# Experiment 4

## 1. Kendall's Tau

**given**

| $i$ | $r(y_i^a)$ | $r(y_i^b)$ | $q_i$ |
|---|---|---|---|
| 5 | 1 | 2 | 1 |
| 6 | 2 | 3 | 1 |
| 7 | 3.5 | 5 | 2 |
| 4 | 3.5 | 1 | 0 |
| 3 | 5 | 4 | 0 |
| 2 | 6 | 7.5 | 2 |
| 8 | 7 | 7.5 | 1 |
| 1 | 8 | 6 | 0 |

$q_i$: number of $r\left(y_i^b\right)$ lower or equal $r\left(y_i^b\right)$ and in the order above under $r\left(y_i^b\right)$

It is sorted according to increasing ranks $r(y_i^a)$ of the first sequence.
Hint: If two elements had the same rank, then from both the average value is taken.

$$\tau = 1 - \frac{4\sum_{i=1}^{n} q_i}{n \cdot (n-1)} = 1 - \frac{4 \cdot (1+1+2+2+1)}{8 \cdot 7} = 1 - \frac{4 \cdot 7}{8 \cdot 7} = 0.5$$

this example was taken from:
**Walz Guido**, *Lexikon der Mathematik: Band 2*. Springer Berlin Heidelberg, 2017. doi: 10.1007/978-3-662-53504-2

Hint: It is not equivalent to the correct formula from Wikipedia (see below)
[https://de.wikipedia.org/wiki/Rangkorrelationskoeffizient#Kendalls_Tau](https://de.wikipedia.org/wiki/Rangkorrelationskoeffizient#Kendalls_Tau)

$$\tau = \frac{\#concordant - \#discordant}{S} = \frac{C - D}{\sqrt{(C + D + T_a) \cdot (C + D + T_b)}}$$

| | | | | |
|---|---|---|---|---|
| $concordant$: | $y_i^a < y_j^a \wedge y_i^b < y_j^b$ | \| | $y_i^a > y_j^a \wedge y_i^b > y_j^b$ | (sort order agreement) |
| $discordant$: | $y_i^a < y_j^a \wedge y_i^b > y_j^b$ | \| | $y_i^a > y_j^a \wedge y_i^b < y_j^b$ | (sort order disagreement) |
| $T_a$: | $y_i^a = y_j^a \wedge y_i^b \neq y_j^b$ | | | (bindings in a) |
| $T_b$: | $y_i^a \neq y_j^a \wedge y_i^b = y_j^b$ | | | (bindings in b) |

**idea**

sort by $r(y_i^a)$ in ascending order and compute signs for $r(y_i^b) - r(y_j^b)$ for $i < j$ and count signs

assumption: $y_i^a$ and $y_i^b$ are unique

if $\boldsymbol{r(y_i^b) - r(y_j^b) < 0}$

then **concordant** because $r(y_i^a) - r(y_j^a) < 0$ holds,
since $r(y_i^a)$ are sorted in ascending order

else **discordant**

| $\boldsymbol{r(y_i^a)}$ | 1 | 2 | 3.5 | 3.5 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $\boldsymbol{r(y_i^b)}$ | 2 | 3 | 5 | 1 | 4 | 7.5 | 7.5 | 6 |

**calculation**

count boxes: $\frac{n \cdot (n-1)}{2} = \frac{8 \cdot 7}{2} = 28$

| 1-2 | 1-3.5 | 1-3.5 | 1-5 | 1-6 | 1-7 | 1-8 |
|---|---|---|---|---|---|---|
| **2-3** | **2-5** | **2-1** | **2-4** | **2-7.5** | **2-7.5** | **2-6** |
| - | - | + | - | - | - | - |

| 2-3.5 | 2-3.5 | 2-5 | 2-6 | 2-7 | 2-8 |
|---|---|---|---|---|---|
| **3-5** | **3-1** | **3-4** | **3-7.5** | **3-7.5** | **3-6** |
| - | + | - | - | - | - |

| 3.5-3.5 | 3.5-5 | 3.5-6 | 3.5-7 | 3.5-8 |
|---|---|---|---|---|
| **5-1** | **5-4** | **5-7.5** | **5-7.5** | **5-6** |
| a | + | - | - | - |

| 3.5-5 | 3.5-6 | 3.5-7 | 3.5-8 |
|---|---|---|---|
| **1-4** | **1-7.5** | **1-7.5** | **1-6** |
| - | - | - | - |

| 5-6 | 5-7 | 5-8 |
|---|---|---|
| **4-7.5** | **4-7.5** | **4-6** |
| - | - | - |

| 6-7 | 6-8 |
|---|---|
| **7.5-7.5** | **7.5-6** |
| b | + |

| 7-8 |
|---|
| **7.5-6** |
| + |

$C$: 21 $D$: 5 $T_a$: 1 $T_b$: 1

$$\tau = \frac{C-D}{\sqrt{(C+D+T_a)\cdot(C+D+T_b)}} = \frac{21-5}{\sqrt{27\cdot 27}} \approx 0.5925926$$ (correct)

```
> cor(c(8,6,5,3.5,1,2,3.5,7), c(6,7.5,4,1,2,3,5,7.5), method="kendall")
[1] 0.5925926
```

# 2. Tukey's Biweight

**W-Estimator** (usual fast implementation – works nicely)

Hint: algorithm from Statistical Algorithms Description Document, 2002, Affymetrix

```
#########################################################
#' Implements Tukey's one-step biweight algorithm for robust mean calculation
#' using a w-estimator instead of a m-estimator.
#' Hint: Parameter c is by default set on value 9.
#'
#' @param x {vector} the values
#' for which the value has to be computed
#' @param c {numerical} tuning parameter
#'
#' @return {numerical} the function value
#' @source Statistical Algorithms Description Document, 2002, Affymetrix
#' at page 22
Math.tukeysBiweightRobustMean <- function(x, c = 9) {
  median <- median(x);
  mad <- mad(x, constant = 1);  # median(c * |x_i - median(x)|), hint: default method multiplies with constant c = 1.4826

  # epsilon to avoid division by zero
  uValues <- (x-median)/(c * mad + Maths.EPSILON);  # analogous to (x-mean)/sd

  weights <- sapply(uValues, Math.weight);  # where outlier weights removed because their values are 0

  mean <- sum(weights*x)/sum(weights);  # where outlier weights removed because their values are 0

  return(mean);  # compute mean
}


#########################################################
#' Implements Tukey's weighting function.
#'
#' @param u {numerical} the value for which a weight has to be computed
#'
#' @return {numerical} a weight
Math.weight <- function(u) {
  if (abs(u) <= 1) {
    return((1-u^2)^2);
  }

  return(0);
}
```

**given**
$y = \langle 2, 3, 5 \rangle$
$c = 9$

**calculation**
$\text{median} = 3$
$\text{MAD}_y = \text{median}\{|2-3|, |3-3|, |5-3|\} = \text{median}\{0, 1, 2\} = 1$

$$m_1 = \frac{2-3}{9 \cdot 1 + 0.0001} = \frac{-1}{9.0001} \approx -0.111$$

$$m_2 = \frac{3-3}{9 \cdot 1 + 0.0001} = 0$$

$$m_3 = \frac{5-3}{9 \cdot 1 + 0.0001} = \frac{2}{9.0001} \approx 0.222$$

---

$$w(m_1) = \begin{cases} (1 - m_1^2)^2 & , |m_1| \leq 1 \\ 0 & , |m_1| > 1 \end{cases} = \left(1 - \left(\frac{-1}{9.0001}\right)^2\right)^2 \approx 0.975$$

$$w(m_2) = \left(1 - 0^2\right)^2 = 1$$

$$w(m_3) = \left(1 - \left(\frac{2}{9.0001}\right)^2\right)^2 \approx 0.904$$

---

$$\bar{y} = \frac{w(m_1) \cdot 2 + w(m_2) \cdot 3 + w(m_3) \cdot 5}{w(m_1) + w(m_2) + w(m_3)} = 3.288936942$$

```
> dplR::tbrm(c(2,3,5));
[1] 3.288936

> Math.tukeysBiweightRobustMean(a)
[1] 3.288937
```