

V1075 Monte Carlo Report

Contents

1	Setup	1
2	Bootstrapping $\delta^{18}O_p$ by Taxon	2
3	Water $\delta^{18}O_{sw}$ Simulations	4
4	Water Temperature Simulations	7
5	Air Temperature Simulations	9

Purpose: This report performs Monte Carlo simulations to quantify uncertainty in estimates of $\delta^{18}O_{surface.water}$ ($\delta^{18}O_{sw}$) and paleotemperature derived from $\delta^{18}O_{phosphate}$ ($\delta^{18}O_p$) values of vertebrate fossils collected from the V1075 bonebed of the Cloverly Formation.

Structure: The document includes steps for data loading, bootstrapping of $\delta^{18}O_p$, simulation of $\delta^{18}O_{sw}$ and paleotemperature distributions, and uncertainty quantification.

Reproducibility: To run this report, open the associated RStudio Project file (Frontiers_V1075_Project.Rproj). This automatically sets the correct working directory. Then run this .Rmd file.

Repository: All data and code used are provided at: github.com/.../Frontiers_V1075_Project → Start by reading the README.md file in that directory.

Citation: Allen et al., 2025. *Frontiers in Earth Science*, DOI: 10.3389/feart.2025.1497416

1 Setup

```
# Define directories using here::here()
# Correct project-relative paths
data_dir <- here("data")
results_dir <- here("results")
models_dir <- here("models")

# Read in data
```

```

V1075_MCdata <- read_csv(file.path(data_dir, "V1075MC_data.csv"))
NIST120c <- read_csv(file.path(data_dir, "V1075_NIST120c.csv"))

# Load models using project-root-relative paths
Amiot_lm <- readRDS(here("models", "Amiot_reg_lm.rds"))
Barrick_lm <- readRDS(here("models", "Barrick_reg_lm.rds"))
PLN_lm <- readRDS(here("models", "PLNd180p_reg_lm.rds"))
TwTa_lm <- readRDS(here("models", "TwTa_reg_lm.rds"))

# Set constants
n_iterations <- 1000
set.seed(123)

```

2 Bootstrapping $\delta^{18}O_p$ by Taxon

```

# Subset taxa into a list of vectors
taxa_list <- list(
  Lepisosteidae      = subset(V1075_MCdata, Taxon == "Lepisosteids")$d180,
  Hybodontiformes    = subset(V1075_MCdata, Taxon == "Hybodonts")$d180,
  `Glyptops sp.`     = subset(V1075_MCdata, Taxon == "Glyptops sp.")$d180,
  `Naomichelys sp.`  = subset(V1075_MCdata, Taxon == "Naomichelys sp.")$d180,
  `Neosuchian G`     = subset(V1075_MCdata, Taxon == "Neosuchian G")$d180,
  `Neosuchian A`     = subset(V1075_MCdata, Taxon == "Neosuchian A")$d180,
  `Neosuchian B`     = subset(V1075_MCdata, Taxon == "Neosuchian B")$d180,
  NIST120c_aka_NBS120c = NIST120c$d.180.160
)

# Define bootstrap function
taxon_bootstrap <- function(values) {
  data.frame(means = replicate(n_iterations, mean(sample(values, replace = TRUE))))
}

# Apply bootstrap to each taxon
bootstrap_results <- lapply(taxa_list, taxon_bootstrap)

# Name the results list for easy access
names(bootstrap_results) <- names(taxa_list)

bootstrap_results <- lapply(taxa_list, taxon_bootstrap)

# Combine bootstrap data
combined_data <- bind_rows(
  lapply(names(bootstrap_results), function(t) {
    tibble(group = t, values = bootstrap_results[[t]]$means)
  })
)

# Summarize d180p
summary_d180p <- combined_data %>%
  group_by(group) %>%
  summarise(

```

```

    Mean_d18Op = mean(values),
    CI_Lower = quantile(values, 0.025, na.rm = TRUE),
    CI_Upper = quantile(values, 0.975, na.rm = TRUE)
  )
summary_d18Op

```

```

## # A tibble: 8 x 4
##   group          Mean_d18Op CI_Lower CI_Upper
##   <chr>          <dbl>     <dbl>   <dbl>
## 1 Glyptops sp.      13.8      12.6    14.9
## 2 Hybodontiformes  12.0       9.77   15.3
## 3 Lepisosteidae     13.8      13.0    14.5
## 4 NIST120c_aka_NBS120c 22.6      22.4    22.7
## 5 Naomichelys sp.   14.3      14.0    14.7
## 6 Neosuchian A      17.1      16.0    18.5
## 7 Neosuchian B      16.6      15.5    17.6
## 8 Neosuchian G      14.0      12.4    15.5

```

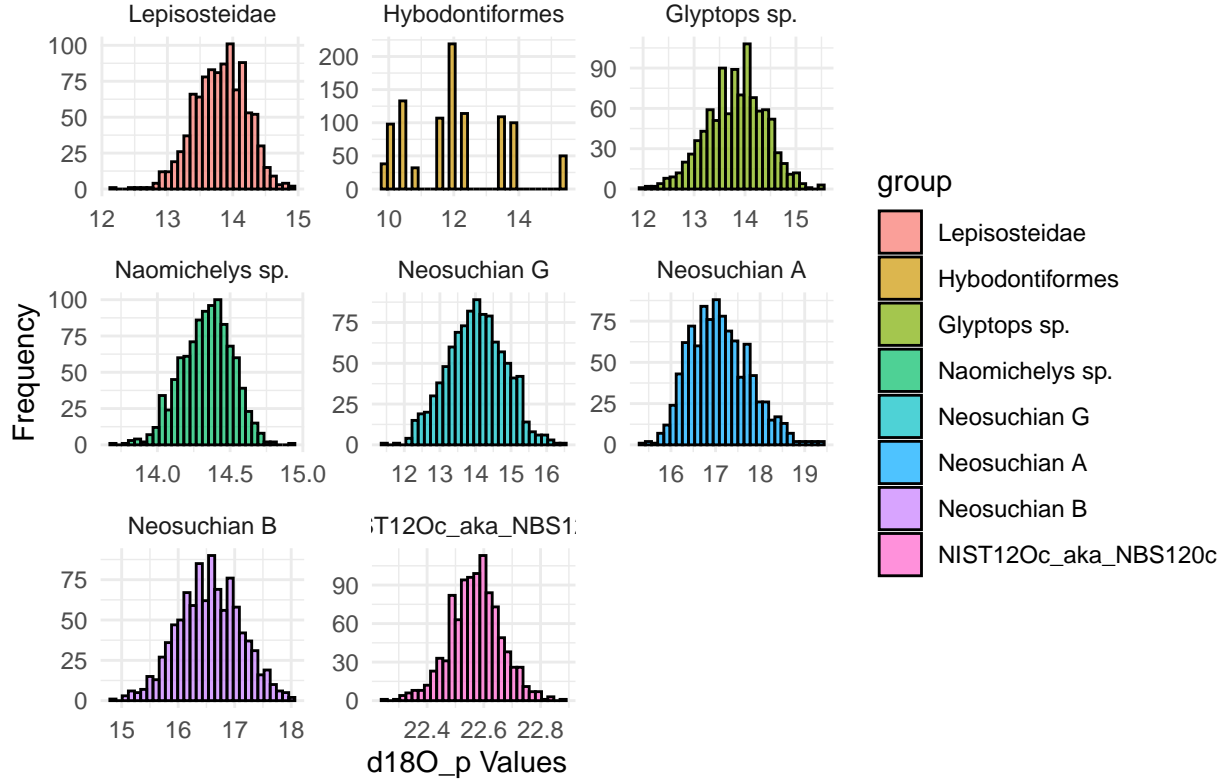
```

# Prepare to plot histogram of d18Op simulations
combined_data$group <- factor(combined_data$group,
  levels = names(taxa_list))

# Plot
ggplot(combined_data, aes(x = values, fill = group)) +
  geom_histogram(position = "identity", alpha = 0.7, bins = 30, color = "black") +
  labs(title = "Histogram of Resampled Means", x = "d180_p Values", y = "Frequency") +
  theme_minimal() +
  facet_wrap(~group, scales = "free")

```

Histogram of Resampled Means



3 Water $\delta^{18}O_{sw}$ Simulations

```
# Create d18Osw simulation function
simulate_water <- function(means_vector, model) {
  intercept_sim <- rnorm(n_iterations, mean = coef(model)[1], sd = summary(model)$coefficients[1,2])
  slope_sim <- rnorm(n_iterations, mean = coef(model)[2], sd = summary(model)$coefficients[2,2])
  residuals_sim <- rnorm(n_iterations, mean = 0, sd = summary(model)$sigma)

  rep(means_vector, each = n_iterations) * rep(slope_sim, times = length(means_vector)) +
    rep(intercept_sim, times = length(means_vector)) +
    rep(residuals_sim, times = length(means_vector))
}

# List of target taxa and corresponding regression models
water_sim_targets <- list(
  "Glyptops sp." = Barrick_lm, # turtles -> Barrick regression
  "Naomichelys sp." = Barrick_lm, # turtles -> Barrick regression
  "Neosuchian G" = Amiot_lm, # crocs -> Amiot regression
  "Neosuchian A" = Amiot_lm, # crocs -> Amiot regression
  "Neosuchian B" = Amiot_lm # crocs -> Amiot regression
)

# Initialize a list to store results
```

```

water_sim_results <- list()

# Initialize list to store full simulated distributions
water_sim_distributions <- list()

# Loop over each target taxon
set.seed(123)
for (taxon in names(water_sim_targets)) {
  sim_result <- simulate_water(bootstrap_results[[taxon]]$means, water_sim_targets[[taxon]])

  # Save summary results
  water_sim_results[[taxon]] <- list(
    mean = mean(sim_result),
    ci = quantile(sim_result, probs = c(0.025, 0.975))
  )

  # Save full distribution
  water_sim_distributions[[taxon]] <- sim_result

  # Print summary results
  cat("\n", taxon, "Water Mean:", round(water_sim_results[[taxon]]$mean, 2))
  cat("\n95% CI:", round(water_sim_results[[taxon]]$ci, 2), "\n")
}

```

```

##
## Glyptops sp. Water Mean: -8.29
## 95% CI: -10.52 -5.95
##
## Naomichelys sp. Water Mean: -7.77
## 95% CI: -9.81 -5.8
##
## Neosuchian G Water Mean: -7.6
## 95% CI: -11.4 -3.67
##
## Neosuchian A Water Mean: -5.06
## 95% CI: -8.76 -1.17
##
## Neosuchian B Water Mean: -5.52
## 95% CI: -9.32 -1.91

```

```

# Simulate single d180w distribution based on crocG and Glyptops d180w
bootstrapped_d180w <- replicate(n_iterations, {
  sample_glyp <- sample(water_sim_distributions[["Glyptops sp."], size = 1, replace = TRUE)
  sample_crocG <- sample(water_sim_distributions[["Neosuchian G"]], size = 1, replace = TRUE)
  mean(c(sample_glyp, sample_crocG)) # Average of the two
})

# Analyze combined distribution
mean_d180w <- mean(bootstrapped_d180w)
sd_d180w <- sd(bootstrapped_d180w)
quantiles <- quantile(bootstrapped_d180w, probs = c(0.025, 0.975))

mean_alld180water_synth <- mean_d180w

```

```

alld180water_lower <- quantiles[1]
alld180water_upper <- quantiles[2]

# Print results
cat(paste0("Mean  $\delta^{18}O_{\text{w}}$  $: ", round(mean_d180w, 2), "\n"))

## Mean  $\delta^{18}O_{\text{w}}$  $: -7.9

cat(paste0("95\% CI for  $\delta^{18}O_{\text{w}}$  $: [", round(quantiles[1], 2), ", ", round(quantiles[2], 2),

## 95\% CI for  $\delta^{18}O_{\text{w}}$  $: [-10.03, -5.77]

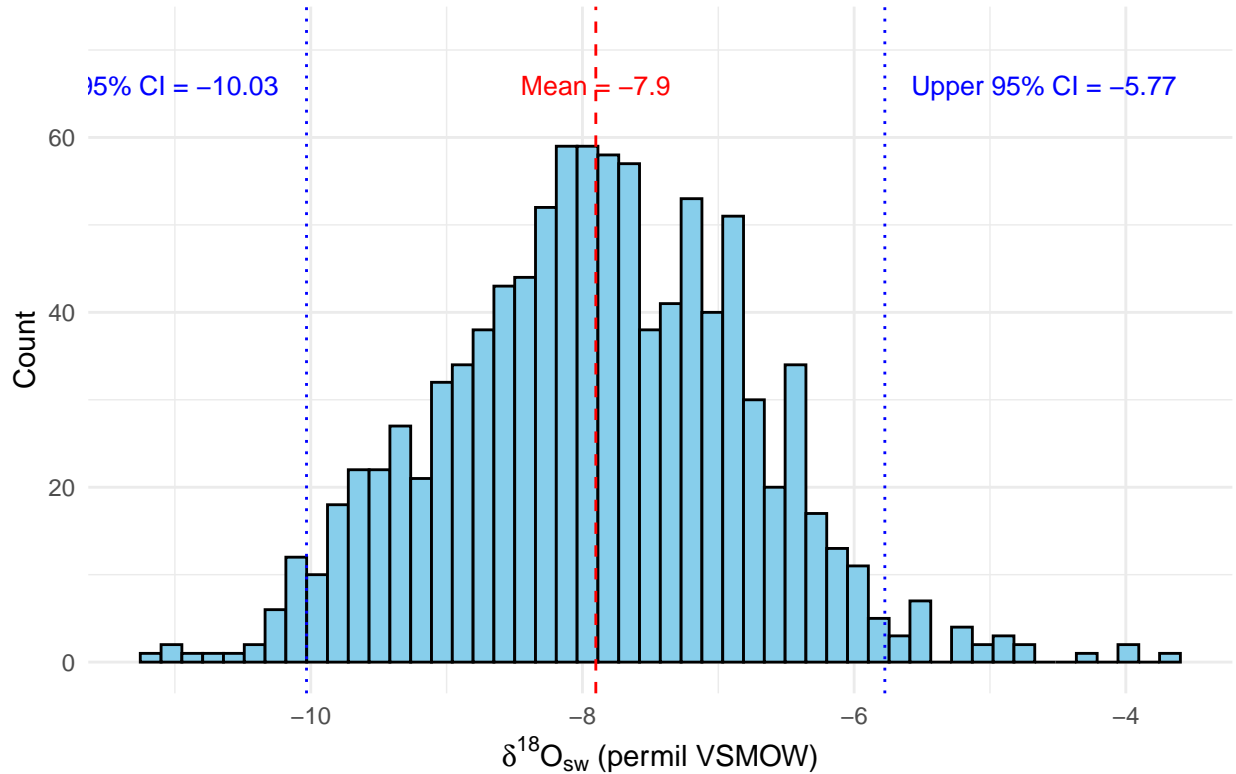
mean_alld180water_synth <- mean_d180w
alld180water_lower <- quantiles[1]
alld180water_upper <- quantiles[2]

# Plot d180sw Histogram
# Compute y-axis range for label placement
temp_plot <- ggplot(data.frame(d180w = bootstrapped_d180w), aes(x = d180w)) +
  geom_histogram(bins = 50)
y_max <- max(ggplot_build(temp_plot)$data[[1]]$count)
buffer <- y_max * 0.1
label_y <- y_max + buffer

# Generate plot
ggplot(data.frame(d180w = bootstrapped_d180w), aes(x = d180w)) +
  geom_histogram(fill = "skyblue", color = "black", bins = 50) +
  geom_vline(xintercept = mean_alld180water_synth, color = "red", linetype = "dashed") +
  geom_vline(xintercept = c(alld180water_lower, alld180water_upper), color = "blue", linetype = "dotted") +
  annotate("text", x = mean_alld180water_synth, y = label_y,
    label = paste("Mean =", round(mean_alld180water_synth, 2)),
    vjust = 0, color = "red", size = 3.5) +
  annotate("text", x = alld180water_lower, y = label_y,
    label = paste("Lower 95% CI =", round(alld180water_lower, 2)),
    vjust = 0, color = "blue", size = 3.5, hjust = 1.1) +
  annotate("text", x = alld180water_upper, y = label_y,
    label = paste("Upper 95% CI =", round(alld180water_upper, 2)),
    vjust = 0, color = "blue", size = 3.5, hjust = -0.1) +
  ylim(0, label_y * 1.1) +
  labs(
    title = expression("Simulated " $\delta^{18}O$ "[sw]" distribution"),
    x = expression( $\delta^{18}O$ "[sw]" (permil VSMOW)),
    y = "Count"
  ) +
  theme_minimal()

```

Simulated $\delta^{18}\text{O}_{\text{sw}}$ distribution



```
ggsave(file.path(results_dir, "d18Osw_Histogram.png"))
```

4 Water Temperature Simulations

```
delta_0p <- combined_data$values[combined_data$group == "Lepisosteidae"]
delta_0w <- bootstrapped_d18Ow
NIST_sim <- combined_data$values[combined_data$group == "NIST120c_aka_NBS120c"]

# Simulate intercepts and slopes separately
intercept_sim <- rnorm(n_iterations, mean = coef(PLN_lm)[1], sd = summary(PLN_lm)$coefficients[1,2])
slope_sim <- rnorm(n_iterations, mean = coef(PLN_lm)[2], sd = summary(PLN_lm)$coefficients[2,2])

# Simulate residuals
residual_sim <- rnorm(n_iterations, mean = 0, sd = summary(PLN_lm)$sigma)

# Now compute Tw_simulations properly
set.seed(123)
Tw_simulations <- intercept_sim +
  slope_sim * (delta_0p + (22.6 - NIST_sim) - delta_0w) +
  residual_sim

mean_Tw <- mean(Tw_simulations)
ci_Tw <- quantile(Tw_simulations, probs = c(0.025, 0.975), na.rm = TRUE)
```

```
cat("\nWater Temperature Mean:", mean_Tw)
```

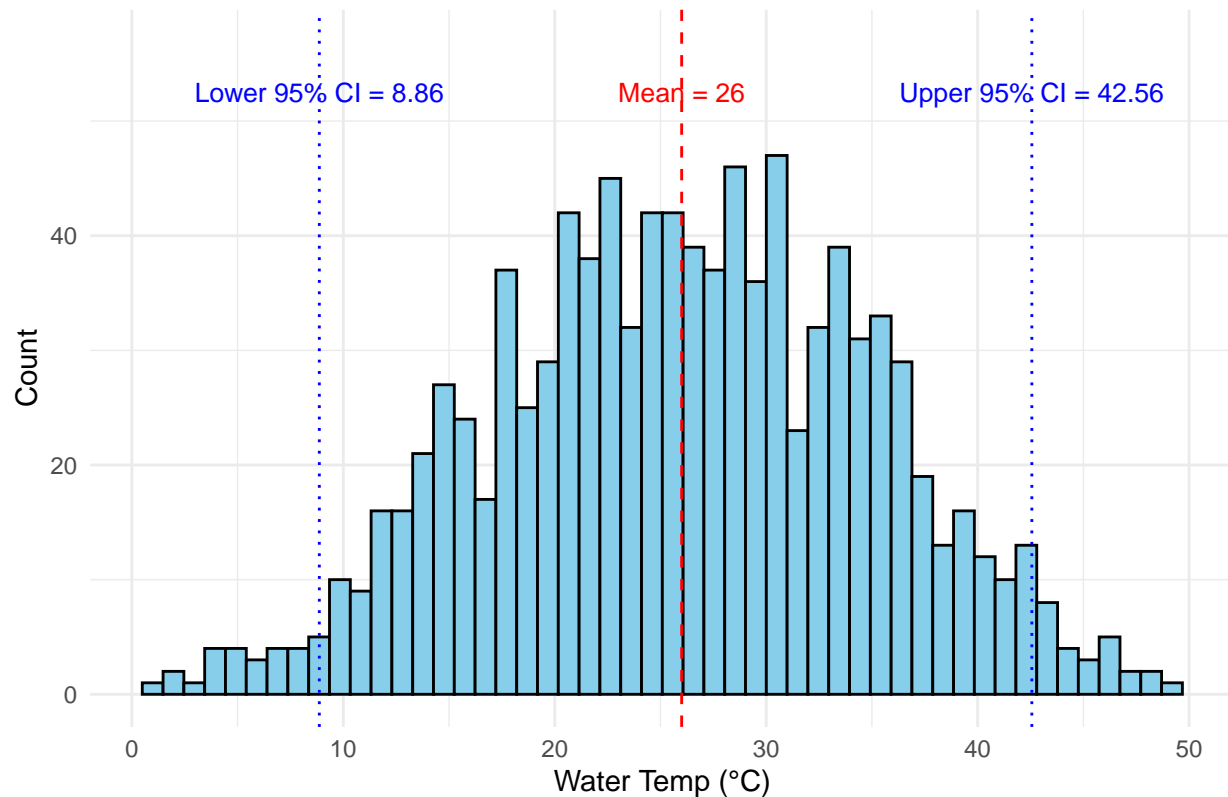
```
##  
## Water Temperature Mean: 26.00227
```

```
cat("\n95% CI:", ci_Tw, "\n")
```

```
##  
## 95% CI: 8.864743 42.56411
```

```
# Plot Tw histogram  
# Calculate max y-value for adjustment (temporary plot to get y)  
temp_plot <- ggplot(data.frame(Tw = Tw_simulations), aes(x = Tw)) +  
  geom_histogram(bins = 50)  
y_max <- ggplot_build(temp_plot)$data[[1]]$count %>% max()  
  
# Add 10% headroom above max y-value for labels  
buffer <- y_max * 0.1  
label_y <- y_max + buffer  
  
# Final plot with labels and spacing  
ggplot(data.frame(Tw = Tw_simulations), aes(x = Tw)) +  
  geom_histogram(fill = "skyblue", color = "black", bins = 50) +  
  geom_vline(xintercept = mean_Tw, color = "red", linetype = "dashed") +  
  geom_vline(xintercept = ci_Tw, color = "blue", linetype = "dotted") +  
  annotate("text", x = mean_Tw, y = label_y, label = paste0("Mean = ", round(mean_Tw, 2)),  
    vjust = 0, color = "red", size = 3.5) +  
  annotate("text", x = ci_Tw[1], y = label_y, label = paste0("Lower 95% CI = ", round(ci_Tw[1], 2)),  
    vjust = 0, color = "blue", size = 3.5) +  
  annotate("text", x = ci_Tw[2], y = label_y, label = paste0("Upper 95% CI = ", round(ci_Tw[2], 2)),  
    vjust = 0, color = "blue", size = 3.5) +  
  ylim(0, label_y * 1.1) +  
  labs(title = "Simulated Water Temperature (Tw)",  
    x = "Water Temp (°C)", y = "Count") +  
  theme_minimal()
```


Simulated Water Temperature (Tw)



```
ggsave(file.path(results_dir, "Tw_Histogram.png"))
```

5 Air Temperature Simulations

```
# Simulate intercept and slope
intercept_sim <- rnorm(n_iterations, mean = coef(TwTa_lm)[1], sd = summary(TwTa_lm)$coefficients[1, 2])
slope_sim <- rnorm(n_iterations, mean = coef(TwTa_lm)[2], sd = summary(TwTa_lm)$coefficients[2, 2])
residuals_sim <- rnorm(n_iterations, mean = 0, sd = summary(TwTa_lm)$sigma)

# Now simulate Ta based on Tw_simulations
set.seed(123)
Ta_simulations <- intercept_sim + slope_sim * Tw_simulations + residuals_sim
mean_Ta <- mean(Ta_simulations)
ci_Ta <- quantile(Ta_simulations, probs = c(0.025, 0.975))

cat("\nAir Temperature Mean:", mean_Ta)

##
## Air Temperature Mean: 24.24343

cat("\n95% CI:", ci_Ta, "\n")
```

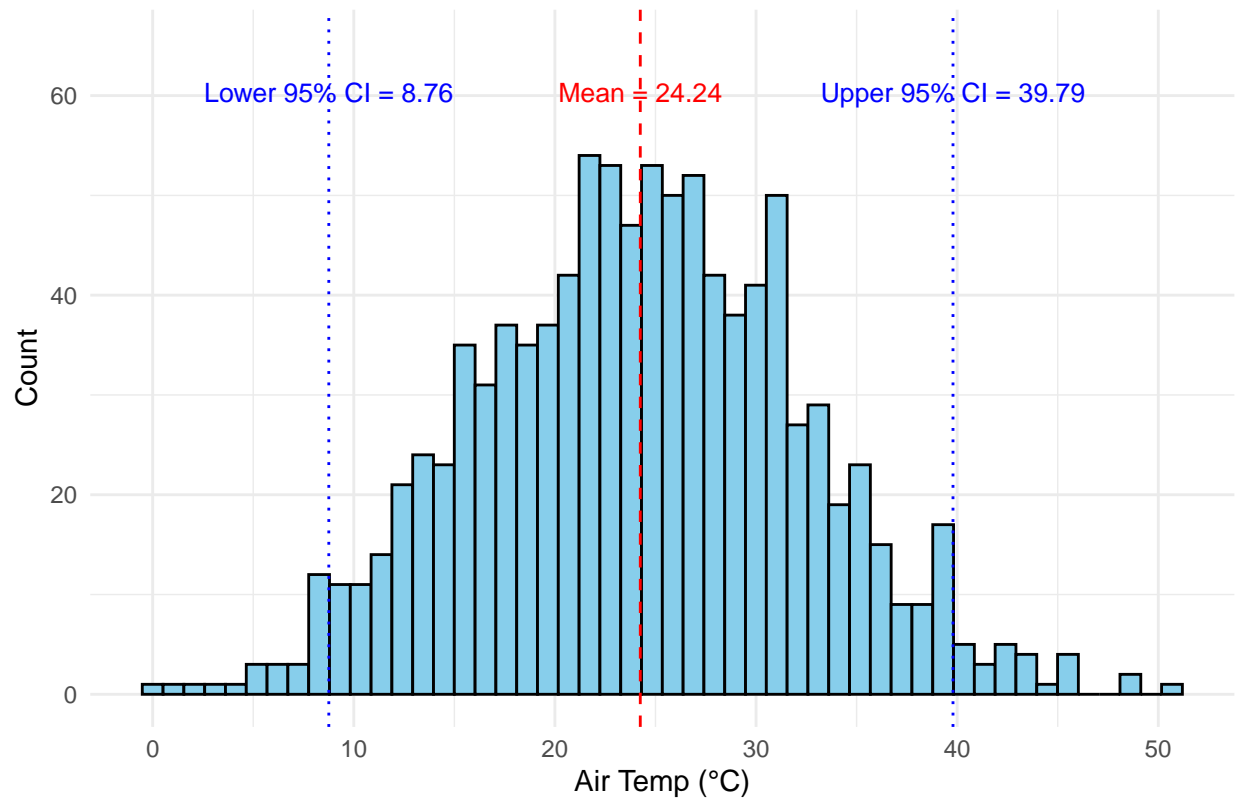
```
##
## 95% CI: 8.757978 39.79234

# Plot Ta histogram
# Calculate max y-value for adjustment (temporary plot to get y)
temp_plot <- ggplot(data.frame(Ta = Ta_simulations), aes(x = Ta)) +
  geom_histogram(bins = 50)
y_max <- ggplot_build(temp_plot)$data[[1]]$count %>% max()

# Add 10% headroom above max y-value for labels
buffer <- y_max * 0.1
label_y <- y_max + buffer

# Final plot with labels and spacing
ggplot(data.frame(Ta = Ta_simulations), aes(x = Ta)) +
  geom_histogram(fill = "skyblue", color = "black", bins = 50) +
  geom_vline(xintercept = mean_Ta, color = "red", linetype = "dashed") +
  geom_vline(xintercept = ci_Ta, color = "blue", linetype = "dotted") +
  annotate("text", x = mean_Ta, y = label_y, label = paste0("Mean = ", round(mean_Ta, 2)),
    vjust = 0, color = "red", size = 3.5) +
  annotate("text", x = ci_Ta[1], y = label_y, label = paste0("Lower 95% CI = ", round(ci_Ta[1], 2)),
    vjust = 0, color = "blue", size = 3.5) +
  annotate("text", x = ci_Ta[2], y = label_y, label = paste0("Upper 95% CI = ", round(ci_Ta[2], 2)),
    vjust = 0, color = "blue", size = 3.5) +
  ylim(0, label_y * 1.1) +
  labs(title = "Simulated Air Temperature (Ta)",
    x = "Air Temp (°C)", y = "Count") +
  theme_minimal()
```

Simulated Air Temperature (Ta)



```
ggsave(file.path(results_dir, "Ta_Histogram.png"))
```