. Save your `R` codes in one single `.r` file and name it in the formate `S3005_M_sid_name.r`. E.g., `S3005_M_1155001234_ChanKinWai.r`.

   Submission without properly signed honor code will not be graded. Submissions with incorrect formats will not be graded. Unclear or blurred answers will not be graded. Late submission will not be graded. You may re-submit your answers as many time as you wish, however, only the last submission will be graded.

Honor Code

Please read the Honor Code below and sign your name.

> *The Chinese University of Hong Kong places very high importance on honesty in academic work submitted by students, and adopts a policy of zero tolerance on academic dishonesty. All work presented in this project, including ideas, discoveries, interpretations, codes, derivations, and conclusions, should solely reflect independent effort, and strictly adhere to the standard of academic integrity and the instructions above (see the academic honesty guide provided by CUHK for more details).*

I understand the above instructions, and affirm that all work conforms to the standards of the Honor Code.

Signature: 

Print Name: Lam Hoi Chun

Student ID: 1155192755

Date: 27 Oct 2024

Q1.1

We are interested in testing

$$H_0 : \sigma_1 = \sigma_2 = \cdots = \sigma_n$$

against the alternative

$$H_1 : \sigma_1 > \sigma_2 > \cdots > \sigma_n$$

where H_1 is the statement of Kimberley's suspection.

Q1.2

T is a simple linear rank statistic with

$$s(i) = \left| \frac{i}{n+1} - \frac{1}{2} \right| \quad \text{and} \quad c(i) = i$$

Accordingly we can compute

$$\begin{aligned}
\bar{c} &= \frac{1}{n} \sum_{i=1}^{n} i = \frac{1}{n} \cdot \frac{n(n+1)}{2} = \frac{n+1}{2} \\
\bar{s} &= \frac{1}{n} \sum_{i=1}^{n} \left| \frac{i}{n+1} - \frac{1}{2} \right| \\
&= \frac{1}{n} \sum_{i=1}^{(n+1)/2} \left(\frac{1}{2} - \frac{i}{n+1} \right) + \frac{1}{n} \sum_{i=(n+1)/2}^{n} \left(\frac{i}{n+1} - \frac{1}{2} \right) \\
&= \frac{1}{n} \cdot \frac{n-1}{8} + \frac{1}{n} \cdot \frac{n-1}{8} \\
&= \frac{n-1}{4n}
\end{aligned}$$

$$\bar{c^2} = \frac{1}{n} \sum_{i=1}^{n} i^2 = \frac{1}{n} \cdot \frac{n(n+1)(2n+1)}{6} = \frac{(n+1)(2n+1)}{6}$$

$$\sigma_c^2 = \bar{c^2} - (\bar{c})^2 = \frac{(n+1)(2n+1)}{6} - \left(\frac{n+1}{2} \right)^2 = \frac{n^2-1}{12}$$

$$\bar{s^2} = \frac{1}{n} \sum_{i=1}^{n} \left(\left| \frac{i}{n+1} - \frac{1}{2} \right| \right)^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{i}{n+1} - \frac{1}{2} \right)^2 = \frac{1}{n} \cdot \frac{n(n-1)}{12(n+1)} = \frac{n-1}{12(n+1)}$$

$$\sigma_s^2 = \bar{s^2} - (\bar{s})^2 = \frac{n-1}{12(n+1)} - \left(\frac{n-1}{4n} \right)^2 = \frac{(n-1)(n^3+3)}{48n^2(n+1)}$$

Q1.2 (Cont'd)

Under the null hypothesis, $X_{1:n}$ are continuous and IID by Theorem 3.1, so we have

$$\mathrm{E}(T) = n\bar{c}\bar{s} = n \cdot \frac{n+1}{2} \cdot \frac{n-1}{4n} = \frac{n^2-1}{8}$$
$$\mathrm{Var}(T) = \frac{n^2}{n-1}\sigma_s^2\sigma_c^2 = \left[\frac{n^2}{n-1}\right]\left[\frac{(n-1)(n^3+3}{48n^2(n+1)}\right]\left[\frac{n^2-1}{12}\right] = \frac{(n-1)(n^3+3)}{576}$$

Q1.3

We have to check the whether Assumption 3.1 holds

- (Data Condition) Under H_0, $X_{1:n}$ are CIID
- (Score Condition) The score generating function is $S(u) = u$ is not a constant and satisfies

$$\int_0^1 S^2(u)du = \int_0^1 u^2 du = \frac{1}{3}\left[u^3\right]_0^1 = \frac{1}{3} < \infty$$

- (Coefficient Condition) From Example 3.13 of the lecture notes, we have

$$\mathrm{UAN}_n = \frac{\max_{1\le i\le n}\left[c(i)-\bar{c}\right]^2}{\sum_{i=1}^n \left[c(i)-\bar{c}\right]^2} \to 0$$

We found that Assumption 3.1 holds, so under H_0, by Theorem 3.1, we have

$$t := \frac{T - \mathrm{E}(T)}{\mathrm{Var}(T)} \xrightarrow{d} \mathrm{N}(0,1)$$

Q1.4

Interpretation of T

Recall that T is a simple linear rank statistic, which

- (Coefficient) $c(i) = i$ can give weighting to each observation. Hence it handles n-sample problem.
- (Score) $s(i) = |\frac{i}{n+1} - \frac{1}{2}|$ is decreasing and allows it to measure the trend of the scale of n-sample.

So T measures the trend of the scale of the samples, which is sensible for Kimberley's suspection.

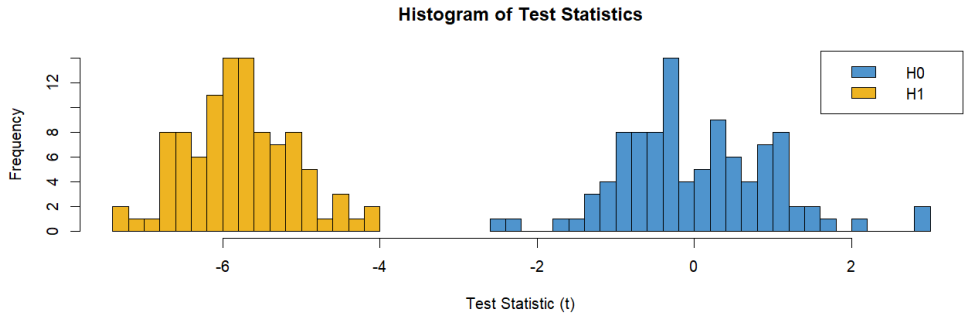
Q1.4 (Cont'd)

Stimulation

100 test statistics t are computed from H_0 and H_1, using the following R code:

```r
# 100 test statistics under H0
test_statistics_H0 <- numeric(num_samples)
for (i in 1:num_samples) {
  set.seed(i)
  x <- rnorm(n)
  test_statistics_H0[i] <- scaleTrendTest(x)
}

# 100 test statistics under H1
test_statistics_H1 <- numeric(num_samples)
for (i in 1:num_samples) {
  set.seed(i)
  x <- rnorm(n) * (n:1) / n
  test_statistics_H1[i] <- scaleTrendTest(x)
}
```

Then, a histogram for the test statistics t is produced to observe their distribution.



By observing the distribution, we can argue T is a sensible test

Under H_1 $\Rightarrow$ The data with larger index i have smaller variability
$\Rightarrow$ Contributions to the weighted absolute difference from higher-weighted terms are minimized
$\Rightarrow$ T is likely to be small relative to the null distribution
$\Rightarrow$ t is likely to be small relative to the null distribution

Testing Procedures

- By critical value, we reject H_0 iff t is small $\iff$ $t < \text{qnorm}(1 - \alpha_0)$
- By p-value, we can compute $\hat{p} = 1 - \text{pnorm}(t)$ and reject H_0 at size α_0 if $\hat{p} < \alpha_0$

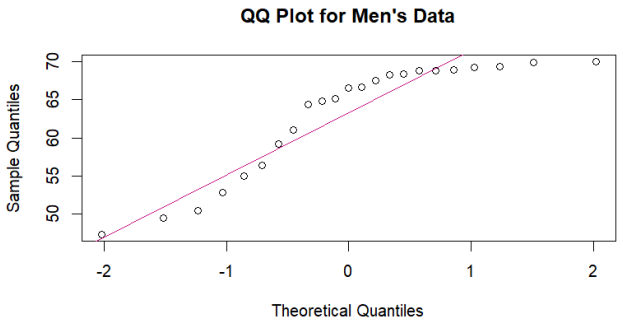
Q2

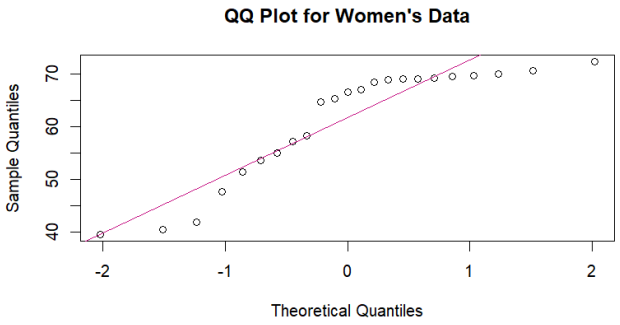
The email for Kathy:

Dear Kathy,

I am writing to reply to your inquiry regarding the statistical analysis.

Firstly, the t-test performed may be inaccurate since there is bias in data selection. Choosing the data from 1976 due to women's stronger performance may not represent the overall trend. Besides, the data of men's and women's performance are not normally distributed (please see the QQ Plot below), leading to the t-test's inaccuracy.





Therefore, a 2-sample location test (rank-sum test) is suggested. We use men's and women's data from 1928 (when both data exist) as samples 1 (x) and 2 (y) respectively.

Model assumption:

$$\begin{cases} x = \mu_1 + \varepsilon_i \\ y = \mu_2 + \varepsilon_i \end{cases}$$, where ε_i are continuous, independent and identically distributed.

Then, we test $H_0 : \mu_1 = \mu_2$ vs $H_1 : \mu_1 > \mu_2$ using wilcox.test() function in R, by specifying the "alternative" is "greater", then p-value ≈ 0.57 can be computed, which indicates that there is weak evidence against the H_0, which implies that we cannot conclude that males have an advantage in discus throw, and it aligns with your suspection.

I hope I have been of some help in addressing your questions. Please let me know if you have any more questions about my response.

Best,
Anthea

Q3.1

Under H_0, $g(\cdot)$ is a constant function, which $\mu_1 = \mu_2 = \ldots = \mu_n$. So, the distribution of X_i is

$$X_i = \mu + \varepsilon_i$$

Note that R_i are uniformly distribution within 1, …, n. So the total number of pairs $= n(n-1)$

Then, we consider the case where $\min(R_1, R_2) = k$. The total number of possible pairs is

$$(n-k) + (n-k) = 2(n-k)$$

Hence, the total number of possible pairs, i.e. $k = 1, \ldots, n-1$, is

$$\begin{aligned} S &= \sum_{k=1}^{n-1} 2k(n-k) \\ &= 2\left[n \cdot \sum_{k=1}^{n-1} k - \sum_{k=1}^{n-1} k^2 \right] \\ &= \frac{n(n+1)(n-1)}{3} \end{aligned}$$

Therefore, the expected value is

$$\mathrm{E}\big\{ \min(R_1, R_2) \big\} = \frac{S}{n(n-1)} = \frac{n+1}{3}$$

Note that $\mathrm{E}\big\{ \min(R_1, R_2) \big\} = \mathrm{E}\big\{ \min(R_i, R_j) \big\}$ for all $i \neq j$, then we can know that

$$\mathrm{E}\big\{ \min(R_i, R_j) \big\} = \frac{n+1}{3}$$

Also note that the double summation in $\widehat{\rho}_{\mathrm{C}}^{(m)}$ contains total of $(n-m) \times m$ terms, while each of the expected value are the same. So we can rewrite

$$\begin{aligned} \mathrm{E}\big[\widehat{\rho}_C^{(m)}\big] &= -2 + \frac{6}{\lambda} \cdot \mathrm{E}\left[\sum_{i=1+m}^{n} \sum_{j=1}^{m} \min(R_i, R_{i-j}) \right] \\ &= -2 + \frac{6}{\lambda} \cdot m(n-m) \cdot \frac{n+1}{3} \\ &= -2 + \frac{6}{(n+1)(n-m)m} \cdot m(n-m) \cdot \frac{n+1}{3} \\ &= -2 + 2 \\ &= 0 \end{aligned}$$

Q3.2

Proof

$$
\begin{aligned}
\widehat{\rho}_{\mathrm{C}}^{(1)} &= -2 + \frac{6}{\lambda}\sum_{i=2}^{n}\sum_{j=1}^{1}\min(R_i, R_{i-1}) \\
&= -2 + \frac{3}{\lambda}\sum_{i=2}^{n}\left(R_i + R_{i-1} - |R_i - R_{i-1}|\right) \\
&= -2 + \frac{3}{\lambda}\sum_{i=2}^{n}(R_i + R_{i-1}) - \frac{3}{\lambda}\sum_{i=2}^{n}|R_i - R_{i-1}| \\
&= -2 + \frac{3}{n^2-1}\sum_{i=2}^{n}(R_i + R_{i-1}) - \frac{3}{n^2-1}\sum_{i=2}^{n}|R_i - R_{i-1}| \\
&= -3 + \frac{3}{n^2-1}\sum_{i=2}^{n}(R_i + R_{i-1}) + \left(1 - \frac{3}{n^2-1}\sum_{i=2}^{n}|R_i - R_{i-1}|\right) \\
&= -3 + \frac{3}{n^2-1}\sum_{i=2}^{n}(R_i + R_{i-1}) + \widehat{\rho}_{\mathrm{C}}(\{1,\ldots,n\}, X_{1:n}) \\
&= -3 + \frac{3}{n^2-1}\left[\left(\sum_{i=1}^{n}R_i\right) - R_1\right] + \frac{3}{n^2-1}\left[\left(\sum_{i=1}^{n}R_i\right) - R_n\right] + \widehat{\rho}_{\mathrm{C}}(\{1,\ldots,n\}, X_{1:n}) \\
&= -3 + \frac{3}{n^2-1}\cdot\frac{n(n+1)}{2} - \frac{3}{n^2-1}R_1 + \frac{3}{n^2-1}\cdot\frac{n(n+1)}{2} - \frac{3}{n^2-1}R_n + \widehat{\rho}_{\mathrm{C}}(\{1,\ldots,n\}, X_{1:n}) \\
&= -3 + \frac{3n}{2(n-1)} - \frac{3}{n^2-1}R_1 + \frac{3n}{2(n-1)} - \frac{3}{n^2-1}R_n + \widehat{\rho}_{\mathrm{C}}(\{1,\ldots,n\}, X_{1:n}) \\
&= -3 + \frac{3n}{(n-1)} - \frac{3}{n^2-1}R_1 - \frac{3}{n^2-1}R_n + \widehat{\rho}_{\mathrm{C}}(\{1,\ldots,n\}, X_{1:n}) \\
&= \frac{3}{n-1} - \frac{3}{n^2-1}R_1 - \frac{3}{n^2-1}R_n + \widehat{\rho}_{\mathrm{C}}(\{1,\ldots,n\}, X_{1:n}) \\
&= \widehat{\rho}_{\mathrm{C}}(\{1,\ldots,n\}, X_{1:n}) + \frac{3}{n-1}\left(1 - \frac{R_1}{n+1} - \frac{R_n}{n+1}\right)
\end{aligned}
$$

Rejecting H_0

Under H_1, if we assume $g(\cdot)$ is a increasing function, larger i would lead to larger X_i, and thus R_i, vice versa. So, there will be a strong positive or negative relationship between i and X_i, deviating the value of $\widehat{\rho}_C(\{1,\ldots,n\}, X_{1:n})$ from 0. Therefore, $\widehat{\rho}_C^{(1)}$ will be larger than the value under H_0, which i and X_i are not correlated. So, it is sensible to reject H_0 when $\widehat{\rho}_C^{(1)}$ is larger some c. Besides, $\widehat{\rho}_C$ is sensitive to monotonic trends and other dependency, which allow it to handle general H_1.

Q3.3

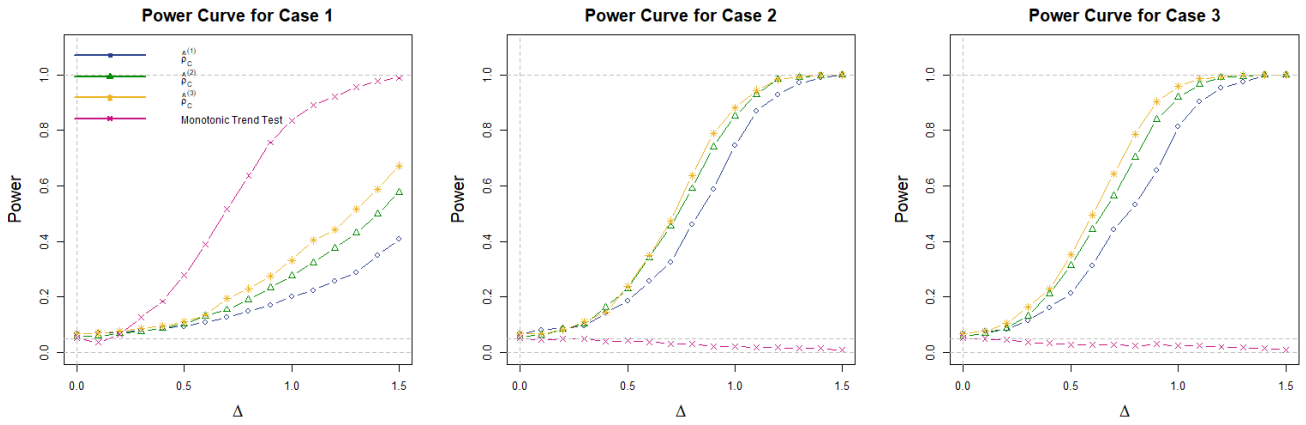
The following R code is used to generate the power curves:

```r
# Function to compute hat{rho_C^(m)}
compute_rho_C_m = function(X, m) {
  n = length(X)
  R = rank(X)
  lambda = (n+1)*(n-m)*m
  sum_min_R = 0
  for (j in 1:m) {
    index_i = (1+m):n
    index_j = (1+m-j):(n-j)
    min_R = pmin(R[index_i], R[index_j])
    sum_min_R = sum_min_R + sum(min_R)
  }
  rho_C_m = -2 + (6/lambda)*sum_min_R
  rho_C_m
}

# Function to compute p-value for hat{rho_C^(m)}
compute_p_value = function(rho_C_m, m, n) {
  z_value = rho_C_m / sqrt(2/(5*m*n) + (8*m) / (15*n^2))
  p_value = 1 - pnorm(z_value)
  p_value
}

# Function to generate X_i for 3 cases
generate_X = function(n, Delta, case_num) {
  u = (1:n) / (n+1)
  epsilon = rt(n, df=3)
  if (case_num == 1) {
    mu = Delta*u^3
  } else if (case_num == 2) {
    mu = Delta*8*(u-0.5)^2
  } else if (case_num == 3) {
    mu = Delta*(2*u^2*cos(6*pi*u))
  }
  X = mu + epsilon
  X
}

# Function to compute p-values for all tests
compute_p_values = function(X, m_values) {
  n = length(X)
  p_values = numeric(length(m_values) + 1)
  names(p_values) = c(paste0("rho_C", m_values), "monotonic_trend")

  for (i in seq_along(m_values)) {
    m = m_values[i]
    rho_C_m = compute_rho_C_m(X, m)
    p_values[i] = compute_p_value(rho_C_m, m, n)
  }

  # Using Spearman Correlation Test to perform Monotonic Trend Test (time index
      against data)
  time_index = 1:n
```

```r
  p_value_s = cor.test(time_index, X, method = "spearman",
                       exact = FALSE)$p.value
  p_values["monotonic_trend"] = p_value_s
  p_values
}

# Simulation parameters
n = 200
nRep = 500
Delta_all = seq(0, 1.5, by = 0.1)
size = 0.05
cases = 1:3
methods = c("rho_C^(1)", "rho_C^(2)", "rho_C^(3)", "monotonic_trend")
m_values = 1:3
power_results = array(NA, dim = c(length(Delta_all), length(methods),
                      length(cases)),
                      dimnames = list(paste0("Delta=", Delta_all),
                      methods,
                      paste0("Case", cases)))

# Run simulations
for (case_num in cases) {
  cat("Simulating for Case", case_num, "\n")
  for (Delta_idx in seq_along(Delta_all)) {
    Delta = Delta_all[Delta_idx]
    rejections = matrix(0, nrow = nRep, ncol = length(methods))
    colnames(rejections) = methods
    for (rep in 1:nRep) {
      set.seed(rep)
      X = generate_X(n, Delta, case_num)
      p_values = compute_p_values(X, m_values)
      rejections[rep, ] = (p_values < size)
    }
    power = colMeans(rejections)
    power_results[Delta_idx, , paste0("Case", case_num)] = power
  }
}

# Plot power curves
par(mfrow = c(1, length(cases)), mar = c(4.5, 5, 3, 2))
col = c("royalblue4", "green4", "goldenrod2", "mediumvioletred")
pch = c(1, 2, 8, 4)
lwd = rep(1.5, 4)
lty = 1

for (case_num in cases) for (case_num in cases) {
  matplot(Delta_all, power_results[, , paste0("Case", case_num)], type = "b",
          lwd = lwd, lty = lty, col = col, pch = pch,
          ylim = c(0, 1.1), xlab = expression(Delta), ylab = "Power",
          main = paste("Power Curve for Case", case_num), cex.lab = 1.5,
          cex.main = 1.5)
  abline(h = c(0, size, 1), lty = 2, col = "gray")
  abline(v = 0, lty = 2, col = "gray")
  if (case_num == 1) {
    legend("topleft",
           lwd = 2, lty = lty, col = col, pch = pch, cex = 0.85, bty="n",
```

```
            legend = c(expression(hat(rho)[C]^{(1)}),
                       expression(hat(rho)[C]^{(2)}),
                       expression(hat(rho)[C]^{(3)}),
                       "Monotonic Trend Test"),
            )
  }
}
```

The power curves of the tests based on $\widehat{\rho}_C^{(1)}, \widehat{\rho}_C^{(2)}, \widehat{\rho}_C^{(3)}$ and the trend test in Example 3.13 of the lecture note are as follows:



From the power curves, we can see that:

- Size performance: The size of all the tests are very close to 0.05.
- Power performance:
  - (Case 1) The monotonic trend test is the most powerful test since $g(u)$ is now monotonic increasing trend.
  - (Case 2 & 3) $\widehat{\rho}_{\mathrm{C}}^{(2)}$ and $\widehat{\rho}_{\mathrm{C}}^{(3)}$ have similar power, which are the best tests, while $\widehat{\rho}_{\mathrm{C}}^{(1)}$ has a slightly lower power, and the monotonic trend test has the least power.

According to simulation, I would prefer $\widehat{\rho}_{\mathrm{C}}^{(3)}$ since

- (General dependency) It fits to more cases since it can detect dependencies when the trend is non-monotonic. Besides, it is the most powerful test except than the monotonic trend test in detecting monotonic trend.
- (Power performance) It is the most powerful test for detecting non-monotonic trend.
- (Meaning of m) In general, $\widehat{\rho}_{\mathrm{C}}^{(m)}$ with a larger (but reasonable) value of m is better, since it enhances the ability to detect complex or non-monotonic trends which would lead to deviations from H_0.

Q4: (Q1)

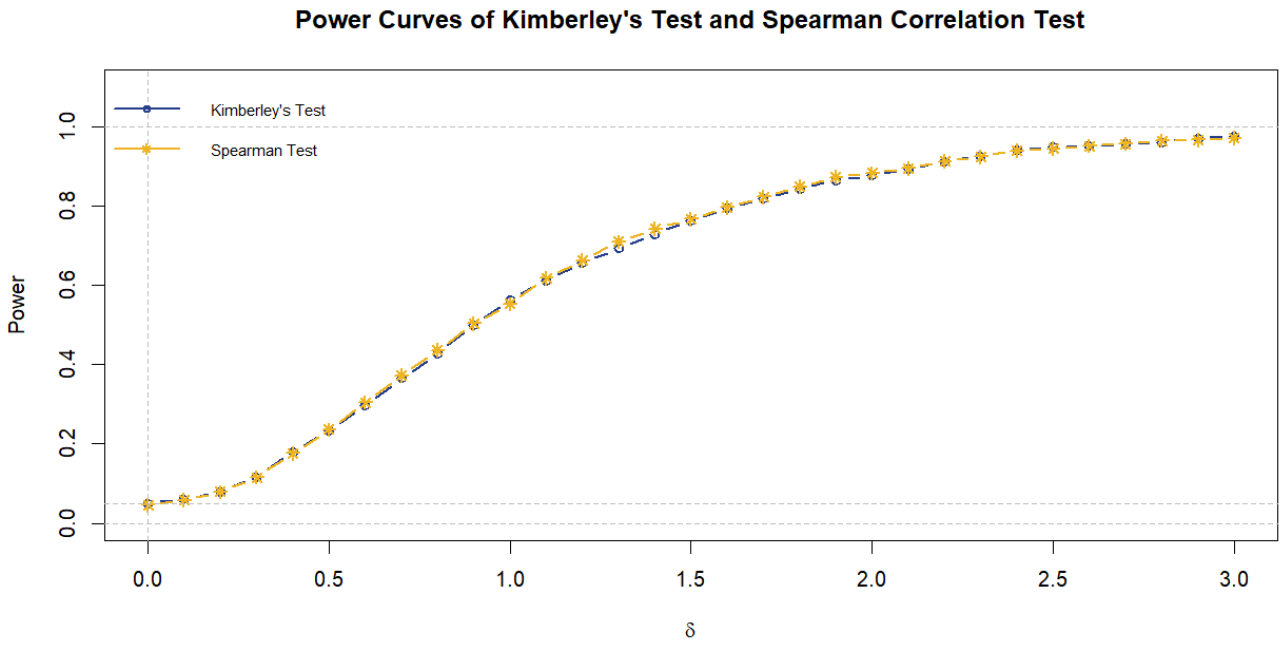
Objective

We would like to conduct a simulation experiment to compare the power of Kimberley's Test with the Spearman correlation test.

Reason of comparing with Spearman correlation test

The Spearman test was selected as a comparison because it can be used as a n-sample trend test, which is similar to the Kimberley's Test. The Spearman test can be used to evaluate the correlation between the indices of observations and their absolute deviations from the mean. Absolute deviations is used since it would be able to detect the ordered variance changes.

Power Curves



The power curves of the 2 tests are similar because they both can detect monotonic trend. Under H_1, where the variance of observations decreases progressively. Therefore, we can see that the power of the 2 tests are both high. Moreover, their size performance is similar.

Another important fact is that the 2 tests both are lower power (e.g. compared with Q3.3), since when δ is small, the variability of the data is also small. So, the tests requires larger δ to detect the trend and to reject H_0, thus leading to a higher power.

Preferred test

Kimberley's Test is preferred since it is designed to detect trends in ordered variances by assigning weights to observations based on their rank, so it can handle more scenarios than the Spearman test.

Q4: (Q1) (Cont'd)

R code

```r
# Define the scaleTrendTest() function if not defined in Q1
scaleTrendTest = function(x){
  r = rank(x)
  n = length(r)
  T = sum((1:n)*abs(r/(n+1)-1/2))
  ET = (n^2-1)/8
  varT = (n-1)*(n^2+3)/576
  t = (T-ET)/sqrt(varT)
  t
}

# Simulation parameters
n <- 100
num_simulations <- 1000
alpha <- 0.05
delta_values <- seq(0, 3, by = 0.1)

# Storing Power
power_kimberley <- numeric(length(delta_values))
power_spearman <- numeric(length(delta_values))

# Run simulations
for (k in seq_along(delta_values)) {
  delta <- delta_values[k]
  reject_kimberley <- 0
  reject_spearman <- 0

  for (iRep in 1:num_simulations) {
    set.seed(iRep)
    if(iRep%%500==0) cat(">> ")

    sigma_i <- 1 + delta*(1-(1:n)/n)
    x <- rnorm(n, mean = 0, sd = sigma_i)

    # Kimberley's Test
    t_kimberley <- scaleTrendTest(x)
    p_value_kimberley <- 2 * (1 - pnorm(abs(t_kimberley)))
    if (p_value_kimberley < alpha) {
      reject_kimberley <- reject_kimberley + 1
    }

    # Spearman Rank Correlation Test
    abs_deviation <- abs(x - mean(x))
    spearman_test <- cor.test(1:n, abs_deviation, method = "s")
    if (spearman_test$p.value < alpha) {
      reject_spearman <- reject_spearman + 1
    }
  }

  # Calculate power for each test
  power_kimberley[k] <- reject_kimberley / num_simulations
  power_spearman[k] <- reject_spearman / num_simulations
}
```

```r
# Plot the power curves
matplot(delta_values, cbind(power_kimberley, power_spearman), type = "b",
        lty = 1, col = c("royalblue4", "goldenrod2"), lwd = 2, pch = c(1, 8),
        xlab = expression(delta), ylab = "Power", ylim = c(0, 1.1),
        main = "Power Curves of Kimberley's Test and Spearman Correlation Test"
           )
abline(h = c(0, alpha, 1), lty = 2, col = "gray")
abline(v = 0, lty = 2, col = "gray")
legend("topleft", legend = c("Kimberley's Test", "Spearman Test"),
       col = c("royalblue4", "goldenrod2"), lty = 1, lwd = 2, pch = c(1, 8),
          cex = 0.75, bty = "n")
```