Reference Solutions of HW1

Please contact lu-c26@mails.tsinghua.edu.cn for any questions or suggestions.

Problem 1 We consider completely randomized assignment with fixed $N_{\mathrm{t}} = N_{\mathrm{c}} = n = 26$, under which all $\binom{52}{26}$ assignments are equally likely.

(a) Flying on both types of days controls for the flight itself. Blinding the pilot reduces treatment-dependent changes in flight operations. Both help isolate the effect of cloud seeding.

(b) **Slides 2, page 19.** Denote the number of random assignments by $K = 1000$. Permute treatment labels and compute

$$T^{\mathrm{dif,obs}} = \bar{Y}_{\mathrm{t}}^{\mathrm{obs}} - \bar{Y}_{\mathrm{c}}^{\mathrm{obs}} = 277.4, \qquad \hat{p} = \frac{1}{K}\sum_{b=1}^{K} \mathbf{1}\{T^{\mathrm{dif},b} \geqslant T^{\mathrm{dif,obs}}\} = 0.019,$$

Reported numerical results use Python. Monte Carlo estimates may vary across runs and software.

- Python

```python
# HW1_1b
import numpy as np

seeded = np.array([
    2745.6, 274.7, 115.3, 1697.1, 274.7, 92.4, 1656.4, 255.0, 40.6, 978.0, 242.5, 32.7, 703.4,
    200.7, 31.4, 489.1, 198.6, 17.5, 430.0, 129.6, 7.7, 334.1, 119.0, 4.1, 302.8, 118.3
])
unseeded = np.array([
    1202.6, 87.0, 26.0, 830.1, 81.2, 24.4, 372.4, 68.5, 21.4, 345.5, 47.3, 17.3, 321.2,
    41.1, 11.5, 244.3, 36.6, 4.9, 163.0, 29.0, 4.9, 147.8, 28.6, 1.0, 95.0, 26.3
])

y = np.r_[seeded, unseeded]
K = 1000
rng = np.random.default_rng(2026)
# Each row is one reassignment; the first 26 days are treated.
permuted = np.array([rng.permutation(y) for _ in range(K)])
t_obs = seeded.mean() - unseeded.mean()
t_sim = permuted[:, :26].mean(axis=1) - permuted[:, 26:].mean(axis=1)
p_mean = np.mean(t_sim >= t_obs)
print(f"T = {t_obs:.1f}, p = {p_mean:g}")
```

```
T = 277.4, p = 0.019
```

- R

```r
# HW1_1b
seeded <- c(
    2745.6, 274.7, 115.3, 1697.1, 274.7, 92.4, 1656.4, 255.0, 40.6, 978.0, 242.5, 32.7, 703.4,
    200.7, 31.4, 489.1, 198.6, 17.5, 430.0, 129.6, 7.7, 334.1, 119.0, 4.1, 302.8, 118.3
)
unseeded <- c(
    1202.6, 87.0, 26.0, 830.1, 81.2, 24.4, 372.4, 68.5, 21.4, 345.5, 47.3, 17.3, 321.2,
    41.1, 11.5, 244.3, 36.6, 4.9, 163.0, 29.0, 4.9, 147.8, 28.6, 1.0, 95.0, 26.3
)

y <- c(seeded, unseeded)
K <- 1000
set.seed(2026)
# Each column is one reassignment; the first 26 days are treated.
permuted <- replicate(K, sample(y))
t_obs <- mean(seeded) - mean(unseeded)
t_sim <- colMeans(permuted[1:26, ]) - colMeans(permuted[27:52, ])
p_mean <- mean(t_sim >= t_obs)
cat(sprintf("T = %.1f, p = %g\n", t_obs, p_mean))
```

```
T = 277.4, p = 0.017
```

(c) **Slides 2, page 19.** Conditional on the observed data, let p^* be the exact p-value. Independent reassignments give $K\hat{p} \sim \text{Binomial}(K, p^*)$, hence

$$\hat{\text{SE}}(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{K}} = \sqrt{\frac{0.019(1-0.019)}{1000}} \approx 0.00432.$$

(d) **Slides 2, pages 22–23.** When we assume equal variances, with $N = N_\text{t} + N_\text{c} = 2n = 52$, we obtain

$$t^\text{obs} = \frac{\bar{Y}_\text{t}^\text{obs} - \bar{Y}_\text{c}^\text{obs}}{\sqrt{\frac{N}{N_\text{t}N_\text{c}(N-2)}\left[\sum_{i:W_i=1}(Y_i^\text{obs} - \bar{Y}_\text{t}^\text{obs})^2 + \sum_{i:W_i=0}(Y_i^\text{obs} - \bar{Y}_\text{c}^\text{obs})^2\right]}} \approx 2.00,$$

$$p \approx 0.0256.$$

When we allow unequal variances instead, we obtain

$$t^\text{obs} = \frac{\bar{Y}_\text{t}^\text{obs} - \bar{Y}_\text{c}^\text{obs}}{\sqrt{s_\text{t}^2/N_\text{t} + s_\text{c}^2/N_\text{c}}} \approx 2.00, \qquad p \approx 0.0269.$$

In large samples, both the randomization distribution in (b) and the traditional t reference distributions admit normal approximations, giving similar p-values. The traditional t-tests give slightly larger p-values for the data.

- Python

```python
# HW1_1d
from scipy import stats

for case, equal_var in [("Assume equal variances", True), ("Do not assume equal variances", False)]:
    result = stats.ttest_ind(seeded, unseeded, equal_var=equal_var, alternative="greater")
    df = f"{result.df:.0f}" if equal_var else f"{result.df:#.3g}"
    print(f"{case}: t = {result.statistic:#.3g}, df = {df}, p = {result.pvalue:#.3g}")
```

```
Assume equal variances: t = 2.00, df = 50, p = 0.0256
Do not assume equal variances: t = 2.00, df = 33.9, p = 0.0269
```

- R

```r
# HW1_1d
for (equal_var in c(TRUE, FALSE)) {
  result <- t.test(seeded, unseeded, var.equal = equal_var, alternative = "greater")
  case <- if (equal_var) "Assume equal variances" else "Do not assume equal variances"
  df <- format(result$parameter, digits = 3, trim = TRUE)
  cat(sprintf("%s: t = %#.3g, df = %s, p = %#.3g\n",
              case, result$statistic, df, result$p.value))
}
```

```
Assume equal variances: t = 2.00, df = 50, p = 0.0256
Do not assume equal variances: t = 2.00, df = 33.9, p = 0.0269
```

(e) We can use $T^{\text{var}} = s_{\text{t}}^2 - s_{\text{c}}^2$ and its randomization upper tail, where both sample variances use denominator $n-1$.

$$T^{\text{var,obs}} = 345966.6872, \qquad \hat{p}_{\text{var}} = \frac{1}{K}\sum_{b=1}^{K} \mathbf{1}\{T^{\text{var},b} \geqslant T^{\text{var,obs}}\} = 0.103.$$

Other forms, such as a difference in log variances, are also acceptable. Note that this test, which is more sensitive to changes in variability, is still based on Fisher's sharp null, rather than a null hypothesis of equal variances.

- Python

```python
# HW1_1e
t_var = seeded.var(ddof=1) - unseeded.var(ddof=1)
var_sim = (permuted[:, :26].var(axis=1, ddof=1)
           - permuted[:, 26:].var(axis=1, ddof=1))
p_var = np.mean(var_sim >= t_var)
print(f"Variance difference = {t_var:.4f}, p = {p_var:g}")
```

```
Variance difference = 345966.6872, p = 0.103
```

- R

```r
# HW1_1e
t_var <- var(seeded) - var(unseeded)
var_sim <- apply(permuted[1:26, ], 2, var) - apply(permuted[27:52, ], 2, var)
p_var <- mean(var_sim >= t_var)
cat(sprintf("Variance difference = %.4f, p = %g\n", t_var, p_var))
```

```
Variance difference = 345966.6872, p = 0.074
```

(f) **Slides 2, page 13.** Since taking logs turns multiplicative effects into additive effects, we use

$$T^{\log} = \frac{1}{N_{\text{t}}} \sum_{i:W_i=1} \log Y_i^{\text{obs}} - \frac{1}{N_{\text{c}}} \sum_{i:W_i=0} \log Y_i^{\text{obs}}.$$

$$T^{\log,\text{obs}} \approx 1.14, \qquad \hat{p}_{\log} = \frac{1}{K}\sum_{b=1}^{K} \mathbf{1}\{T^{\log,b} \geqslant T^{\log,\text{obs}}\} = 0.005.$$

Note that all observations are positive, so the logs are well defined.

- Python

```python
# HW1_1f
t_log = np.log(seeded).mean() - np.log(unseeded).mean()
log_sim = (np.log(permuted[:, :26]).mean(axis=1)
           - np.log(permuted[:, 26:]).mean(axis=1))
p_log = np.mean(log_sim >= t_log)
print(f"Log-mean difference = {t_log:#.3g}, p = {p_log:g}")
```

```
Log-mean difference = 1.14, p = 0.005
```

- R

```r
# HW1_1f
t_log <- mean(log(seeded)) - mean(log(unseeded))
log_sim <- colMeans(log(permuted[1:26, ])) - colMeans(log(permuted[27:52, ]))
p_log <- mean(log_sim >= t_log)
cat(sprintf("Log-mean difference = %#.3g, p = %g\n", t_log, p_log))
```

```
Log-mean difference = 1.14, p = 0.006
```

(g) At the 5% significance level, the mean and log-mean tests reject Fisher's sharp null and provide evidence in the direction of increased rainfall, while the variance-based test does not reject. Note that these results do not establish an increase on every day or a constant treatment effect.

Problem 2 Here $n = n_1 + n_0$ is the total sample size, with n_1 treated and n_0 control units. We extend the stated rank definition by breaking ties randomly, independently of treatment assignment, and keep these ranks fixed throughout the test. Then the ranks are a permutation of $\{1, \dots, n\}$, as in the case without ties. For example, $(7, 8, 8, 99)$ receives ranks $(1, 2, 3, 4)$ or $(1, 3, 2, 4)$ with equal probability. Without this extension, if all values are equal, $E(T_W) = n_1 n \neq n_1(n+1)/2$ so the conclusion in (b) does not hold.

(a) Under Fisher's sharp null, the ranks are fixed and the treated ranks form a simple random sample of size n_1 from $\{1, \dots, n\}$. Larger or smaller treated outcomes tend to give correspondingly larger or smaller rank sums, so T_W can serve as a test statistic.

(b) With n_1 treated units selected uniformly, for $i \neq j$,

$$E(W_i) = \frac{n_1}{n}, \qquad \mathrm{Var}(W_i) = \frac{n_1 n_0}{n^2}, \qquad E(W_i W_j) = \frac{n_1(n_1-1)}{n(n-1)}, \qquad \mathrm{Cov}(W_i, W_j) = -\frac{n_1 n_0}{n^2(n-1)}.$$

Since $\sum_{i=1}^n R_i = n(n+1)/2$ and $\sum_{i=1}^n R_i^2 = n(n+1)(2n+1)/6$, we obtain

$$E(T_W) = \sum_{i=1}^n R_i E(W_i) = \frac{n_1(n+1)}{2},$$

$$\mathrm{Var}(T_W) = \sum_{i=1}^n R_i^2 \,\mathrm{Var}(W_i) + 2\sum_{i<j} R_i R_j \,\mathrm{Cov}(W_i, W_j) = \frac{n_1 n_0}{n^2(n-1)} \left[n \sum_{i=1}^n R_i^2 - \left(\sum_{i=1}^n R_i \right)^2 \right] = \frac{n_1 n_0 (n+1)}{12}.$$

(c) **Slides 2, page 23.** Here $n_1 = n_0 = 26$ and $n = 52$. Using the exact randomization distribution of the rank sum,

$$T_W^{\mathrm{obs}} = 824, \qquad T_W^{\mathrm{obs}} - \frac{n_1(n+1)}{2} = 135, \qquad p = \Pr(T_W \geqslant 824) \approx 0.00648.$$

Both functions, Python's `scipy.stats.mannwhitneyu` and R's `wilcox.test`, use the equivalent test statistic $U = T_W - n_1(n_1+1)/2$, which gives the same upper-tail p-value. We compute its exact distribution by counting assignments for each possible rank sum and reject Fisher's sharp null at the 5% significance level.

Using the original ranks $R_i = \#\{j : Y_j \leqslant Y_i\}$ gives $T_W^{\mathrm{obs}} = 825$, $T_W^{\mathrm{obs}} - n_1(n+1)/2 = 136$, and $p \approx 0.00643$.

- Python

```
# HW1_2c
# Sort by outcome, breaking ties with random keys.
order = np.lexsort((rng.random(52), y))
ranks = np.empty(52, dtype=int)
ranks[order] = np.arange(1, 53)
t_w = ranks[:26].sum()
p_exact = stats.mannwhitneyu(ranks[:26], ranks[26:],
                             alternative="greater", method="exact").pvalue
print(f"Random tie-breaking: T = {t_w}, centered T = {t_w - 26 * 53 / 2:g}, "
      f"p (exact) = {p_exact:#.3g}")

original_ranks = stats.rankdata(y, method="max").astype(int)
t_original = original_ranks[:26].sum()
# counts[k, s]: number of subsets of size k with rank sum s.
counts = np.zeros((27, original_ranks.sum() + 1), dtype=np.int64)
counts[0, 0] = 1
for r in original_ranks:
    for k in range(26, 0, -1):
        counts[k, r:] += counts[k - 1, :-r]
p_original = counts[26, t_original:].sum() / counts[26].sum()
print(f"Original ranks: T = {t_original}, shifted T = {t_original - 26 * 53 / 2:g}, "
      f"p (exact) = {p_original:#.3g}")
```

```
Random tie-breaking: T = 824, centered T = 135, p (exact) = 0.00648
Original ranks: T = 825, shifted T = 136, p (exact) = 0.00643
```

- R

```
# HW1_2c
ranks <- rank(y, ties.method = "random")
t_w <- sum(ranks[1:26])
p_exact <- wilcox.test(ranks[1:26], ranks[27:52], alternative = "greater", exact = TRUE)$p.value
cat(sprintf("Random tie-breaking: T = %g, centered T = %g, p (exact) = %#.3g\n",
            t_w, t_w - 26 * 53 / 2, p_exact))

original_ranks <- rank(y, ties.method = "max")
t_original <- sum(original_ranks[1:26])
# counts[k + 1, s + 1]: number of subsets of size k with rank sum s.
M <- sum(original_ranks) + 1
counts <- matrix(0, nrow = 27, ncol = M)
counts[1, 1] <- 1
for (r in original_ranks) {
  for (k in 26:1) {
    counts[k + 1, (r + 1):M] <- counts[k + 1, (r + 1):M] + counts[k, 1:(M - r)]
  }
}
p_original <- sum(counts[27, (t_original + 1):M]) / sum(counts[27, ])
cat(sprintf("Original ranks: T = %g, shifted T = %g, p (exact) = %#.3g\n",
            t_original, t_original - 26 * 53 / 2, p_original))
```

```
Random tie-breaking: T = 824, centered T = 135, p (exact) = 0.00648
Original ranks: T = 825, shifted T = 136, p (exact) = 0.00643
```

(d) **Slides 2, page 23.** Applying the stated CLT without continuity correction gives

$$Z^{\text{obs}} = \frac{T_W^{\text{obs}} - n_1(n+1)/2}{\sqrt{n_1 n_0 (n+1)/12}} \approx 2.47, \qquad p \approx 1 - \Phi(Z^{\text{obs}}) \approx 0.00674.$$

We reject Fisher's sharp null at the 5% significance level.

- Python

```
# HW1_2d
z = (t_w - 26 * 53 / 2) / np.sqrt(26 * 26 * 53 / 12)
print(f"Random tie-breaking: z = {z:#.3g}, p = {stats.norm.sf(z):#.3g}")
```

```
Random tie-breaking: z = 2.47, p = 0.00674
```

- R

```
# HW1_2d
z <- (t_w - 26 * 53 / 2) / sqrt(26 * 26 * 53 / 12)
cat(sprintf("Random tie-breaking: z = %#.3g, p = %#.3g\n", z, pnorm(z, lower.tail = FALSE)))
```

```
Random tie-breaking: z = 2.47, p = 0.00674
```

(e) **Slides 2, page 13.** The rank-sum statistic is less affected by extreme observations because it uses their order rather than their magnitude. The difference in means retains magnitude information and measures changes on the original rainfall scale, but can be dominated by a few extreme values. Under the sharp null, assigning two of $(7, 8, 8, 99)$ to treatment makes the sign of the observed mean difference depend entirely on which group receives 99, whereas the rank-sum statistic limits its contribution to the highest rank.

Problem 3 The observed potential outcomes are shown below, using the format of Table 2 in the question.

(i) The average causal effect, conditional on a left neighbor taking treatment, is $\tau_1 = 8^{-1}\sum_{i=1}^{8}\{Y_i(1,1) - Y_i(0,1)\}$. An estimator is

$$\hat{\tau}_1 = \frac{\sum_{i=1}^{8}\mathbb{I}\{D_i = 1, D_{i-1} = 1\}Y_i}{\#\{i : D_i = 1, D_{i-1} = 1\}} - \frac{\sum_{i=1}^{8}\mathbb{I}\{D_i = 0, D_{i-1} = 1\}Y_i}{\#\{i : D_i = 0, D_{i-1} = 1\}} = -7\%.$$

Table 2: Observed and Unobserved Potential Outcomes

Unit	D_i	D_{i-1}	$Y_i(1,1)$	$Y_i(1,0)$	$Y_i(0,1)$	$Y_i(0,0)$
1	1	1	4%			
2	1	1	2%			
3	0	1			8%	
4	1	0		9%		
5	0	1			12%	
6	0	0				13%
7	0	0				4%
8	1	0		1%		

(ii) The average causal effect, conditional on a left neighbor taking control, is $\tau_0 = 8^{-1}\sum_{i=1}^{8}\{Y_i(1,0) - Y_i(0,0)\}$. An estimator is

$$\hat{\tau}_0 = \frac{\sum_{i=1}^{8}\mathbb{I}\{D_i = 1, D_{i-1} = 0\}Y_i}{\#\{i : D_i = 1, D_{i-1} = 0\}} - \frac{\sum_{i=1}^{8}\mathbb{I}\{D_i = 0, D_{i-1} = 0\}Y_i}{\#\{i : D_i = 0, D_{i-1} = 0\}} = -3.5\%.$$

(iii) The effect modification is the difference between (i) and (ii):

$$\tau_{\text{int}} = \tau_1 - \tau_0, \qquad \hat{\tau}_{\text{int}} = \hat{\tau}_1 - \hat{\tau}_0 = -7\% - (-3.5\%) = -3.5\%.$$

Problem 4

1. Take $n = 3$ and consider the following cases.

(a) $\delta_1 = \delta_2$

i	$Y_i(1)$	$Y_i(0)$
1	4	2
2	5	3
3	6	4

(b) $\delta_1 > \delta_2$

i	$Y_i(1)$	$Y_i(0)$
1	4	3
2	5	4
3	6	2

(c) $\delta_1 < \delta_2$

i	$Y_i(1)$	$Y_i(0)$
1	4	4
2	5	2
3	6	3

(a) $\delta_1 = \text{median}\{4,5,6\} - \text{median}\{2,3,4\} = 5 - 3 = 2, \quad \delta_2 = \text{median}\{4-2, 5-3, 6-4\} = \text{median}\{2,2,2\} = 2.$
(b) $\delta_1 = \text{median}\{4,5,6\} - \text{median}\{3,4,2\} = 5 - 3 = 2, \quad \delta_2 = \text{median}\{4-3, 5-4, 6-2\} = \text{median}\{1,1,4\} = 1.$
(c) $\delta_1 = \text{median}\{4,5,6\} - \text{median}\{4,2,3\} = 5 - 3 = 2, \quad \delta_2 = \text{median}\{4-4, 5-2, 6-3\} = \text{median}\{0,3,3\} = 3.$

2. Comparison of estimands δ_1 and δ_2.

   (a) $\delta_1 = \text{median}\{Y_i(1)\}_{i=1}^n - \text{median}\{Y_i(0)\}_{i=1}^n$ represents the difference in the population medians under treatment and control. It is useful when you are interested in understanding the overall effect of treatment on the population, and it summarizes the treatment effect by comparing the central tendency (median) of the outcomes in both groups. For example, consider a policy intervention aimed at reducing crime rates in a city. If the goal is to assess the overall impact of the policy on crime levels across the city, δ_1 would be appropriate as it provides a summary measure of the change in the median crime rate across areas within the city due to the intervention.

   (b) $\delta_2 = \text{median}\{Y_i(1) - Y_i(0)\}_{i=1}^n$ captures the median of the individual treatment effects. It is beneficial when you want to focus on the individual treatment effects rather than the group averages, and it summarizes the median individual effect relative to the same individual's outcome without treatment. For example, consider a clinical trial evaluating the effectiveness of a new drug for pain relief. In this case, δ_2 would be more relevant as it captures the median individual effect of the drug.

Problem 5 This is an open question. Relevant topics include units, treatments and potential outcomes (**Slides 1, page 28**), SUTVA (**Slides 1, page 38**), pre-treatment covariates (**Slides 1, page 43**), etc.