

# STATS604 Project 3

Chandler Nielsen

2025-10-29

## Reading in Data

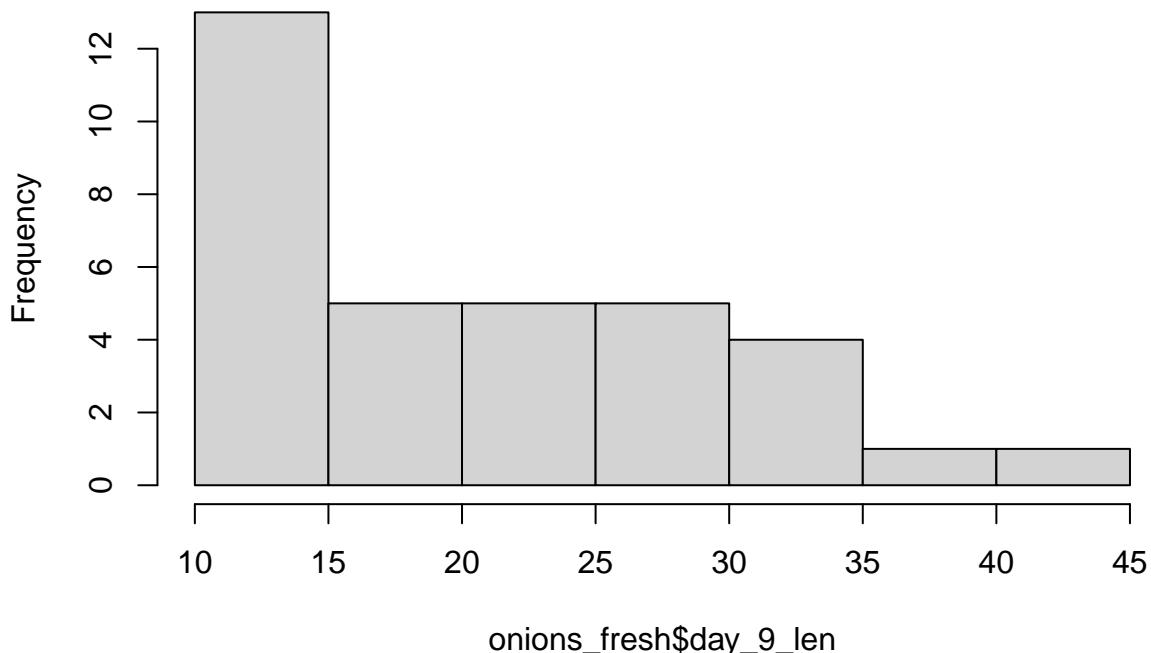
```
proj3_data <- read.csv("../data/final_dataset.csv")
```

## EDA

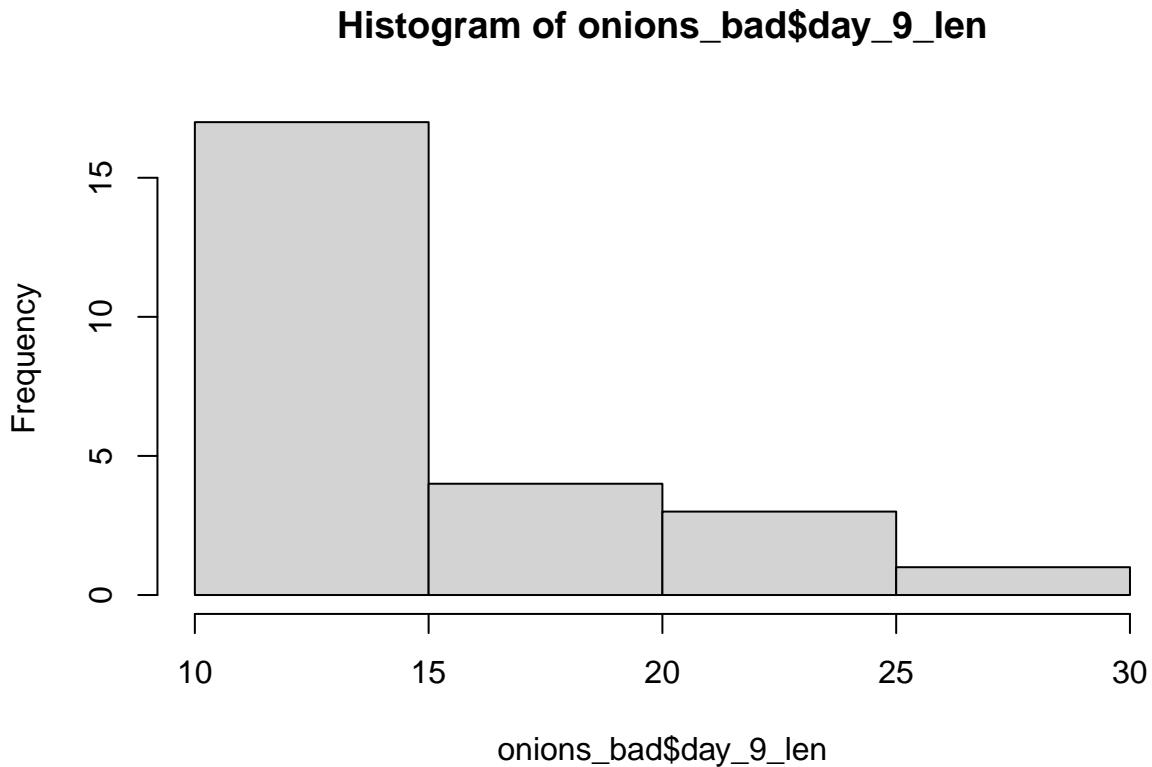
Separating data into fresh/not fresh:

```
onions_fresh <- proj3_data[proj3_data["freshness"] == 1, ]
onions_bad <- proj3_data[proj3_data["freshness"] == 0, ]
p1 <- hist(onions_fresh$day_9_len)
```

**Histogram of onions\_fresh\$day\_9\_len**

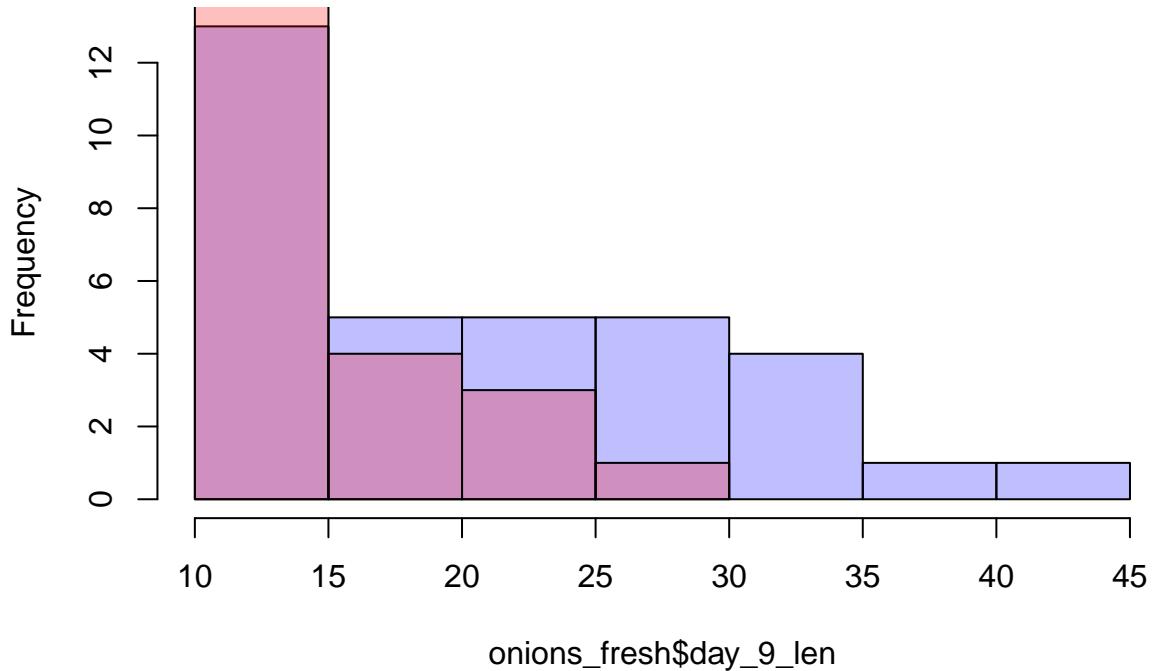


```
p2 <- hist(onions_bad$day_9_len)
```



```
plot(p1, col = rgb(0, 0, 1, 1/4))
plot(p2, col = rgb(1, 0, 0, 1/4), add=T)
```

## Histogram of onions\_fresh\$day\_9\_len



## Freshness Permutation Tests

### Difference in Proportions Test - Water Effect

Consider the statistic

$$\text{diff}_s = \text{Prop}(\text{fresh} = 1 \mid \text{water} = 1, \text{sunlight} = s) - \text{Prop}(\text{fresh} = 1 \mid \text{water} = 0, \text{sunlight} = s)$$

Performing difference in proportions test while using the statistic

$$T_{obs} = \frac{n_{\text{dark}} \cdot \text{diff}_{\text{dark}} + n_{\text{sun}} \cdot \text{diff}_{\text{sun}}}{n_{\text{dark}} + n_{\text{sun}}}$$

I will also pool labs. First, I will one-hot-encode the sunlight/water variables. We have the following:

```
# Keeping rows only with SW/SN/DW/DN
experimental_condition <- grep("^[SD][WN][0-9]+$", proj3_data$label)
df_exp <- proj3_data[experimental_condition, , drop = FALSE]
df_exp$sunlight <- ifelse(substr(df_exp$label, 1, 1) == "S", 1, 0)
df_exp$water <- ifelse(substr(df_exp$label, 2, 2) == "W", 1, 0)
```

Next, I will define a function to compute the water effect controlling for sunlight:

```

water_effect_stat <- function(input_data) {

  # Calculate difference in proportion for each sunlight group
  sun_levels <- sort(unique(input_data$sunlight))
  diff <- numeric(length(sun_levels))
  wts <- numeric(length(sun_levels))

  for (i in seq_along(sun_levels)) {
    s <- sun_levels[i]
    sub <- input_data[input_data$sunlight == s, , drop=FALSE]

    p_water <- mean(sub$freshness[sub$water == 1])
    p_nowater <- mean(sub$freshness[sub$water == 0])

    diff[i] <- p_water - p_nowater
    wts[i] <- nrow(sub)
  }

  sum(diff * wts) / sum(wts)
}

T_obs <- water_effect_stat(df_exp)
T_obs

```

## [1] 0.4393939

Now I can run the permutation test:

```

set.seed(82803)
B <- 10000
T_perm <- numeric(B)

sun_levels <- sort(unique(df_exp$sunlight))

for (b in seq_len(B)) {
  dat_perm <- df_exp

  # Permute water inside each sunlight stratum
  for (s in sun_levels) {
    light_ind <- which(df_exp$sunlight == s)
    dat_perm$water[light_ind] <- sample(df_exp$water[light_ind])
  }

  T_perm[b] <- water_effect_stat(dat_perm)
}

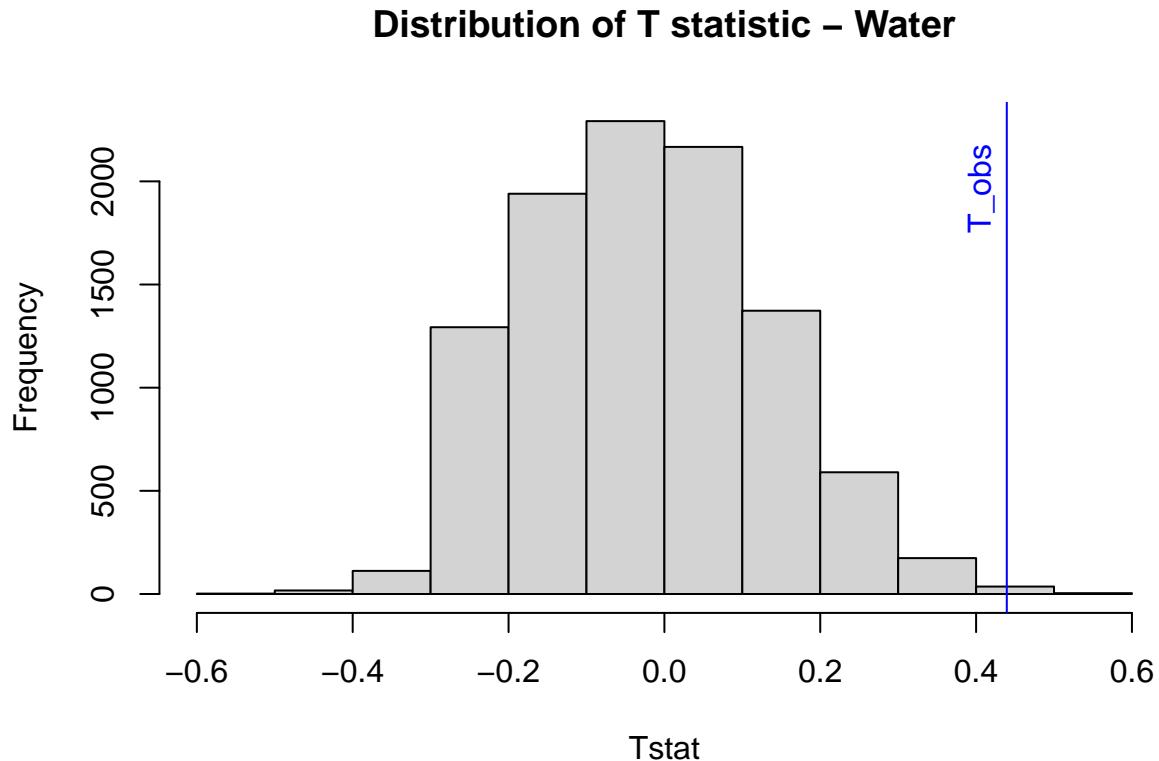
# One sided p-value (water increases freshness)
p_one_sided <- (sum(T_perm >= T_obs) + 1) / (B + 1)
print(p_one_sided)

```

## [1] 0.00409959

The above is the one-sided p-value. We can also produce a histogram for the distribution of permuted statistics. This is illustrated below:

```
hist(T_perm,
  main = "Distribution of T statistic - Water",
  xlab = "Tstat")
abline(v = T_obs, col="blue")
text(x = 0.38444848, y = 1750, labels = "T_obs", srt = 90, pos = 4, col = "blue")
```



### Difference in Proportions Test - Sunlight Effect

We will now perform the same test as before, but now we will condition on the water and test whether sunlight was significant.

```
sun_effect_stat <- function(input_data) {

  # Calculate difference in proportion for each water group
  water_levels <- sort(unique(input_data$water))
  diffs <- numeric(length(water_levels))
  wts <- numeric(length(water_levels))

  for (i in seq_along(water_levels)) {
    s <- water_levels[i]
    sub <- input_data[input_data$water == s, , drop=FALSE]
```

```

    p_sun <- mean(sub$freshness[sub$sunlight == 1])
    p_nosun <- mean(sub$freshness[sub$sunlight == 0])

    diffs[i] <- p_sun - p_nosun
    wts[i] <- nrow(sub)

}

sum(diffs * wts) / sum(wts)
}

T_obs <- sun_effect_stat(df_exp)
T_obs

```

```
## [1] 0.06060606
```

Running permutation test to assess effect of sunlight:

```

set.seed(82803)
B <- 10000
T_perm <- numeric(B)

water_levels <- sort(unique(df_exp$water))

for (b in seq_len(B)) {
  dat_perm <- df_exp

  # Permute water inside each water stratum
  for (s in water_levels) {
    water_ind <- which(df_exp$water == s)
    dat_perm$sunlight[water_ind] <- sample(df_exp$sunlight[water_ind])
  }

  T_perm[b] <- sun_effect_stat(dat_perm)
}

# One sided p-value (sun increases freshness)
p_one_sided <- (sum(T_perm >= T_obs) + 1) / (B + 1)
print(p_one_sided)

```

```
## [1] 0.2568743
```

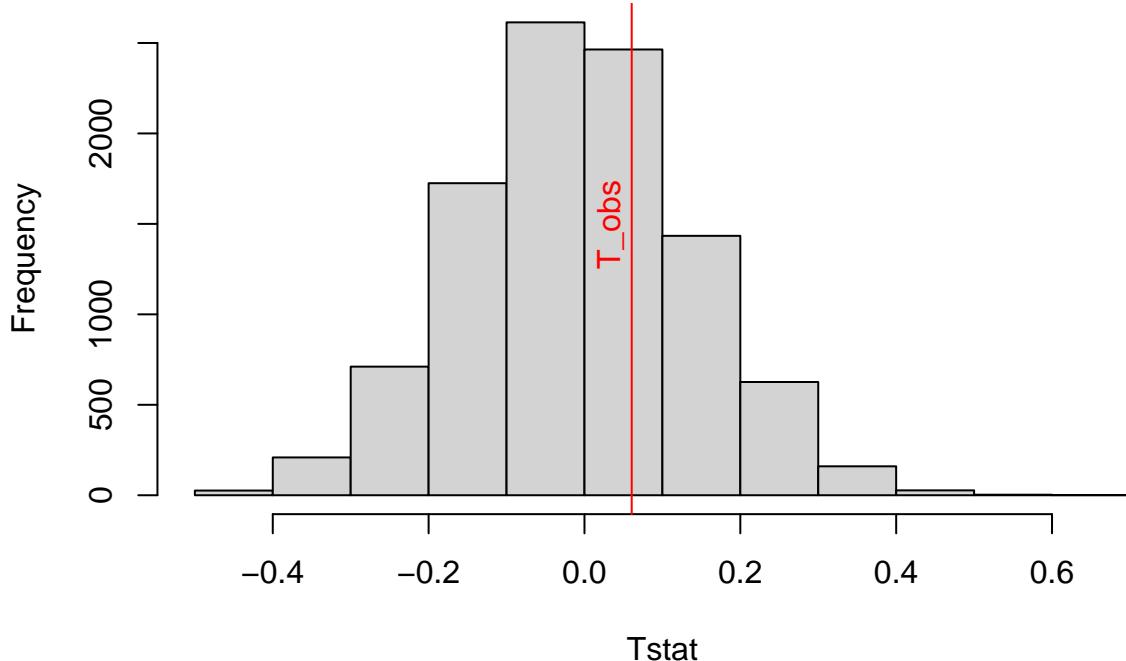
Evidently, sunlight is not associated with an increase in freshness. Once again, we produce a histogram illustrating the distribution of our statistic and our observed statistic value:

```

hist(T_perm,
      main = "Distribution of T statistic - Sunlight",
      xlab = "Tstat")
abline(v = T_obs, col="red")
text(x = 0.0115152, y = 1250, labels = "T_obs", srt = 90, pos = 4, col = "red")

```

## Distribution of T statistic – Sunlight



### Difference in Proportions Test - Fridge/Darkness no water

Typically, people think it is best to store plants in the fridge as opposed to one of the experimental setups considered in this lab. Thus, as a final permutation test, we consider the difference in proportions between those green onions from the fridge with those from the seemingly best condition: darkness with no water.

We first subset the data for these labels:

```
is_fridge_nowater <- grep("FN[0-9]+$", proj3_data$label)
is_darkness_water <- grep("DW[0-9]+$", proj3_data$label)

df_final <- proj3_data[is_fridge_nowater | is_darkness_water, , drop = FALSE]

# Create a two-level group variable
df_final$group <- ifelse(grep("FN", df_final$label),
                         "fridge",
                         "dark_water")
```

Next, we can compute our observed difference in proportions

```
mean_fridge <- mean(df_final$freshness[df_final$group == "fridge"])
mean_dark <- mean(df_final$freshness[df_final$group == "dark_water"])

T_obs <- mean_dark - mean_fridge
T_obs
```

```

## [1] 0.1666667

set.seed(123)
B <- 10000
T_perm <- numeric(B)

for (b in seq_len(B)) {
  # randomly permute which samples are called "fridge" vs "dark_water"
  permuted_group <- sample(df_final$group)

  mean_fridge_b <- mean(df_final$freshness[permuted_group == "fridge"])
  mean_dark_b   <- mean(df_final$freshness[permuted_group == "dark_water"])

  T_perm[b] <- mean_dark_b - mean_fridge_b
}

p_one_sided <- (sum(T_perm >= T_obs) + 1) / (B + 1)
p_one_sided

```

```
## [1] 0.3243676
```

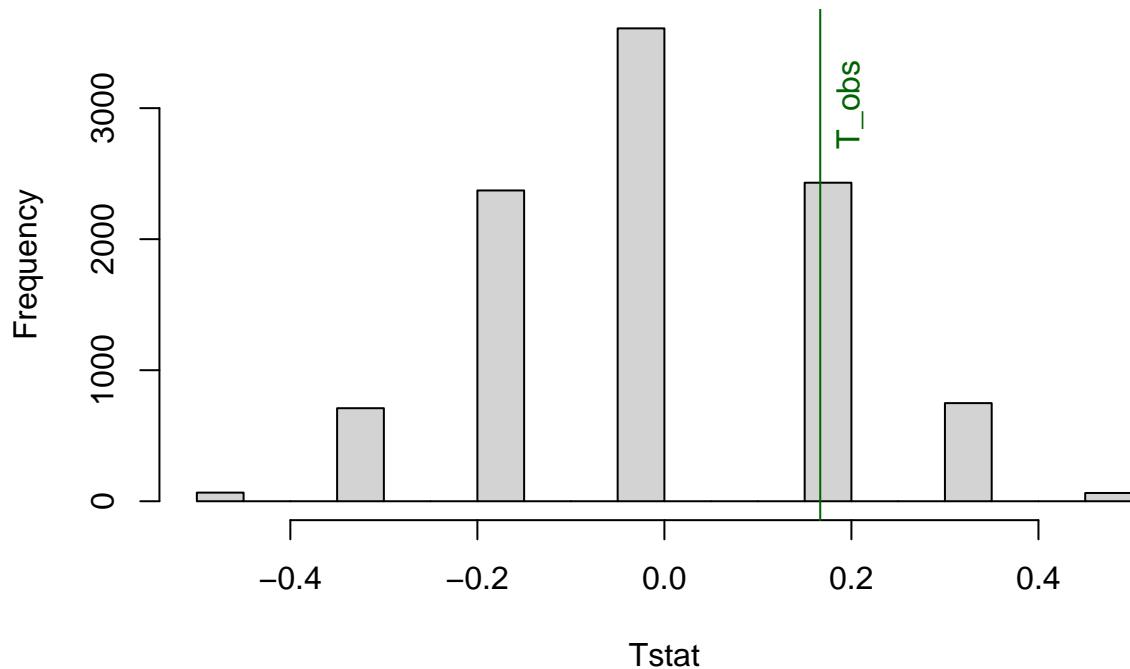
Our observed difference suggests that green onions left in the darkness with water are more fresh than those green onions in the fridge with no water. However, the p-value suggests that there is not a statistically significant difference between these groups.

```

hist(T_perm,
      main = "Distribution of T statistic - Dark + Water versus Fridge",
      xlab = "Tstat")
abline(v = T_obs, col="darkgreen")
text(x = 0.18, y = 2690, labels = "T_obs", srt = 90, pos = 4, col = "darkgreen")

```

## Distribution of T statistic – Dark + Water versus Fridge



```

cond_table <- aggregate(
  freshness ~ sunlight + water,
  data = df_exp,
  FUN = function(x) c(mean = mean(x), n = length(x))
)

# aggregate() packs mean and n into a matrix column; unpack it nicely:
cond_summary <- data.frame(
  sunlight = cond_table$sunlight,
  water    = cond_table$water,
  n        = cond_table$freshness[, "n"],
  fresh_rate = cond_table$freshness[, "mean"]
)

# Add readable labels
cond_summary$light_label <- ifelse(cond_summary$sunlight == 1, "Sunlight", "Darkness")
cond_summary$water_label <- ifelse(cond_summary$water == 1, "Water", "No Water")

cat("Freshness rates by condition:\n")

## Freshness rates by condition:

for (i in seq_len(nrow(cond_summary))) {
  cat(sprintf(
    "%s, %s: n=%d, mean freshness=%.3f\n",

```

```

    cond_summary$light_label[i] ,
    cond_summary$water_label[i] ,
    cond_summary$n[i] ,
    cond_summary$fresh_rate[i]
  )))
}

## Darkness, No Water: n=12, mean freshness=0.167
## Sunlight, No Water: n=8, mean freshness=0.500
## Darkness, Water: n=12, mean freshness=0.833
## Sunlight, Water: n=12, mean freshness=0.667

```

## Height Permutation Tests

In addition, we are also interested in the final (Day 9) length of the onions.

```

# Keep only SW/SN/DW/DN rows (pooling all labs)
experimental_condition <- grep1("^(SW|SN|DW|DN)", proj3_data$label)
df_exp <- proj3_data[experimental_condition, , drop = FALSE]

# One-hot style indicators aligned with partner's code
df_exp$sunlight <- ifelse(substr(df_exp$label, 1, 1) == "S", 1L, 0L)      # 1=Sun, 0=Shade
df_exp$water     <- ifelse(substr(df_exp$label, 2, 2) == "W", 1L, 0L)      # 1=Water, 0=NoWater

# Outcome
stopifnot("day_9_len" %in% names(df_exp))

```

## EDA (Height)

Medians of day\_9\_len by cell:

```

df_exp$cell <- paste0(ifelse(df_exp$sunlight==1,"S","D"),
                      ifelse(df_exp$water==1,"W","N"))
tab_n <- table(df_exp$cell)
tab_med <- tapply(df_exp$day_9_len, df_exp$cell, median, na.rm = TRUE)
data.frame(cell = names(tab_n), n = as.integer(tab_n), median_day9 = as.numeric(tab_med))

##   cell  n median_day9
## 1  DN 12    12.05
## 2  DW 12    28.45
## 3  SN  8    12.65
## 4  SW 12    21.20

```

We ran three permutation tests for the final (Day 9) length of the onions:

- Assess potential difference between green onions that received water and those that did not while conditioning on sunlight.
- Assess potential difference between green onions that received sunlight and those that did not while conditioning on water.

- Assess potential difference between those that received water in the shade versus those that were placed in the fridge without water.

Let  $Y$  denote day-9 length (`day_9_len`). For a binary group indicator  $G \in \{0, 1\}$  and a binary conditioning variable (stratum)  $S \in \{0, 1\}$ , define the within-stratum difference in medians

$$\Delta_s = \text{median}\{Y \mid G = 1, S = s\} - \text{median}\{Y \mid G = 0, S = s\}, \quad s \in \{0, 1\}.$$

With  $n_s$  the number of observations in stratum  $s$ , the (observed) weighted statistic is

$$T_{\text{obs}} = \frac{\sum_s n_s \Delta_s}{\sum_s n_s}.$$

```
# HELPERS

diff_in_medians <- function(y, g) {
  y <- as.numeric(y)
  g <- as.integer(g)
  median(y[g==1], na.rm=TRUE) - median(y[g==0], na.rm=TRUE)
}

weighted_strat_median_stat <- function(input_data, grp, strat, outcome) {
  levels_s <- sort(unique(input_data[[strat]]))
  diffss <- numeric(length(levels_s))
  wts <- numeric(length(levels_s))
  for (i in seq_along(levels_s)) {
    s <- levels_s[i]
    sub <- input_data[input_data[[strat]] == s, , drop = FALSE]
    diffss[i] <- diff_in_medians(sub[[outcome]], sub[[grp]])
    wts[i] <- nrow(sub)
  }
  sum(diffss * wts) / sum(wts)
}

perm_test_stratified_median <- function(input_data, grp, strat, outcome,
  B = 10000L, seed = 123, alternative = c("greater", "less")) {
  alternative <- match.arg(alternative)
  set.seed(seed)

  T_obs <- weighted_strat_median_stat(input_data, grp, strat, outcome)

  T_perm <- numeric(B)
  strata_levels <- sort(unique(input_data[[strat]]))
  for (b in seq_len(B)) {
    dat_perm <- input_data
    for (s in strata_levels) {
      idx <- which(input_data[[strat]] == s)
      dat_perm[[grp]][idx] <- sample(input_data[[grp]][idx])
    }
    T_perm[b] <- weighted_strat_median_stat(dat_perm, grp, strat, outcome)
  }

  if (alternative == "greater") {
    p <- (sum(T_perm >= T_obs) + 1) / (B + 1)
  } else {
    p <- (sum(T_perm <= T_obs) + 1) / (B + 1)
  }
}
```

```

} else {
  p <- (sum(T_perm <= T_obs) + 1) / (B + 1)
}

list(T_obs = T_obs, T_perm = T_perm, p_value = p, alternative = alternative)
}

```

### Test 1: Water Effect (conditioning on Sunlight)

- **Goal:** assess whether watering increases height.
- **Null  $H_0$ :** within each sunlight stratum, the median heights for watered and unwatered are equal.
- **Alternative  $H_1$ :** watered onions have larger medians (one-sided, “greater”).
- **Statistic:** for each sunlight level  $s$ , set  $G = \#\{\text{Water} = 1\}$ , so  $\Delta_s = \text{median}_{W=1} - \text{median}_{W=0}$ ; combine to  $T_{\text{obs}}$  as above.

### Permutation procedure.

Within each sunlight stratum  $s$ , randomly permute the water labels (preserving group sizes) and recompute  $T$  to obtain  $\{T_b\}_{b=1}^B$  under  $H_0$ .

#### One-sided p-value (right tail).

$$p = \frac{\#\{T_b \geq T_{\text{obs}}\} + 1}{B + 1}.$$

```

df_exp$grp_water <- df_exp$water
df_exp$strat_sun <- df_exp$sunlight

set.seed(82803)
B <- 10000L
water_res <- perm_test_stratified_median(
  input_data = df_exp,
  grp = "grp_water",
  strat = "strat_sun",
  outcome = "day_9_len",
  B = B,
  seed = 82803,
  alternative = "greater"
)

cat(sprintf("Water effect - Observed T: %.4f\n", water_res$T_obs))

## Water effect - Observed T: 12.8318

cat(sprintf("Water effect - One-sided p-value: %s\n", sprintf('%.2e', water_res$p_value)))

## Water effect - One-sided p-value: 1.00e-04

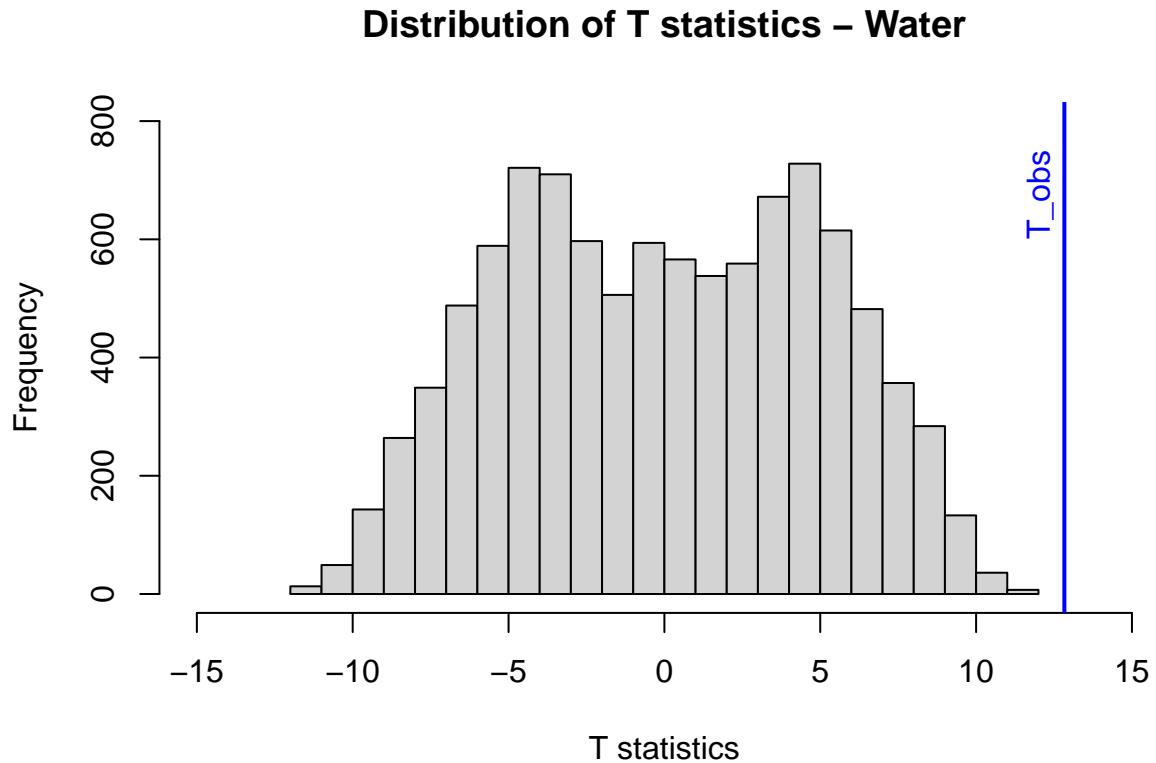
```

The p-value suggests there is very strong evidence that, within each sunlight level, watering increases the median day-9 height. We can also produce a histogram for the distribution of permuted statistics. This is illustrated below:

```

hist(water_res$T_perm, main = "Distribution of T statistics - Water",
      xlab = "T statistics", breaks=25, xlim = c(-15, 15), ylim = c(0,800))
abline(v = water_res$T_obs, col = "blue", lwd = 2)
text(x = 11.5, y = 600, labels = "T_obs", srt = 90, pos = 4, col = "blue")

```



### Test 2: Sunlight Effect (conditioning on Water)

- **Goal:** assess whether shade yields greater height than sun (conditional on water).
- **Null  $H_0$ :** within each water stratum, the median heights for sun and shade are equal.
- **Alternative  $H_1$ :** shade has larger medians. Since the statistic is (Sun – Shade), this is a one-sided “less” test.
- **Statistic:** for each water level  $w$ , set  $G = \mathbb{1}\{\text{Sun} = 1\}$  so that  $\Delta_w = \text{median}_{\text{Sun}} - \text{median}_{\text{Shade}}$ , and combine as

$$T_{\text{obs}} = \frac{\sum_w n_w \Delta_w}{\sum_w n_w}.$$

#### Permutation procedure.

Within each water stratum  $w$ , randomly permute the sunlight labels, recompute  $T$  to get  $\{T_b\}_{b=1}^B$ .

#### One-sided p-value (left tail).

$$p = \frac{\#\{T_b \leq T_{\text{obs}}\} + 1}{B + 1}.$$

```

df_exp$grp_sun <- df_exp$sunlight
df_exp$strat_w <- df_exp$water

set.seed(82803)
B <- 10000L
sun_res <- perm_test_stratified_median(
  input_data = df_exp,
  grp = "grp_sun",
  strat = "strat_w",
  outcome = "day_9_len",
  B = B,
  seed = 82803,
  alternative = "less"
)

cat(sprintf("Sunlight effect - Observed T: %.4f\n", sun_res$T_obs))

## Sunlight effect - Observed T: -3.6818

cat(sprintf("Sunlight effect - One-sided p-value: %s\n", sprintf('%.2e', sun_res$p_value)))

## Sunlight effect - One-sided p-value: 1.32e-02

```

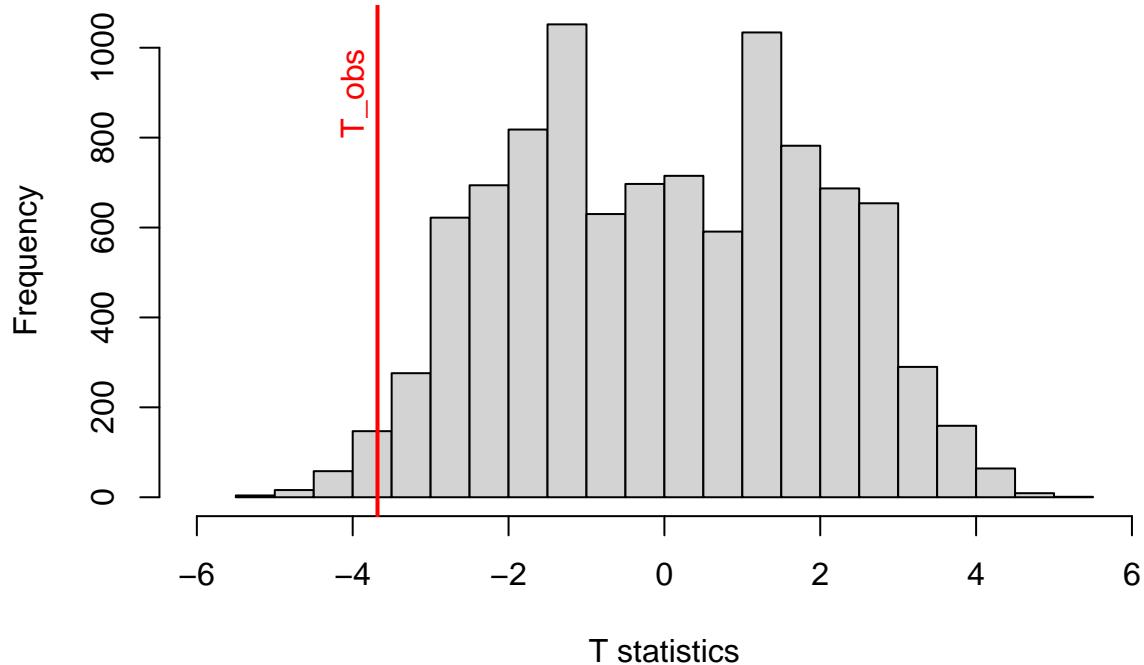
The p-value suggests there is very strong evidence that, within each water condition, sunlight decreases the median day-9 height. We can also produce a histogram for the distribution of permuted statistics. This is illustrated below:

```

hist(sun_res$T_perm, main = "Null distribution: Light effect (median, strat by Water)",
      xlab = "T statistics", xlim = c(-6,6))
abline(v = sun_res$T_obs, col = "red", lwd = 2)
text(x = -4.2, y = 800, labels = "T_obs", srt = 90, pos = 4, col = "red")

```

## Null distribution: Light effect (median, strat by Water)



### Test 3: DW vs FN comparison

We compare **Darkness + Water (DW)** against **Fridge + No Water (FN)** using the difference in medians and a **right-tailed** permutation test (H1: DW > FN).

```
## =====
## Darkness + Water (DW) vs Fridge + No Water (FN)
## =====

# Subset relevant conditions
is_DW <- grep("^\w{0-9}+$", proj3_data$label)
is_FN <- grep("^\w{0-9}+$", proj3_data$label)
df_dwfn <- proj3_data[is_DW | is_FN, , drop = FALSE]

# Create two-level group variable
df_dwfn$group <- ifelse(grep("^\w{0-9}+$", df_dwfn$label), "dark_water", "fridge_nowater")

# Outcome variable
stopifnot("day_9_len" %in% names(df_dwfn))
y <- df_dwfn$day_9_len

# Observed difference in medians (Darkness+Water - Fridge+NoWater)
T_obs <- median(y[df_dwfn$group == "dark_water"], na.rm = TRUE) -
  median(y[df_dwfn$group == "fridge_nowater"], na.rm = TRUE)
cat("\n==== Darkness + Water vs Fridge + No Water ===\n")
```

```

##  

## === Darkness + Water vs Fridge + No Water ===  

cat(sprintf("Observed diff in medians (DW - FN): %.4f\n", T_obs))  

## Observed diff in medians (DW - FN): 16.2500  

# Permutation test (one-sided, H1: DW > FN)  

set.seed(123)  

B <- 10000  

T_perm <- numeric(B)  

for (b in seq_len(B)) {  

  perm_group <- sample(df_dwfn$group)  

  T_perm[b] <- median(y[perm_group == "dark_water"], na.rm = TRUE) -  

    median(y[perm_group == "fridge_nowater"], na.rm = TRUE)  

}  

p_one_sided <- (sum(T_perm >= T_obs) + 1) / (B + 1)  

cat(sprintf("One-sided p-value (H1: DW taller than FN): %s\n",
            sprintf('%.2e', p_one_sided)))  

## One-sided p-value (H1: DW taller than FN): 2.00e-04

```

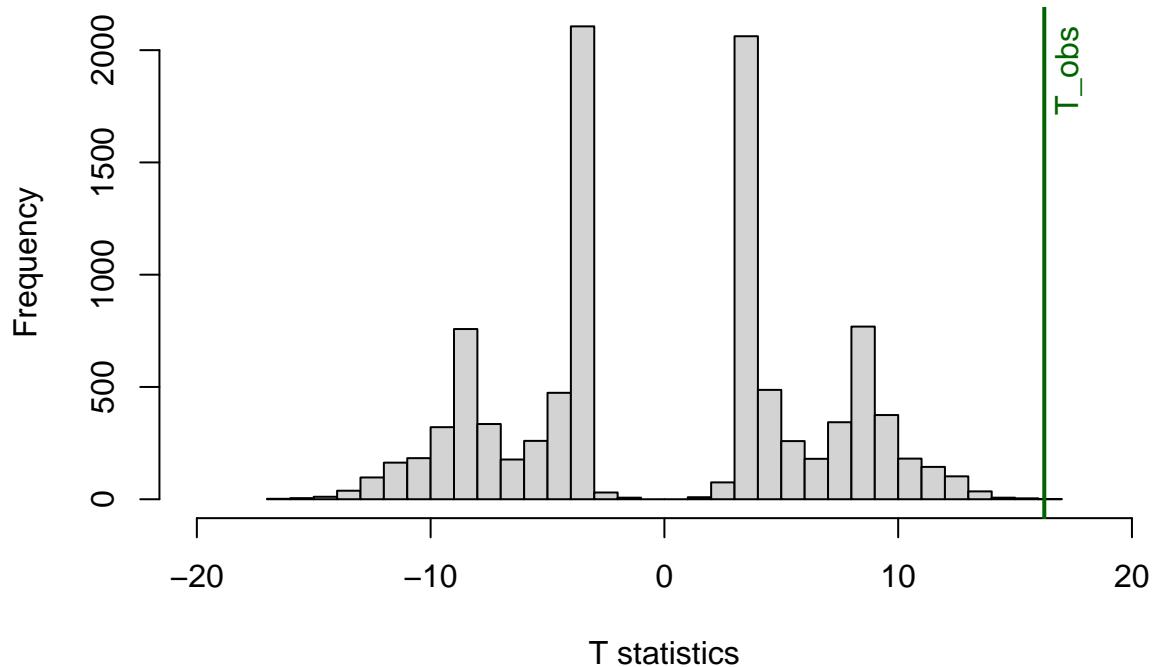
The p-value suggests there is very strong evidence that onions kept in darkness with water (DW) are taller by day 9 than those stored in a fridge with no water (FN). We can also produce a histogram for the distribution of permuted statistics. This is illustrated below:

```

# Visualization
hist(T_perm,
      main = "Null distribution: DW vs FN (median height)",
      xlab = "T statistics", xlim=c(-20,20),
      breaks = 25)
abline(v = T_obs, col = "darkgreen", lwd = 2)
text(x = T_obs+.3, y = max(hist(T_perm, plot = FALSE)$counts)*0.8,
      labels = "T_obs", srt = 90, pos = 4, col = "darkgreen")

```

## Null distribution: DW vs FN (median height)



### Controlling Site-level environment

To control for site-level environment, we re-conduct the above tests by further stratifying by **Lab**. Strata missing a level are excluded from the statistic. As we shall see below, we still have small p-values for all the tests.

```
## ---- helper: filter degenerate strata (needs df_exp, grp, strat) ----
filter_valid_strata <- function(dat, grp, strat) {
  # count #obs at each (stratum, group)
  tab <- with(dat, table(dat[[strat]], dat[[grp]]))
  # keep strata with both groups present (>=1 in each column)
  keep_levels <- rownames(tab)[rowSums(tab > 0) == ncol(tab)]
  dropped <- setdiff(levels(factor(dat[[strat]])), keep_levels)
  dat_keep <- dat[dat[[strat]] %in% keep_levels, , drop = FALSE]
  list(data = dat_keep, kept = keep_levels, dropped = dropped)
}

## ---- TEST 1: Water effect / Lab × Sunlight (with auto-drop of empty cells) ----
df_exp$grp_water      <- df_exp$water                      # 1 = Water, 0 = NoWater
df_exp$strat_lab_sun <- interaction(df_exp$lab, df_exp$sunlight, drop = TRUE)

# filter out strata that don't have BOTH water levels
flt <- filter_valid_strata(df_exp, grp = "grp_water", strat = "strat_lab_sun")
if (length(flt$dropped)) {
  message("Dropped strata (missing a water level): ",
```

```

        paste(flt$dropped, collapse = ", "))
}

## Dropped strata (missing a water level): A.1

df_ws <- flt$data

# if nothing left, stop gracefully
if (nrow(df_ws) == 0L || length(unique(df_ws$strat_lab_sun)) == 0L) {
  stop("No valid Lab × Sunlight strata contain both Water and NoWater. Cannot run the test.")
}

set.seed(82803)
water_lab_sun_res <- perm_test_stratified_median(
  input_data = df_ws,
  grp = "grp_water",
  strat = "strat_lab_sun",
  outcome = "day_9_len",
  B = 10000L,
  seed = 82803,
  alternative = "greater"
)

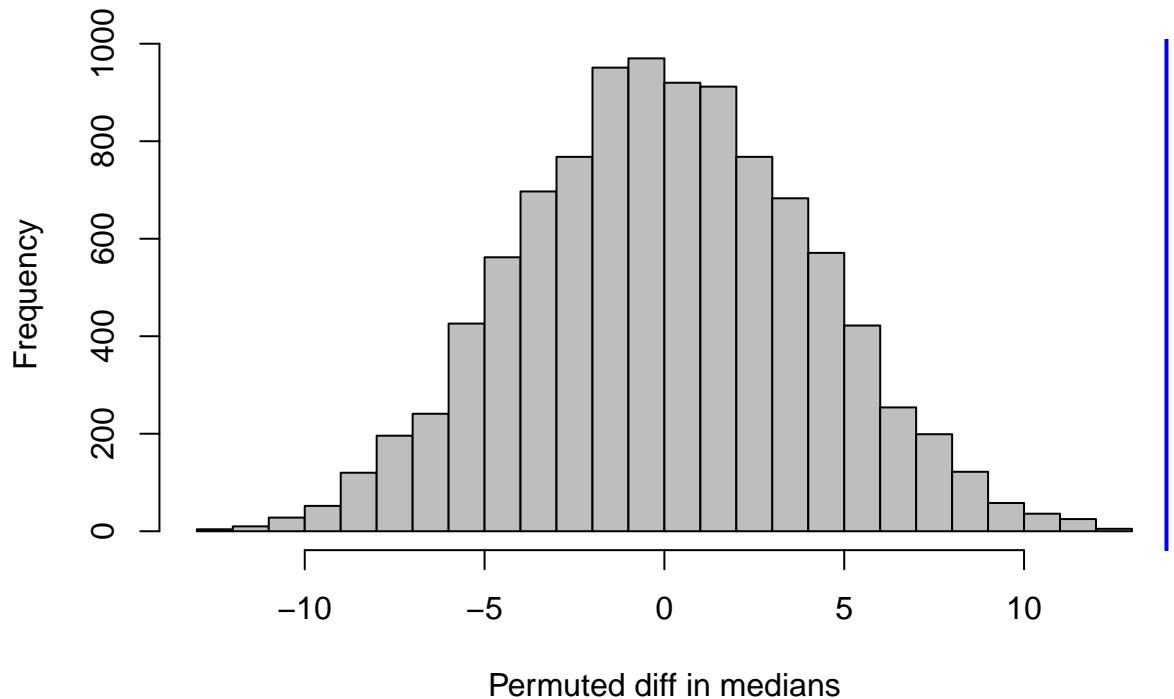
cat(sprintf("Water effect | Lab × Sunlight strata - T_obs: %.4f, p = %s\n",
            water_lab_sun_res$T_obs, sprintf('%.2e', water_lab_sun_res$p_value)))

## Water effect | Lab × Sunlight strata - T_obs: 13.9600, p = 1.00e-04

if (length(water_lab_sun_res$T_perm) > 0L && is.finite(water_lab_sun_res$T_obs)) {
  hist(water_lab_sun_res$T_perm,
        main = "Null distribution: Water effect (stratified by Lab × Sunlight)",
        xlab = "Permuted diff in medians", col = "grey", breaks = 25)
  abline(v = water_lab_sun_res$T_obs, col = "blue", lwd = 2)
} else {
  message("Permutation vector is empty after filtering; no histogram drawn.")
}

```

## Null distribution: Water effect (stratified by Lab x Sunlight)



```
## ---- TEST 2: Sunlight effect / Lab x Water (with auto-drop of empty cells) ----

# Helper (only if not already defined)
if (!exists("filter_valid_strata")) {
  filter_valid_strata <- function(dat, grp, strat) {
    tab <- with(dat, table(dat[[strat]], dat[[grp]]))
    keep_levels <- rownames(tab)[rowSums(tab > 0) == ncol(tab)]
    dropped <- setdiff(levels(factor(dat[[strat]])), keep_levels)
    dat_keep <- dat[dat[[strat]] %in% keep_levels, , drop = FALSE]
    list(data = dat_keep, kept = keep_levels, dropped = dropped)
  }
}

# 1 = Sun, 0 = Shade; alternative "less" since stat = Sun - Shade and H1 is Shade > Sun
df_exp$grp_sun           <- df_exp$sunlight
df_exp$strat_lab_water <- interaction(df_exp$lab, df_exp$water, drop = TRUE)

# Filter out degenerate strata (those missing either Sun or Shade)
flt2 <- filter_valid_strata(df_exp, grp = "grp_sun", strat = "strat_lab_water")
if (length(flt2$dropped)) {
  message("Dropped strata (missing a Sun/Shade level): ",
         paste(flt2$dropped, collapse = ", "))
}

## Dropped strata (missing a Sun/Shade level): A.0
```

```

df_sw <- flt2$data

# If no valid strata remain, fail gracefully
if (nrow(df_sw) == 0L || length(unique(df_sw$strat_lab_water)) == 0L) {
  stop("No valid Lab x Water strata contain both Sun and Shade; cannot run the Sunlight test.")
}

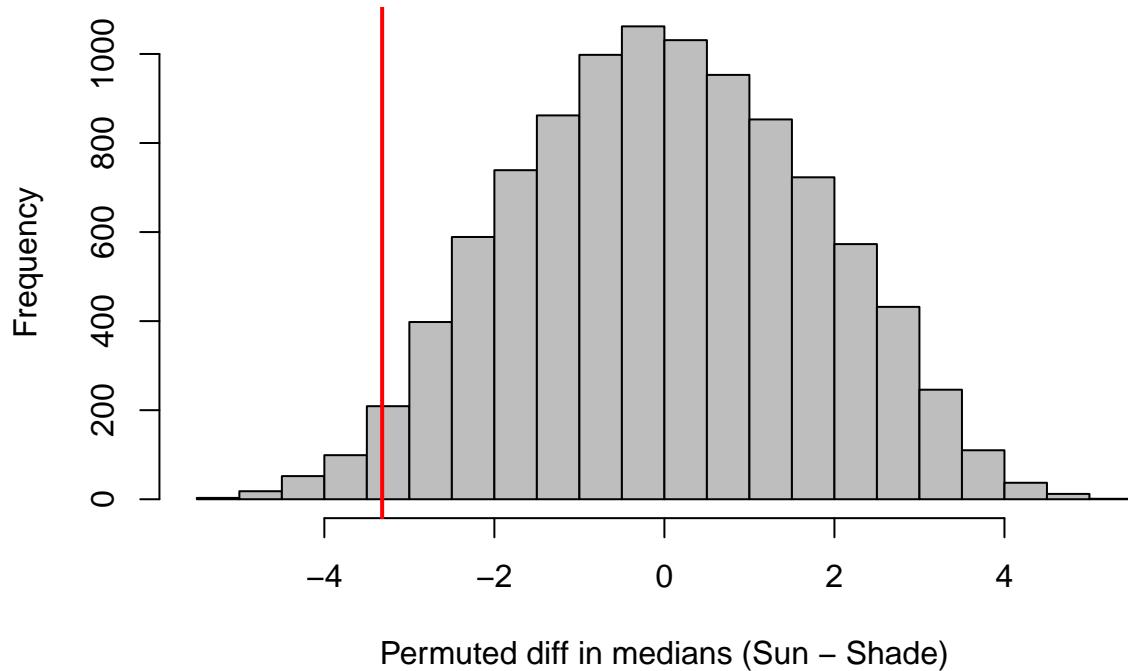
set.seed(82803)
B <- 10000L
sun_lab_water_res <- perm_test_stratified_median(
  input_data = df_sw,
  grp        = "grp_sun",           # 1 = Sun, 0 = Shade
  strat      = "strat_lab_water",   # strata: Lab x Water
  outcome    = "day_9_len",
  B          = B,
  seed       = 82803,
  alternative = "less"            # H1: Shade > Sun
)

cat(sprintf("Sunlight effect | Lab x Water strata - T_obs: %.4f, p = %s\n",
            sun_lab_water_res$T_obs, sprintf('%.2e', sun_lab_water_res$p_value)))
## Sunlight effect | Lab x Water strata - T_obs: -3.3200, p = 2.38e-02

if (length(sun_lab_water_res$T_perm) > 0L && is.finite(sun_lab_water_res$T_obs)) {
  hist(sun_lab_water_res$T_perm,
    main = "Null distribution: Sunlight effect (Lab x Water strata)",
    xlab = "Permuted diff in medians (Sun - Shade)", col = "grey", breaks = 25)
  abline(v = sun_lab_water_res$T_obs, col = "red", lwd = 2)
} else {
  message("Permutation vector is empty after filtering; no histogram drawn.")
}

```

## Null distribution: Sunlight effect (Lab x Water strata)



```
## ---- Test 3: DW vs FN (median length) | Stratified by Lab, one-sided (H1: DW > FN) ----

# Helper (define if not already defined above)
if (!exists("filter_valid_strata")) {
  filter_valid_strata <- function(dat, grp, strat) {
    tab <- with(dat, table(dat[[strat]], dat[[grp]]))
    keep_levels <- rownames(tab)[rowSums(tab > 0) == ncol(tab)]
    dropped <- setdiff(levels(factor(dat[[strat]])), keep_levels)
    dat_keep <- dat[dat[[strat]] %in% keep_levels, , drop = FALSE]
    list(data = dat_keep, kept = keep_levels, dropped = dropped)
  }
}

# Subset to DW and FN rows from the original data
is_DW <- grepl("^\$DW[0-9]+\$", proj3_data$label)
is_FN <- grepl("^\$FN[0-9]+\$", proj3_data$label)
df_dwfn <- proj3_data[is_DW | is_FN, , drop = FALSE]

# Group: 1 = DW (Darkness + Water), 0 = FN (Fridge + No Water)
df_dwfn$grp_dw <- ifelse(grepl("^\$DW\$\$", df_dwfn$label), 1L, 0L)
df_dwfn$strat_lab <- factor(df_dwfn$lab)

stopifnot("day_9_len" %in% names(df_dwfn))

# Drop labs that don't contain BOTH groups (DW and FN)
flt <- filter_valid_strata(df_dwfn, grp = "grp_dw", strat = "strat_lab")
```

```

if (length(flt$dropped)) {
  message("Dropped labs (missing DW or FN): ", paste(flt$dropped, collapse = ", "))
}
df_df <- flt$data

if (nrow(df_df) == OL || length(unique(df_df$strat_lab)) == OL) {
  stop("No lab has both DW and FN observations. Cannot run DW vs FN test.")
}

# One-sided stratified permutation test (median), H1: DW > FN => "greater"
set.seed(123)
B <- 10000L
dw_fn_lab_res <- perm_test_stratified_median(
  input_data = df_df,
  grp = "grp_dw",           # 1=DW, 0=FN
  strat = "strat_lab",      # strata: Lab
  outcome = "day_9_len",
  B = B,
  seed = 123,
  alternative = "greater"
)

cat("\n== DW vs FN (median length) | stratified by Lab ==\n")

## == DW vs FN (median length) | stratified by Lab ==

cat(sprintf("Observed diff in medians (DW - FN): %.4f\n", dw_fn_lab_res$T_obs))

## Observed diff in medians (DW - FN): 15.4667

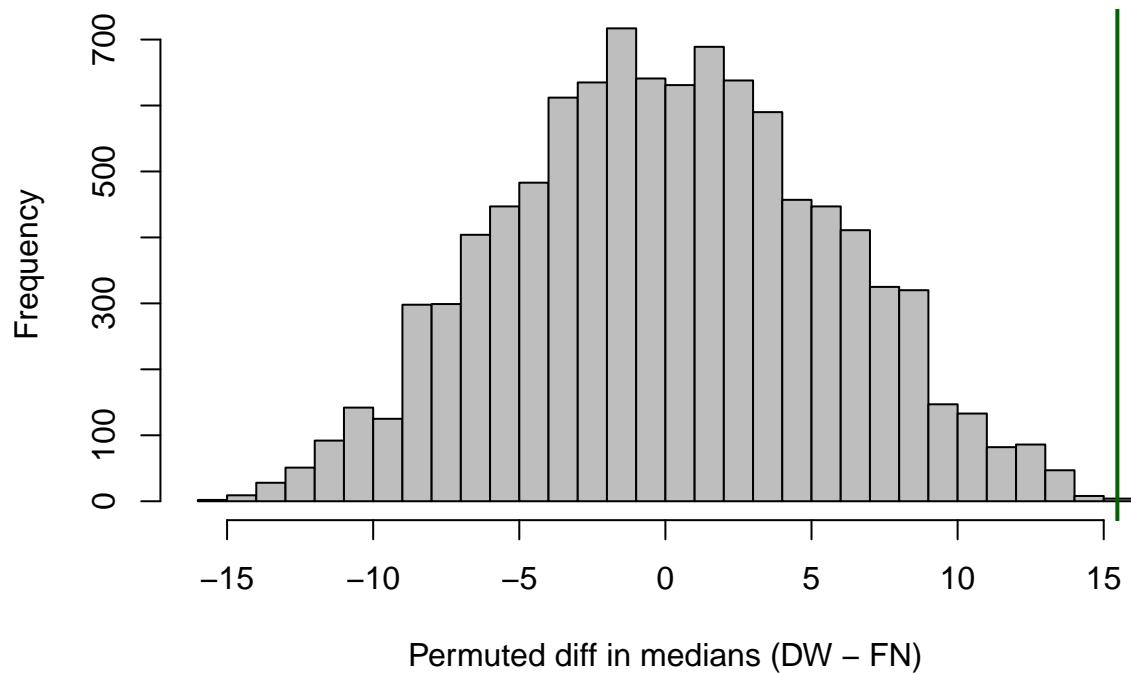
cat(sprintf("One-sided p-value: %s\n", sprintf('%.2e', dw_fn_lab_res$p_value)))

## One-sided p-value: 1.00e-04

# Visualization
if (length(dw_fn_lab_res$T_perm) > OL && is.finite(dw_fn_lab_res$T_obs)) {
  hist(dw_fn_lab_res$T_perm,
       main = "Null distribution: DW vs FN (median), stratified by Lab",
       xlab = "Permuted diff in medians (DW - FN)", breaks = 25, col = "grey")
  abline(v = dw_fn_lab_res$T_obs, col = "darkgreen", lwd = 2)
} else {
  message("Permutation vector is empty after filtering; no histogram drawn.")
}

```

## Null distribution: DW vs FN (median), stratified by Lab



### Conclusions:

- From EDA, we observed a clear association between freshness and height — fresher onions tended to be taller.
- Overall, these findings consistently indicate that access to water and protection from direct sunlight both contribute positively to onion growth, while refrigeration (without water) inhibits it.
- We recorded the lengths of the green onions every three days throughout the experiment. While the present analysis focuses on the day-9 measurements, further analyses could be conducted to study growth trajectories over time.