HW1

2024-10-22

```
load("ants.Rdata")
summary(ants$abundance)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    6.75   35.50   86.31   99.25  767.00
```

```
summary(ants$moisture)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.00   15.40   17.95   17.52   19.90   21.50
```

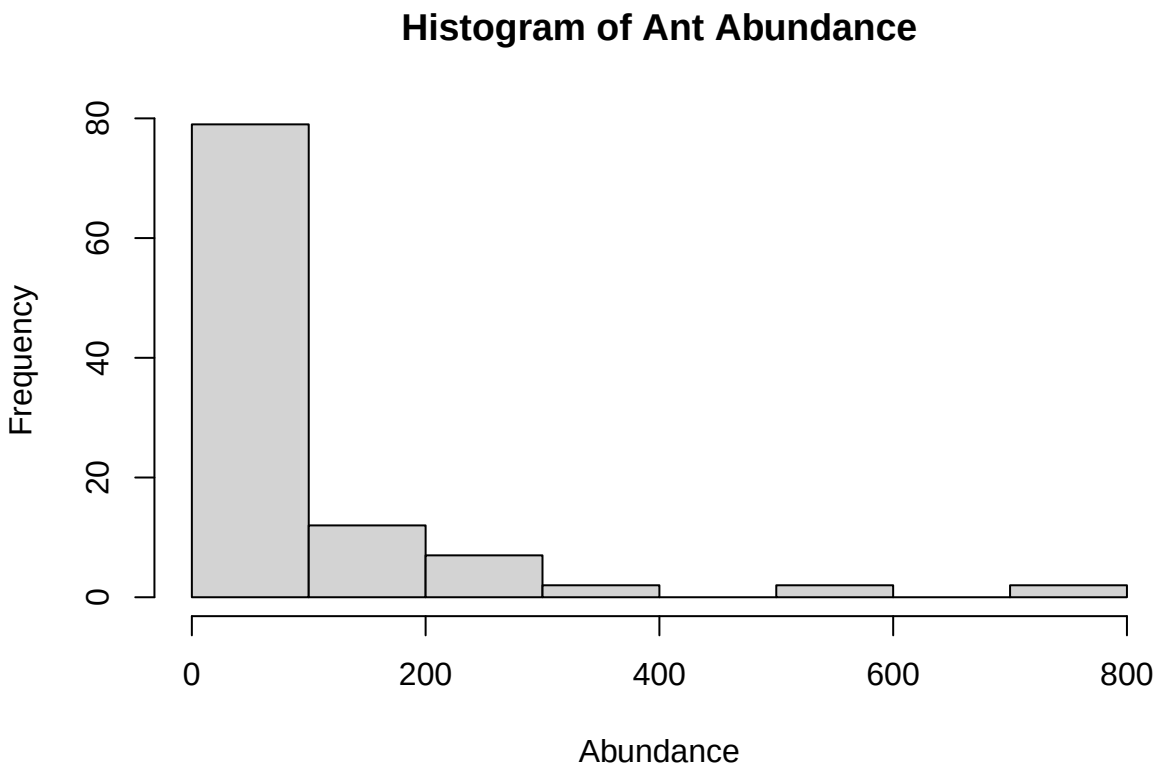
```
var(ants$abundance)
```

```
## [1] 19882.49
```

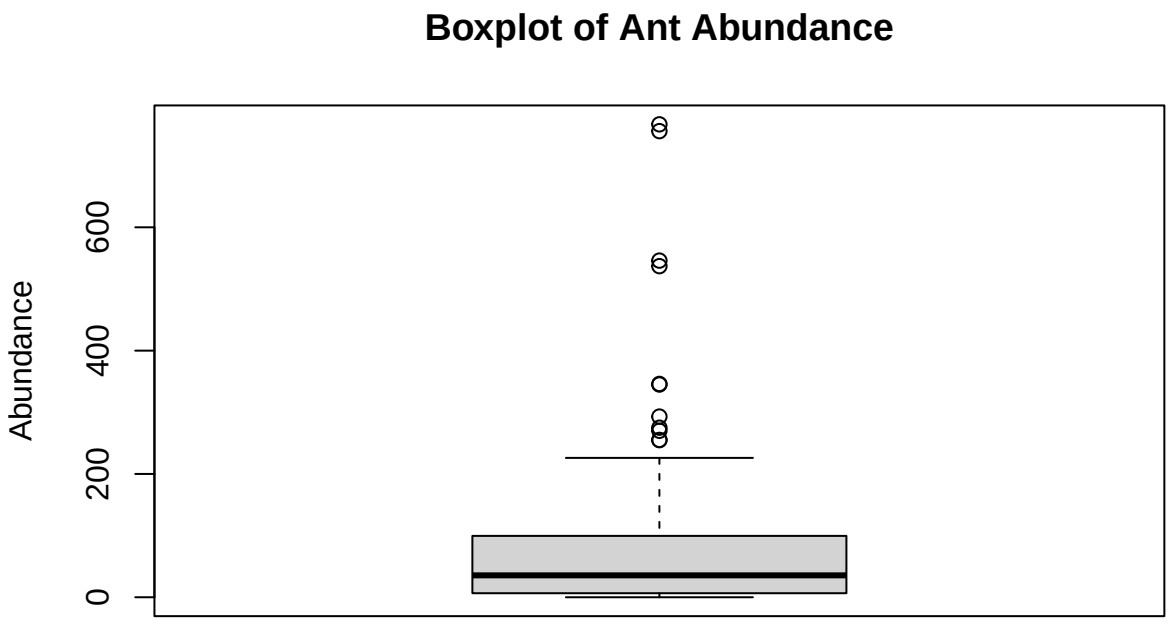
```
var(ants$moisture)
```

```
## [1] 7.083322
```

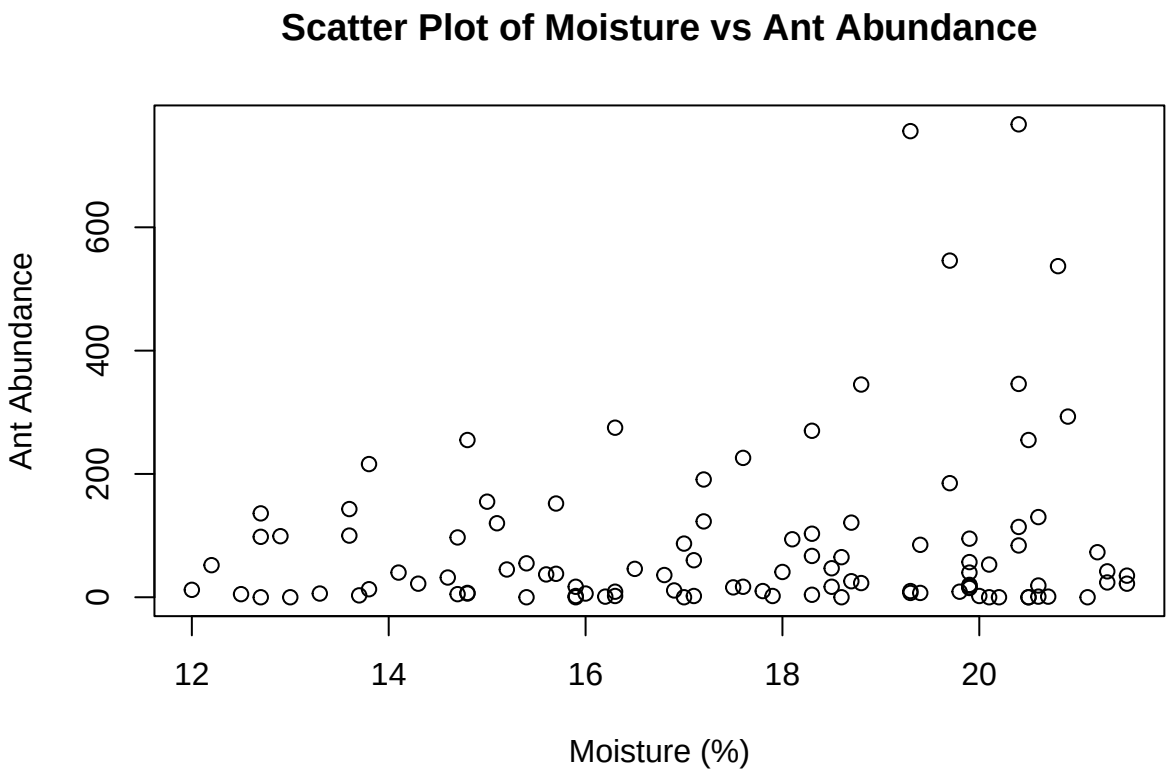
```
hist(ants$abundance, main = "Histogram of Ant Abundance", xlab = "Abundance", ylab = "Frequency")
```




```
boxplot(ants$abundance, main = "Boxplot of Ant Abundance", ylab = "Abundance")
```




```
plot(ants$moisture, ants$abundance,
     main = "Scatter Plot of Moisture vs Ant Abundance",
     xlab = "Moisture (%)", ylab = "Ant Abundance")
```



To define the likelihood function in R, we use a function that computes the product of probabilities for all observations.

```r
# Define the likelihood function
likelihood_function <- function(params, abundance, moisture) {
  beta_0 <- params[1]  # The intercept parameter
  beta_1 <- params[2]  # The coefficient for moisture
  phi <- params[3]     # The overdispersion parameter

  # Calculate the mean (mu) for each observation
  mu <- exp(beta_0 + beta_1 * moisture)

  # Compute the negative binomial likelihood for each observation
  likelihoods <- dnbinom(abundance, size = 1/phi, mu = mu, log = FALSE)

  # Calculate the total likelihood by multiplying individual likelihoods
  total_likelihood <- prod(likelihoods)

  return(total_likelihood)
}

initial_params <- c(1, 0.1, 0.5)  # Example starting values for beta_0, beta_1, and phi
likelihood_value <- likelihood_function(initial_params, ants$abundance, ants$moisture)
likelihood_value
```

```
## [1] 0
```

If we use the set of parameters described in the example above, the likelihood value results 0. One possible reason is that the number is too small to exhibit. That's why we derive the following log likelihood function.

```
# Define the log-likelihood function
log_likelihood <- function(params, abundance, moisture) {
  beta_0 <- params[1]  # The intercept parameter
  beta_1 <- params[2]  # The coefficient for moisture
  phi <- params[3]     # The overdispersion parameter

  # Calculate the mean (mu) for each observation
  mu <- exp(beta_0 + beta_1 * moisture)

  # Compute the log-likelihood for each observation
  log_lik <- sum(
    lgamma(abundance + 1 / phi) - lgamma(1 / phi) - lgamma(abundance + 1) +
    (1 / phi) * log(1 / (1 + mu * phi)) +
    abundance * log(mu * phi / (1 + mu * phi))
  )

  return(log_lik)  # Return the log-likelihood
}

initial_params <- c(1, 0.1, 0.5)  # Example starting values for beta_0, beta_1, and phi
log_likelihood_value <- log_likelihood(initial_params, ants$abundance, ants$moisture)
log_likelihood_value
```

```
## [1] -1148.303
```

Using the same set of parameters, the log likelihood values result -1148, which makes more sense.

Question 3: Estimating Equation for β_1

We are tasked with deriving the estimating equation for β_1, which involves maximizing the log-likelihood function. The maximum likelihood estimate (MLE) is found by solving this equation using numerical optimization.

Step 1: Define the Log-Likelihood Function

We reuse the log-likelihood function from Question 2, which we will optimize to find β_1, β_0, and ϕ.

```
# Define the log-likelihood function
log_likelihood <- function(params, abundance, moisture) {
  beta_0 <- params[1]  # The intercept parameter
  beta_1 <- params[2]  # The coefficient for moisture
  phi <- params[3]     # The overdispersion parameter

  # Calculate the mean (mu) for each observation
  mu <- exp(beta_0 + beta_1 * moisture)

  # Compute the log-likelihood for each observation
  log_lik <- sum(
    lgamma(abundance + 1 / phi) - lgamma(1 / phi) - lgamma(abundance + 1) +
```

```
    (1 / phi) * log(1 / (1 + mu * phi)) +
    abundance * log(mu * phi / (1 + mu * phi))
  )

  return(log_lik)  # Return the log-likelihood
}
```

Question 3: Estimating Equation for β_1

We are tasked with deriving the estimating equation for β_1, which involves maximizing the log-likelihood function. The maximum likelihood estimate (MLE) is found by solving this equation using numerical optimization.

Step 1: Define the Log-Likelihood Function

We use the following log-likelihood function for the negative binomial distribution, where the parameters β_0, β_1, and ϕ need to be estimated.

```
# Define the log-likelihood function
log_likelihood <- function(params, abundance, moisture) {
  beta_0 <- params[1]  # Intercept
  beta_1 <- params[2]  # Coefficient for moisture
  phi <- params[3]     # Overdispersion parameter

  # Calculate the mean (mu) for each observation
  mu <- exp(beta_0 + beta_1 * moisture)

  # Compute the log-likelihood for each observation
  log_lik <- sum(
    lgamma(abundance + 1 / phi) - lgamma(1 / phi) - lgamma(abundance + 1) +
    (1 / phi) * log(1 / (1 + mu * phi)) +
    abundance * log(mu * phi / (1 + mu * phi))
  )

  return(log_lik)
}
```

Step 2: Negative Log-Likelihood Function

Since the `optim()` function in R minimizes functions, we use the negative log-likelihood function for the optimization process. The negative log-likelihood is simply the negative of the log-likelihood function.

```
# Define the negative log-likelihood function for optimization
neg_log_likelihood <- function(params) {
  -log_likelihood(params, ants$abundance, ants$moisture)
}
```

Step 3: Perform Optimization to Estimate Parameters

We now use the `optim()` function to find the maximum likelihood estimates (MLE) for β_0, β_1, and ϕ. The initial guesses for the parameters are provided, and the optimization is performed using the BFGS method.

```r
# Initial parameter guesses: beta_0 = 1, beta_1 = 0.1, phi = 0.5
initial_params <- c(1, 0.1, 0.5)

# Perform optimization to estimate beta_0, beta_1, and phi
optim_result <- optim(initial_params, neg_log_likelihood, method = "BFGS")
```

```
## Warning in log(1/(1 + mu * phi)): NaNs produced
```

```r
# Extract the estimated parameters
estimated_params <- optim_result$par

# Assign the estimates to beta_0_hat, beta_1_hat, and phi_hat
beta_0_hat <- estimated_params[1]
beta_1_hat <- estimated_params[2]
phi_hat <- estimated_params[3]

# Display the estimated parameters
beta_0_hat
```

```
## [1] 2.504213
```

```r
beta_1_hat
```

```
## [1] 0.1091764
```

```r
phi_hat
```

```
## [1] 2.288232
```

Conclusion

The estimated parameters β_0, β_1, and ϕ are calculated using numerical optimization. The MLE of β_1 is particularly important for the analysis, as it describes the relationship between moisture and ant abundance.

These values will be used for further analysis in the next questions.